ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

PREDICCIONES METEREOLÓGICAS OBTENIDAS DE DATOS DE ESTACIONES NAVARRAS UTLIZANDO TÉCNICAS DE MINERÍA DE DATOS

Asier Lizarraga Oroquieta

Ramón Fuentes González

Pamplona, 30 de Julio del 2010

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Agradecimientos

La realización de este proyecto no hubiese sido posible sin la colaboración y paciencia de todas las personas que me han ayudado durante meses en la realización del proyecto así como el material proporcionado.

No podemos dejar de agradecer toda la atención recibida por nuestro tutor Ramón Fuentes, que siempre se ha mostrado accesible y ha prestado ayuda siempre que lo hemos necesitado. Y finalmente, agradecer a nuestros padres, familiares y amigos todo el apoyo que nos han dado y la paciencia que han tenido con nosotros durante los largos meses que nos ha ocupado todo este trabajo. Este proyecto también os pertenece. Muchísimas gracias a todos vosotros.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Índice

Introducción. 6
Que es el proceso de extracción del conocimiento (KDD)? 6
Fase de integración y recopilación 9
Fase de selección limpieza y trasformación 16
Fase de Minería de datos 29
Fase de evaluación e interpretación 34
Fase de difusión, uso y monitorización 37
Como realizar la implantación de un programa de minería de datos 38
Fases de implantación 40
CRISP-DM El modelo y guía de referencia 41
Herramienta WEKA 43
Antecedentes y Objetivos. 46
Recopilación y presentación de datos. 47
Desarrollo del trabajo 48
Estudio de las 3 estaciones: 49
Estación de Montes de Cierzo: 50
Estación de Larrabide 146
Estación de Gorramend 236
Comparativa de los resultados de las Estaciones: 326
Aplicación: 338
Especificación de la aplicación: 338
Pruebas de la aplicación: 339
Análisis de Resultados. 340
Estudio Económico 341
Conclusiones. 342
Manual de usuario 343
Bibliografía. 346

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Introducción.

En primer lugar antes de comenzar a explicar el trabajo realizado en el proyecto, es necesario definir una serie de conceptos claves para entender y comprender el trabajo realizado en el desarrollo de mi proyecto.

Que es el proceso de extracción del conocimiento (KDD)?

El KDD (Knowledge Discovery from Databases, KDD) se define como "El proceso no trivial de identificar patrones valiosos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de datos". En esta definición se resumen cuales deben de ser las propiedades deseables del conocimiento extraído:

- **Valido:** hace referencia a que los patrones deben de seguir siendo precisos para datos nuevos (con un cierto grado de certidumbre), y no solo para aquellos que han usados en su obtención.

- **Novedoso:** que aporte algo desconocido tanto para el sistema y preferiblemente para el usuario.

- **Potencialmente útil:** la información debe de conducir a acciones que reporten algún tipo de benéfico para el usuario.

- **Comprensible:** la extracción de patrones no compresibles dificulta o imposibilita su interpretación, revisión, validación y uso en la toma de decisiones. De hecho una información incomprensible no proporciona conocimiento (al menos desde el punto de vista de su utilidad).

Como se deduce de la anterior definición, el KDD es un proceso complejo que incluye no solo la obtención de los modelos o patrones (el objetivo de la minería de datos) sino también la evaluación y posible interpretación de los mismos, tal como y como se refleja en la **Figura 1.**

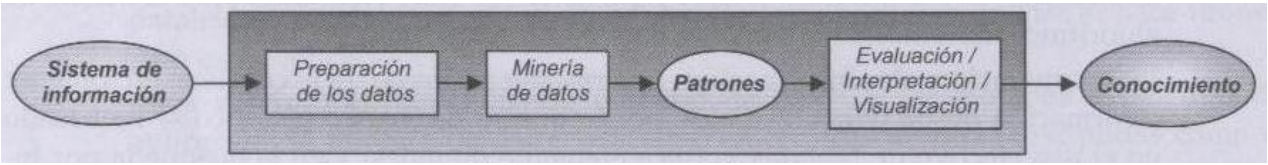


Figura 1: Proceso de KDD

Así los sistemas KDD permiten la selección, limpieza, transformación y proyección de los datos; analizar los datos para extraer patrones y modelos adecuados; evaluar e interpretar los patrones para convertirlos en conocimiento (Knowledge); consolidar el conocimiento resolviendo posibles conflictos con conocimiento previamente extraído; y hacer el conocimiento disponible para su uso.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Esta definición del proceso clarifica la relación entre el KDD y la minería de datos: El KDD es el proceso global de descubrir el conocimiento útil desde las bases de datos mientras que la minería de datos se refiere a la aplicación de los métodos de aprendizaje y estadísticos para la obtención de patrones y modelos. Al ser la fase de generación de modelos, comúnmente se asimila KDD con minería de datos.

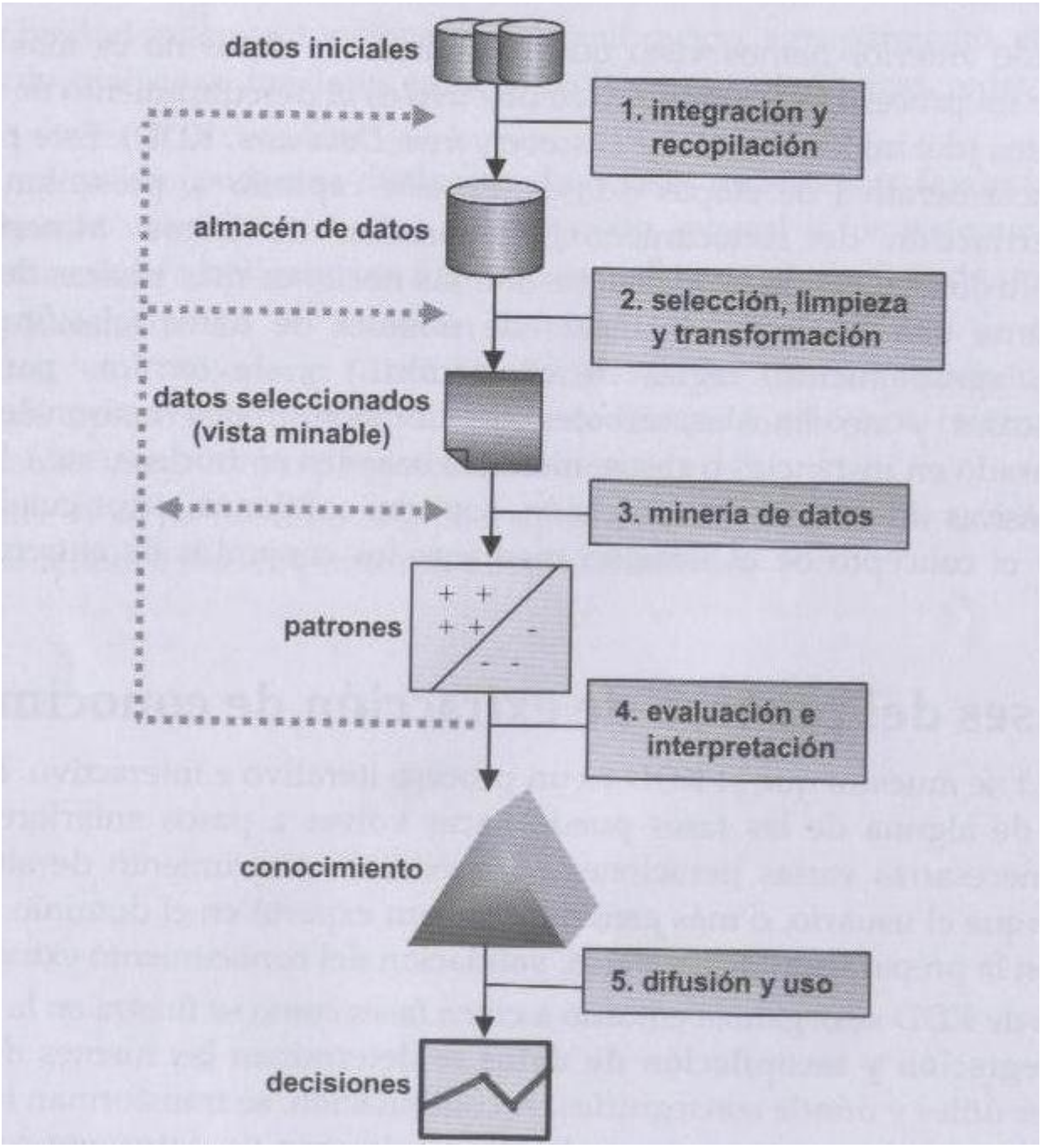


Figura 2: Fases del descubrimiento del conocimiento en bases de datos, KDD.

Como se muestra en la **Figura 2** el KDD es un proceso iterativo e interactivo ya que la salida de alguna de las fases puede hacer volver a alguno de los pasos anteriores y porque a menudo son necesarias varias interacciones para extraer conocimiento de alta calidad.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Es interactivo porque el usuario, o más generalmente un experto en el dominio de problema, debe ayudar en la preparación de los datos, validación del conocimiento extraído, etc. El proceso KDD se ilustra en torno a 5 fases como se ilustra en la figura 2.En la **fase de integración y recopilación de datos** se determinan las fuentes de información que pueden ser útiles y donde conseguirlas. A continuación, se transforman todos los datos a un formato común, frecuentemente mediante un almacén de datos que consigue unificar de manera operativa toda la información recogida, detectando y resolviendo las inconsistencias.

Este almacén de datos (data warehouse) facilita enormemente la "navegación" y visualización previa de sus datos, para discernir qué aspectos puede interesar que sean estudiados. Dado que los datos provienen de diferentes fuentes, pueden contener valores erróneos o faltantes. Estas situaciones se tratan en la fase de **selección, limpieza y trasformación** en la que se eliminan o corrigen los datos incorrectos y se decide la estrategia a seguir con los datos incompletos. Además se proyectan los datos para considerar únicamente aquellas variables o atributos que van a ser relevantes, con el objetivo de hacer más fácil la tarea propia de minería y para que los resultados de la misma sean más útiles. La selección incluye tanto una criba o fusión horizontal (filas/registros) como vertical (columnas/atributos). Las dos primeras fases se suelen englobar bajo el nombre de "Preparación de datos". En la **fase de minería de datos**, se decide cual es la tarea a realizar (clasificar, agrupar, etc.) y se elige el método que se va a utilizar. En la **fase de valuación e interpretación** se evalúan los patrones y se analizan por los expertos, y si es necesario se vuelven a las fases anteriores para una nueva iteración. Esto incluye resolver posibles conflictos con el conocimiento que se disponían anteriormente. Finalmente, en la **fase de difusión** se hace uso del nuevo conocimiento y se hace partícipe de él a todos los posibles usuarios.

A continuación pasamos a ver brevemente las fases del KDD.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Fase de integración y recopilación

El primer paso en el proceso de extracción del conocimiento a partir de datos es precisamente reconocer y reunir los datos con los que se va a trabajar. Si esta recopilación se va a realizar para una tarea puntual y no involucra muchas cantidades y variedades de datos simples, es posible que el sentido común sea suficiente para obtener un conjunto de datos con calidad suficiente como para poder empezar a trabajar. En cambio, si requerimos datos de distintas fuentes, tanto externas como internas a la organización, con datos complejos y variados, posiblemente en grandes cantidades y además cambiantes, con los que se desea realizar a largo o medio plazo diversas tareas de minería de datos, es posible que nuestro sentido común no sea suficiente para hacer una recopilación e integración en condiciones.

Al igual que la tecnología de bases de datos ha desarrollado una series de modelos de datos (como el relacional), de lenguajes de consulta y actualización, de reglas de actividad, etc., para trabajar con información trasnacional de una organización, veremos que existe una tecnología relativamente reciente, denominada "almacén de datos" (data warehouses) que pretende proporcionar metodologías y tecnologías para recopilar e integrar los datos históricos de una organización, cuyo fin es el análisis y obtención de resúmenes e informes complejos y la extracción de conocimiento. Esta tecnología está diseñada para organizar grandes volúmenes de datos de procedencia generalmente estructurada (bases de datos relacionales, por ejemplo), aunque el concepto general es útil para la organización de pequeños conjuntos de datos en aplicaciones de minería de datos.

Supóngase que en una compañía bien implantada en el ámbito europeo queremos analizar aquellos países y gamas de productos en los que las ventas vayan excepcionalmente bien (con el objetivo, por ejemplo, de premiar las oficinas comerciales de cada gama y producto), dicho de una manera más técnica, averiguar si la penetración relativa (teniendo en cuenta la permeabilidad del país en cuestión) de una gama de productos es significativamente mayor que la media de penetración en el conjunto del continente. La compañía dispone, por supuesto, de una base de datos transaccional sobre la cual operan todas las aplicaciones de la empresa: producción, ventas, facturación, proveedores etc. Lógicamente, de cada venta se registra la fecha, la cantidad y el comprador, el país de este. Con toda esta información histórica nos podemos preguntar ¿es esta información suficiente para hacer el análisis anterior? La respuesta a primera vista quizá de manera sorpréndete, es negativa. Pero, aparentemente, si tenemos detalladas las ventas de tal manera que con una consulta SQL las ventas por países de todos los productos y gamas ¿qué más puede hacer falta?

Sencillamente la respuesta, hay que buscarla fuera de la base de datos, en el contexto donde se motiva el análisis. La penetración de un artículo depende de las ventas por habitante. Sino no tenemos en cuenta la población de cada país la respuesta del análisis estará sesgada; será muy probable que entre los países con mayor penetración se encuentre Alemania, y entre los países con menor penetración se encuentre San Marino. Pero no solo eso, es posible que, si deseamos hacer un análisis más perspicaz, nos interese saber la renta per cápita de cada país incluso la distribución por edad de cada país. Dependiendo de la gama, nos puede interesar información externa verdaderamente específica. Por ejemplo las horas de sol anuales de cada país pueden ser una información valiosísima para una compañía de cosméticos. Lógicamente es más difícil vender bronceadores en Lituania que en Grecia o, dicho más técnicamente Lituania tiene menos permeabilidad a la gama de bronceadores que Grecia. Pero este hecho que nos parece tan lógico solo podrá ser descubierto por nuestras herramientas de minería de datos si somos capaces de incorporar información relativa a las horas de sol o al menos, cierta información climática de cada país.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Evidentemente, cada organización deberá recoger diferente información que le pueda ser útil para la tarea de análisis, extracción del conocimiento y, en definitiva, de toma de decisiones. En la **figura 3** se muestra las fuentes de datos que deben de ser requeridas en el caso exterior para un proceso de extracción de conocimiento satisfactorio. Solo conociendo el contexto de cada organización o de cada problema en particular se puede determinar que fuentes externas vana a ser necesarias. Además este proceso es generalmente iterativo. A medida que se va profundizando en un estudio, se pueden ir determinado datos externos que podrían ayudar y se pueden ir añadiendo a nuestros "repositorio de datos" (data warehouse). Por tanto, la tarea de mantener un repositorio o un almacén con toda la información necesaria cobra mayor relevancia y complejidad.

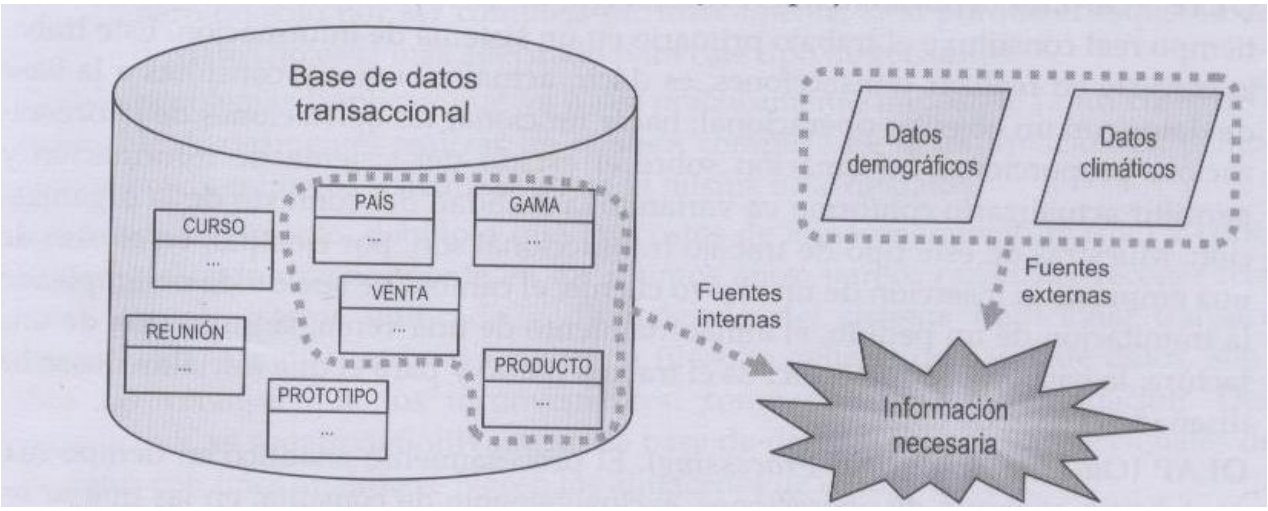


Figura 3: Fuentes de datos requeridas para responder "Países con mayor penetración de bronceadores"

El mantenimiento de esta información plantea cuestiones técnicas. En primer lugar, se requería añadir, puede que frecuentemente, nueva información a nuestro repositorio. Tanto proveniente e actualizaciones de la propia organización como de fuentes externas, tanto proveniente de actualizaciones de la propia organización como de fuentes externas, ya sean actualizaciones como nuevas incorporaciones. En segundo lugar y la que resulta la cuestión principal, ¿Hay que almacenar toda la información en la base transaccional?

A continuación vamos a explicar el concepto de data warehouse y la funcionalidad que tiene.

¿Qué es un Data warehouse?

Un almacén de datos es un conjunto de datos históricos, internos o externos, y descriptivos de un área o contexto de estudio, que están organizados y organizados de tal forma que permiten aplicar eficientemente herramientas para resumir, describir y analizar los datos con el fin de ayudar en la toma de decisiones estratégicas.

La ventaja fundamental de un almacén de datos es su diseño específico y su separación de la base de datos transaccional (base de datos relacional sobre la cual se realizan todas las tareas operativas). Un almacén de datos:

- Facilita el análisis de datos en tiempo real (OLAP)
- No disturba el OLPT (El proceso de transacciones en tiempo real)

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por tanto se puede diferenciar claramente entre bases de datos trasnacionales (u operacionales) y almacenes de datos. Dicha diferencia sea ido marcando más profundamente a medida que las tecnologías propias de amabas bases de datos (y en espacial almacenes de datos) se han ido especializando. De hecho hoy en día hay diferencias calaras, como se ve en la **tabla 1**.

Las diferencias mostradas en la tabla como veremos, distinguen claramente la manera de estructurar y diseñar almacenes de datos respecto a la forma tradicional de hacerlo con r bases de datos transaccionales.

	BASE DE DATOS TRACSACIONAL	**ALMACEN DE DATOS**
Propósito	Operaciones diarias. Soporte de aplicaciones.	Recuperación de la información, informes, análisis, minería de datos.
Tipo de datos	Datos de funcionamiento de la organización.	Datos útiles para el análisis, la sumarizacion, etc.
Características de los datos	Datos de funcionamiento, cambiantes internos, incompletos.	Datos históricos, datos internos y externos, datos descriptivos
Modelo de datos	Datos normalizados.	Datos en estrella, en copo de nieve, parcialmente des normalizados, multidimensionales.
Número y tipo de usuarios	Cientos/miles: aplicaciones, operarios, administrador de la base de datos.	Decenas: directores, ejecutivos, analistas (grajeros, mineros)
Acceso	SQL. Lectura y Escritura.	SQL y herramientas propias (slice & dice, drill, roll, pívot.). Lectura.

Tabla 1: Diferencias entre la base de datos transaccional y el almacén de datos.

Aunque ambas fuentes de datos (transaccional y almacén de datos) están separadas, es importante destacar que gran parte de los datos que se incorporan en un almacén de datos, provienen de la base de datos transaccionales. Esto supone desarrollar una tecnología de volcado y mantenimiento de los datos desde la base de datos transaccional al almacén de datos. Además, el almacén de datos debe integrar datos externos, con lo que en realidad debe de estar actualizándose frecuentemente de diferentes fuentes. El almacén de datos pasa a ser un integrador o recopilador de información de diferentes fuentes. El almacén de datos pasa a ser un integrador o recopilador de información de diferentes fuentes como se observa en la **figura 4**.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

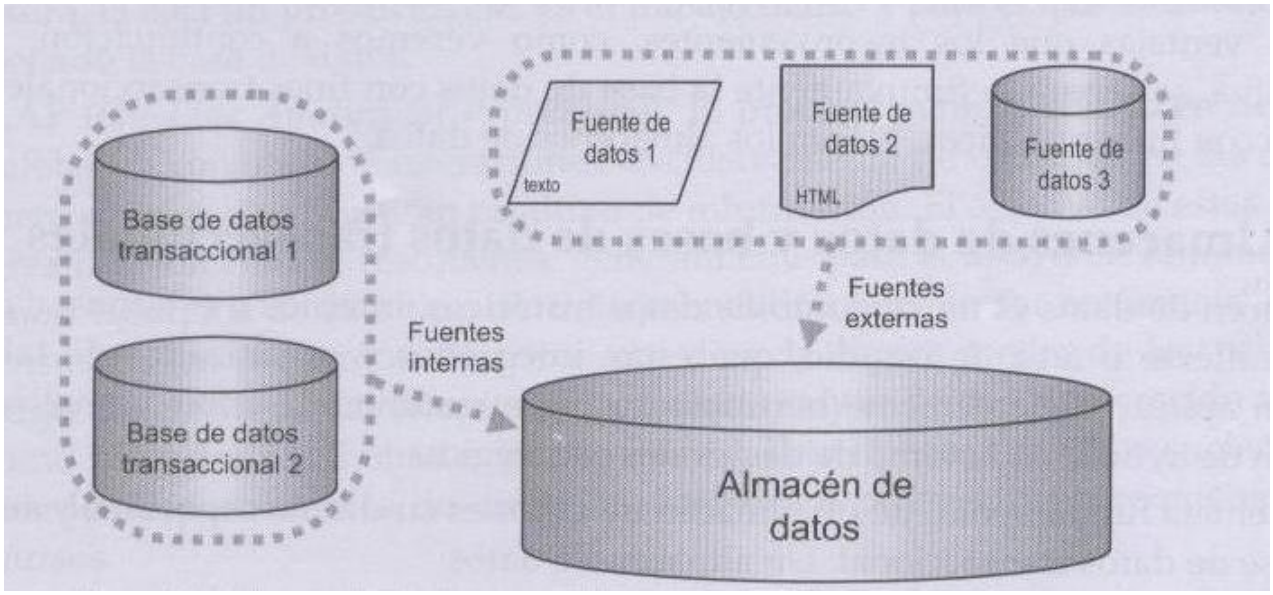


Figura 4: El almacén de datos como integrador de diferentes fuentes de datos.

La organización y el mantenimiento de esta información plantean cuestiones técnicas, fundamentalmente sobre como diseñar el almacén de datos, como cargarlo inicialmente, como mantenerlo y preservar su consistencia. No obstante, son muchas más las ventajas de esta separación que sus inconvenientes. Además, esta separación facilita la incorporación de fuentes externas, que, en otro caso, sería muy difícil de encajar en la base de datos transaccional.

A continuación vamos a explicar la relación existente entre la minería de datos y los almacenes (data warehouse):

Almacenes de datos y minería de datos

El concepto de almacenes de datos nace hace más de una década [Inmon 1992] ligado al concepto de EIS (Executive information System), el sistema de información ejecutivo de una organizaciones realidad, cuando están cubiertas las necesidades operacionales de las organizaciones se plantean herramientas informáticas para asistir o cubrir en las necesidades estrategias.

La definición original de almacén de datos es una "colección de datos orientada a un dominio, integrada, no volátil y variante en el tiempo para ayudar en las decisiones de dirección". A raíz de esta definición, parecería que los almacenes de datos son solo útiles en empresas o instituciones donde altos cargos directivos tengan que tomar decisiones. A partir de ahí, y de la difusión cada vez mayor de las herramientas de businnes intelligence y OLAP, podríamos pensar que lo almacenes de datos no se aplican en otros ámbitos científicos, médicos, ingenieriles, académicas, donde no se tratan con las variables y problemáticas típicas de las organizaciones y empresas.

Al contrario, en realidad, los almacenes de datos pueden utilizarse de muy diferentes maneras, y pueden agilizar muchos procesos diferentes de análisis. En la **figura 5** se pueden observar las distintas aplicaciones y usos que se pueden dar a un almacén de datos: herramientas de consultas e informes, herramientas EIS, herramientas OLAP y herramientas de minería de datos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

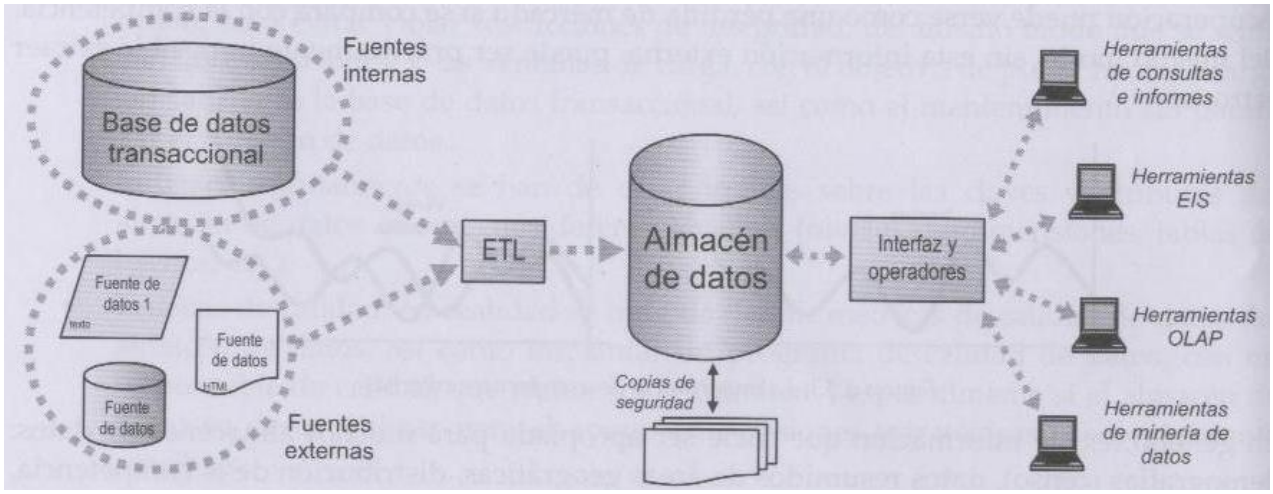


Figura 5: Perspectiva general y usos de un almacén de datos.

La variedad de usos que se muestran el ala figura anterior sugieren también la existencia de diferentes grupos de usuarios: analistas, ejecutivos, investigadores, etc. Según el carácter de estos usuarios se les puede catalogar en dos grandes grupos:

1. "Picapedreros" (o grajeros): se dedican fundamentalmente a realizar informes periódicos, ver la evolución de indicadores, controlar valores anómalos, etc.

2. "exploradores": encargados de encontrar nuevos patrones significativos utilizando técnicas OLAP o minería de datos. La estructura del almacén de datos y sus operadores facilita la obtención de diferentes vistas de análisis o "vistas minables".

Esta diferencia, y el hecho de que se catalogue a exploradores a aquellos que utilizan técnicas OLAP o minería de datos, no nos debe hacer confundir las grandes diferencias de un análisis clásico, básicamente basado en agregación, la visualización y las técnicas descriptivas estadísticas con un uso genuino de la minería de datos que no trasforma los datos en otros datos (más o menos agregados) sino que trasforma los datos en conocimiento.

Los almacenes de datos no son imprescindibles para hacer extracción de conocimiento a partir de datos. En realidad, se puede hacer minería de datos sobre un simple archivo de datos. Sin embargo, las ventajas de utilizar un almacén de datos se amortizan sobradamente a medio y largo plazo.

Esto es especialmente patente cuando nos enfrentamos a grandes volúmenes de datos, o estos aumentan con el tiempo, provienen de fuentes heterogéneas o se va a querer combinar de maneras arbitrarias y no predefinidas. En gran medida un almacén de datos también facilita la limpieza y transformación de datos.

Una vez recopilados los datos en un almacén de datos si se considera necesario se procede a la integración de los datos. Para ello habrá que integrar datos procedentes de diferentes fuentes (bases de datos, archivos, ERP.)

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

La integración es generalmente un proceso que se realiza durante la recopilación de datos y, si se realiza un almacén de datos, durante el proceso de carga del almacén de datos. El primer problema a la hora de realizar una integración desde distintas fuentes es identificar los objetos, es decir, conseguir que datos sobre el mismo objeto se unifiquen y datos de diferentes objetos permanezcan separados. Este problema se conoce como problema del esclarecimiento de la identidad. Existen dos tipos de errores que pueden ocurrir en esta integración:

- Dos o más objetos diferentes se unifican: los datos resultantes mezclaran patrones de diferentes individuos y serán un problema para extraer conocimiento. Esto será más grave cuanto más diferentes sean los dos objetos unificados.

- Dos o más fuentes de objetos iguales se dejan separadas: los patrones del mismo individuo aparecerán repartidos entre varios individuos parciales. Este problema genera menos "ruido" que el anterior, aunque es especialmente problemático cuando se usan valores agregados (el total de compras será mucho menor si consideramos un individuos en la base de datos, por ejemplo).

En general el primer problema es menos frecuente que el segundo, ya que la unificación se realiza generalmente por identificadores externos a la base de datos: numero de identidad, numero de pólizas, matriculas, tarjetas de crédito o fidelizacion, etc.

Cuando se integran (correctamente) dos fuentes diferentes de datos de distintos objetos suele suceder que aparezcan datos faltantes (el dato se registra en una fuente pero no en la otra) o datos inconsistentes (el dato es diferente en una fuente y otro). Esto se puede observar en la **figura 6.**

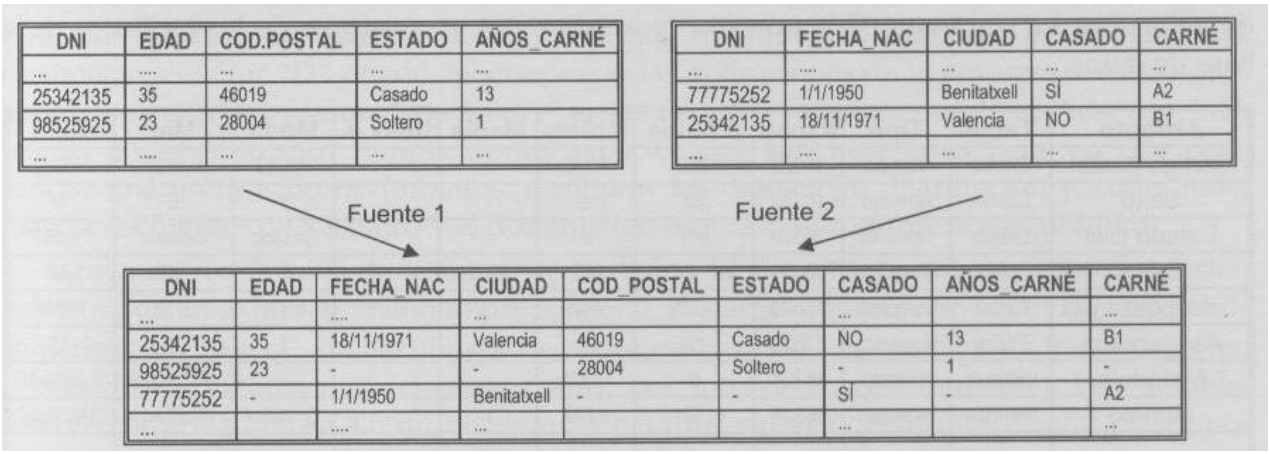


Figura 6: Ejemplos de integración de atributos de distintas fuentes

Lógicamente aparecen campos redundantes total o parcialmente: "edad" y "fecha nac", "ciudad" y "cod_postal", "estado" y "casado". Cuando sea posible se intentaran fusionar. En muchos casos los datos inconsistentes se convierten en faltantes; por ejemplo si el mismo cliente tiene estados civiles diferentes en cada fuente, es predecible dejar el valor a faltante que elegir al azar uno de los dos (o hacer la media).

Otro caso muy interesante en la integración de formatos diferentes, que se produce si tenemos codificaciones diferentes (casado/matrimonio), idiomas diferentes, medias diferentes, etc.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

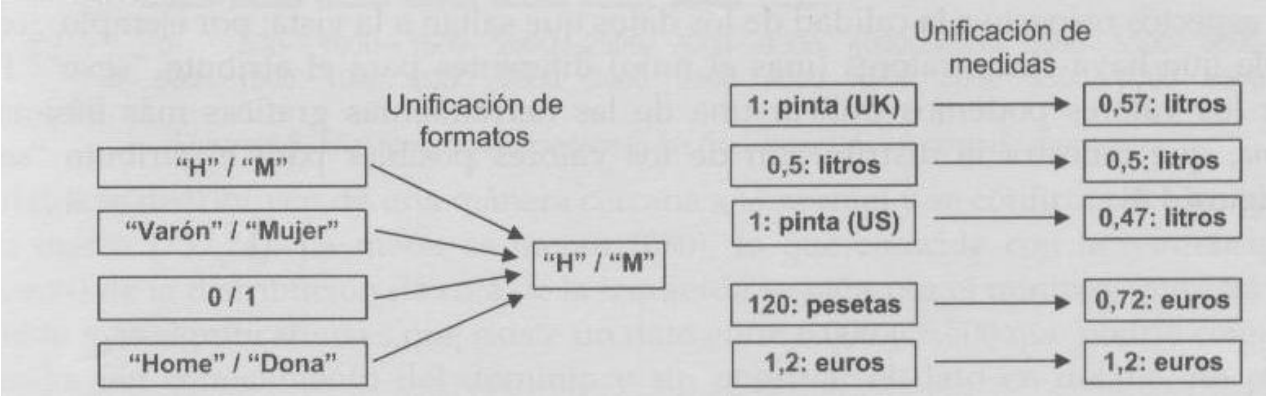


Figura 7: Ejemplos de integración: unificación de formatos y medidas.

En general existe muchas otras situaciones, algunas de ellas relativamente típicas en procesos de migración y fusión de bases de datos, pero hay otras tantas que son especificas del dominio y de las bases de datos integradas y que pueden necesitar soluciones muy especificas.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Fase de selección limpieza y trasformación

La calidad del conocimiento descubierto no solo depende del algoritmo de minería de datos utilizado, sino también de la calidad de los datos minados. Por ello, después de la recopilación el siguiente paso en el proceso KDD es seleccionar y preparar el subconjunto de datos que se va a minar, los cuales constituyen lo que se conoce como vista minable. Este paso es necesario ya que algunos datos coleccionados en la etapa anterior son irrelevantes o innecesarios para la tarea de minería que se desea realizar. Pero demás de la irrelevancia, existen otros problemas que afectan a la calidad de los datos.

A continuación vamos ver en qué consisten la selección, limpieza y trasformación de datos:

Limpieza de datos

Valores Faltantes:

Los valores faltantes, perdidos o ausentes (missing values) pueden ser remplazados por varias razones. En primer lugar, el método de minería de datos que utilicemos puede no tratar bien los datos faltantes. En segundo lugar, podemos querer agregar los datos (especialmente numéricos) para realizar otras vistas minables y que los valores faltantes no nos permitan agregar correctamente (totales medias etc.). En tercer lugar si el método es capaz de tratar campos faltantes es posible que ignore todo el ejemplo (produciendo un sesgo) o es posible que tenga un método de sustitución de campos faltantes que no sea adecuado debido a que no conoce el contexto asociado al atributo faltante.

A la hora de hablar de campos faltantes, debemos tratar de su detección y de su tratamiento. La detección de campos faltantes puede parecer sencilla. Si los datos proceden de una base de datos, basta mirar en la tabla de resumen de atributos/características y ver la cantidad de nulos que tiene cada atributo. El problema es que a veces los campos faltantes no están representados como nulos. Por ejemplo, aunque hay campos en los que las restricciones de integridad del sistema evitan introducir códigos fuera del formato para representar valores faltantes, esto al final ocurre en muchos otros, especialmente en campos de crédito con valor -1, etc. A veces son las propias restricciones de integridad las que causan el problema. Por ejemplo, ¿cuántos sistemas obligan a introducir necesariamente los dos apellidos y fuerzan que a los extranjeros, por tanto, les pongamos algún valor (“-“) en faltantes? Este tipo de situaciones complica la detección de valores faltantes. No obstante, son precisamente aquellos casos más difíciles en los que merece la pena realizar más esfuerzo, ya que muchas veces son este tipo de “nulos camuflados” los que pueden introducir sesgo en el conocimiento extraído.

Finamente, si se ha conseguido establecer los datos faltantes e, idealmente, sus causas procederemos a su tratamiento. Las posibles acciones sobre datos faltantes son:

- **Ignorar (dejar pasar):** algunos algoritmos son robustos a datos faltantes (por ejemplo árboles de decisión).

- **Eliminar (filtrar o remplazar) toda la columna (es decir quitar el atributo todos los ejemplos):** solución externa, pero a veces la proporción de nulos es tan alta que la columna no tiene arreglo.

- **Filtrar la fila:** claramente sesga los datos, porque muchas veces las causas de un dato faltante están relacionadas con casos o tipos especiales.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Remplazar el valor:** Se puede remplazar el valor manualmente (en el caso de que no hay muchos) o automáticamente por un valor que preserve la media o la varianza (globales o por clases/grupos), en el caso de valores numéricos, o por el valor moda, en caso de valores nominales.

Quizás una de las soluciones anteriores más frecuentes cuando tratamos el algoritmo a utilizar o maneja bien los nulos sea remplazar el valor. Si sustituimos un dato faltante por un dato estimado, hemos de tener en cuenta que, en primer lugar, perdemos información, ya que ya no se sabe que el dato era faltante, en segundo lugar, inventarnos información, con los riesgos que pueda tener que sea errónea. El primer problema también ocurre en el caso que eliminemos toda la columna.

Valores erróneos: Detección de valores anómalos:

Del mismo modo que para campos faltantes, para los campos erróneos o inválidos, hemos de distinguir entre la detección y el tratamiento de los mismos. La detección de campos erróneos se puede realizar de maneras muy diversas, dependiendo del formato y origen del campo. En el caso de datos nominales, la detección dependerá fundamentalmente de conocer el formato o los posibles valores del campo. Por ejemplo, si estamos tratando con un atributo con un formato fijo, como la matricula de un vehículo, podemos establecer si una determinada matricula es errónea si no se ajusta al formato los formatos permitidos en un cierto país. Parece evidente que aquellos datos erróneos que si se ajustan al formato serán mucho más difíciles (o imposibles) de detectar.

Resulta especialmente patente el hecho de que si los sistemas de información originales fueron diseñados con buenas restricciones de integridad (no permitiendo, por ejemplo, matriculas con un formato no correcto), todos estos campos erróneos no habrán podido introducirse. Este caso, obviamente, es mucho más aconsejable .En general, en estos casos, la detección de campos erróneos por el formato no es posible, ya que todos los datos se ajustan a los datos correctos, y nos podemos concentrar en otros tipos de errores.

En algunos casos, afortunadamente, es posible detectar campos erróneos por su contenido. Siguiendo con el ejemplo de las matriculas, podemos comparar (preferiblemente de una manera automática o semiautomática) los modelos de vehículos con las matriculas. Si las matriculas se numeran incrementalmente, podemos observar si alguna matricula supuestamente antigua está asociada a un modelo de vehículo muy moderno. Esta situación puede denotar un posible error

En el caso de la detección de valores erróneos en atributos numéricos, esta suele empezar por buscar valores anómalos, atípicos o externos (outliers), también llamados datos aislados, exteriores o periféricos. Es importante destacar que un valor erróneo y un valor anómalo no son lo mismo. Como hemos dicho, existen casos en los que los valores extremos se categorizar como anómalos estadísticamente pero son correctos, es decir, representan un dato fidedigno de la realidad. No obstante así, y todo puede ser un ejemplo, como las redes neuronales.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

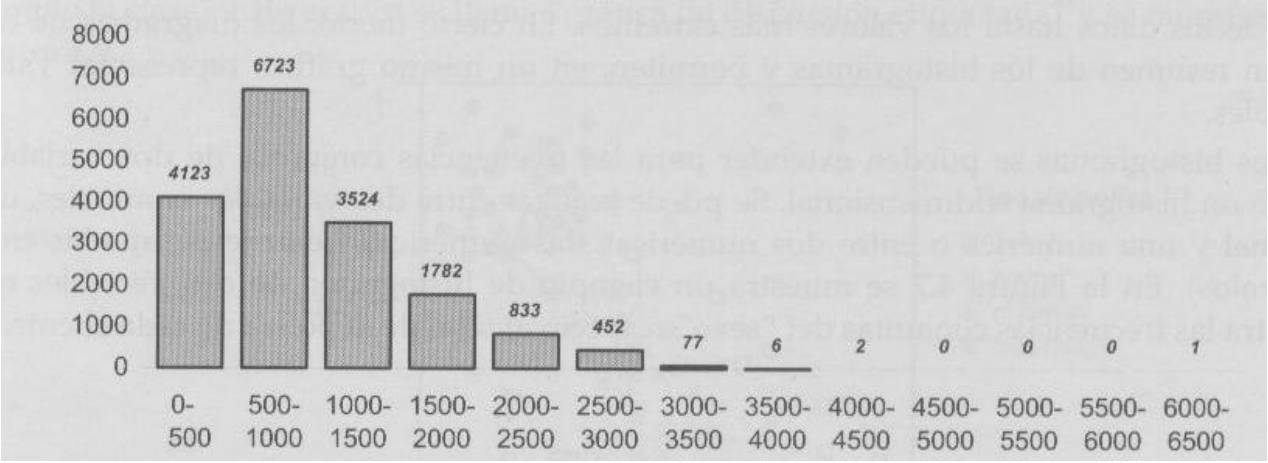


Figura 8: Histograma representando las frecuencias de un atributo numérico.

En el ejemplo de la **figura 8** vemos que existe un valor por encima de los 6000 euros, bastante lejano o aislado de la parte central de la "campana". Dicho valor puede ser determinado como un valor anómalo desde el punto de vista estadística. En general, existen herramientas estadísticas (basadas en test de discordancia) que sugieren los valores anómalos (bajo la distribución asumida o sugerida) y es el usuario el que finalmente el que debe determinar si son errores o no.

Otras técnicas para la detección de valores anómalos consiste en definir una distancia media al resto de individuos (individuo cuyo vecino más próximo o cuyos K-vecinos más próximos están más lejos). O bien donde una fracción de los otros individuos está lejos. Esta es una manera más general que los histogramas y se ha desarrollado en gran medida últimamente.

Esto se puede llevar todavía mucho más allá. En primer lugar, podemos utilizar técnicas de clustering parciales (no todos los elementos caen dentro de un grupo) Aquellos elementos aislados que no entren en un grupo o aquellos grupos minoritarios pueden ser considerados valores extremos. También se puede realizar una detección por predicción. Es decir, se realiza un modelo de clasificación o regresión (dependiendo si el atributo es no minal o numérico) y se predice el valor del atributo para cada ejemplo. Aquellos ejemplos cuya predicción este mas lejos del valor que existe en la instancia (se puede utilizar un clasificador suave, en el caso de valores nominales) puede denotar un error al ser inspeccionado.

Antes de pasar al tratamiento de valores anómalos es importante destacar que no detectar un valor anómalo puede ser un problema importante si el atributo se normaliza posteriormente (entre cero y uno por ejemplo), ya que la mayoría de datos estarán en un rango muy pequeño (entre 0 y 0.1) y puede haber poca precisión o sensibilidad para algunos métodos de minería de datos.

Finalmente, llegamos al tratamiento de valores anómalos o erróneos. Los tratamientos son muy similares a los datos faltantes:

- **Ignorar (dejar pasar):** algunos algoritmos son robustos a datos anómalos (por ejemplo árboles de decisión).

- **Filtrar (eliminar o remplazar) la columna:** solución extrema, pero a veces existe otra columna dependiente con datos de mayor calidad. Preferible a eliminar la columna es remplazarla por una columna discreta diciendo el valor era normal u erróneo (por encima o por debajo). En el caso de los anómalos se puede sustituir por no anómalo, anómalo superior o anómalo inferior.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Filtrar la fila:** pude sesgar los datos, porque muchas veces las causas de un dato erróneo están relacionadas con casos o tipos especiales.

- **Reemplazar el valor:** por el valor 'nulo' si el algoritmo lo trata bien o por máximos o mínimos, dependiendo por donde es anómalo, o por medias

- **Discretizar:** transformar un valor continuo en un valor discreto (por ejemplo muy alto, alto, medio, bajo o muy bajo) hace que los anómalos caigan en 'muy alto' o 'muy bajo' sin mayores problemas.

Los atributos erróneos son mucho más graves cuando afectan a un atributo que va a ser utilizado como clase o como valor de salida de una tarea predicativa de minería de datos.

Transformación de los datos

La transformación de datos engloba, en realidad, cualquier proceso que modifique la forma de los datos. A continuación vamos a ver las diferentes operaciones de transformación de atributos:

Reducción de la dimensionalidad por transformación:

La alta dimensionalidad es, muchas veces, un gran problema a la hora de aprender de los datos. Si tenemos muchas dimensiones (atributos) respecto a la cantidad de instancias o ejemplos, nos encontremos con una situación poco deseable al existir tantos grados de libertad, los patrones extraídos pueden ser caprichosos y poco robustos. Visualmente tenemos un "espacio" prácticamente vació y, por tanto los patrones no tienen datos donde apoyarse a la hora de tomar una u otra forma. Este problema se conoce popularmente como la "la maldición de la dimensionalidad" ("the curse of dimensionality"). Aunque este es el problema mayor y es motivación suficiente para intentar reducir la dimensionalidad, especialmente se tienen pocos ejemplos, existen otros problemas, como por ejemplo, la eficiencia y la dificultad de representación de los datos (en principio, solo podemos representar tres dimensiones).

La reducción de dimensionalidad se puede realizar por selección de un subconjunto de atributos, como veremos en el capitulo siguiente, o bien se puede realizar por transformación, sustituyendo el conjunto de atributos iniciales por otros diferentes, que, geométricamente, se denomine proyección.

Existen muchas técnicas para realizar este tipo de proyección (o que pueden ayudar en este propósito): análisis de las componentes principales (PCA), algunas técnicas de análisis factorial, uso de mapas auto-organizativos, etc.

Análisis de las componentes principales (PCA):

La técnica más tradicional, conocida y eficiente para reducir la dimensionalidad por transformación se denomina "análisis de las componentes principales" (del ingles, "principal components analysis", PCA, también llamada "método Karhumen-Loeve") y consiste en trasformar los atributos o variables originales $x_1,x_2,x_3…x_m$ de los ejemplos en otro conjunto de atributos $f_1,f_2,f_3…$, f_p donde p <= m. Este proceso geométrico se puede ver como un cambio de ejes en la representación (proyección).

Lo interesante de esta transformación es que los nuevos atributos se generan de tal manera que sean independientes entre sí y, además, los primeros fs tengan más relevancia (más contenido informacional) que los últimos. Es decir, f_1 es más relevante que f_2, f_2 es más relevante que f_3 y así sucesivamente. Esto

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

permite seleccionar los k-primeros atributos para diferentes valores de k. Por ejemplo, podemos elegir k=3 cuando queramos representar los datos en un grafico tridimensional o podemos elegir un k mayor o menor según la información que queramos ignorar. La transformación nos asegura que si ignoramos los últimos p-k atributos, estaremos descartando la información menos relevante.

Como hemos comentado el análisis de las componentes principales se suele incluir junto a otras técnicas, en lo que se denomina análisis multivariante (multivariante analysis). De hecho, como el resto de estas técnicas se basan en conceptos algebraicos. Veamos de una manera sencilla en qué consiste el análisis de componentes principales.

El objetivo es convertir un vector orinal $x_i,x_2,..,x_m$ de los ejemplos en otro vector $f_1,f_2,...,f_p$. Donde p <= m. Par ello se computa el PCA usando el método de las covarianzas el cual se explica detalladamente en esta web: http://www.worldlingo.com/ma/enwiki/es/Principal_components_analysis

Aumento de la dimensionalidad por transformación o por construcción

Aunque hay muchos casos donde, razonablemente puede, parecer que reducir la dimensionalidad es positivo, podemos ver otros casos donde nos interesa aumentar el número de atributos. Incluso, hablaremos en muchos casos de generar nuevos atributos que son combinación de otros y, en cierto modo, son atributos redundantes.

Aumento de la dimensionalidad mediante núcleos:

Si en el apartado anterior hemos visto muchos métodos que proyectan los atributos originales en otro conjunto de atributos nuevos y diferentes, con la propiedad de que el numero de atributos después de la proyección sea mayor que el inicial, también se puede hacer una transformación similar en la que el numero de atributos después de la proyección sea mayor que el original.

El objetivo, en estos casos, es aprovecharse de la "maldición de la dimensionalidad". Como no hay mal que por bien no venga, si aumentamos la dimensionalidad conseguimos como hemos dicho anteriormente, que los datos se separan en el espacio, facilitando, por ejemplo, fronteras lineales donde antes no las había. Esto quiere decir que, si realizamos un aumento de dimensionalidad adecuado, podemos convertir algunos problemas no lineales o incluso irresolubles en problemas lineales, al "aclararse" el espacio.

El concepto fundamental en este tipo de aumento de dimensionalidad es la utilización de funciones de núcleo (Kernels), que permiten realizar estas proyecciones de manera adecuada.

Creación de características:

La creación, o agregación, de características consiste en crear nuevos atributos para mejorar la calidad, visualización o comprensibilidad del conocimiento extraído. La mayoría o todos los atributos originales se preservan, con lo que hablamos de añadir nuevos atributos y no de sustituirlos, como en el caso anterior (creación de características). Este proceso recibe muchos nombres: **construcción, creación o descubrimiento de características**, inducción constructiva…

La importancia de añadir atributos se muestra patente cuando existen patrones complejos en los datos que no pueden ser adquiridos por el método de minería de datos utilizado. Por ejemplo, supongamos que tenemos datos de la evolución de la apertura de pólizas en los últimos 21 meses. Queremos obtener el modelo de datos para estimar futuros de contratos de pólizas. Los datos se muestran en la **figura 9** en forma de pequeños círculos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

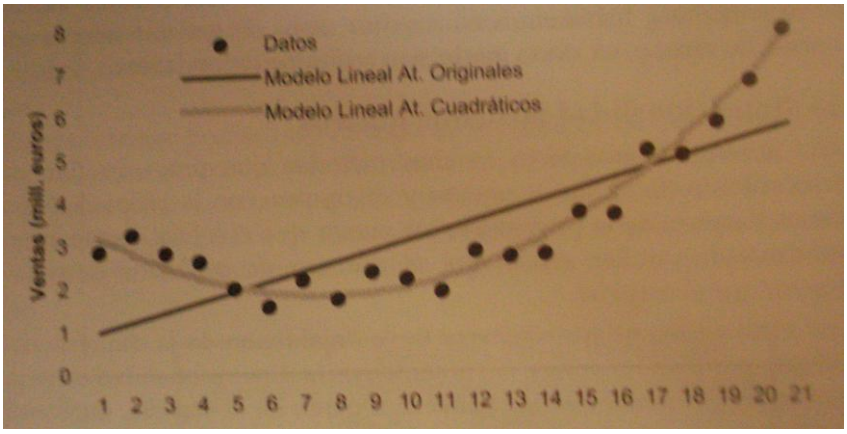


Figura 9: La importancia de crear características

Podemos intentar realizar un modelo de regresión lineal con una variable de entrada y una de salida como sigue:

$y = a_0 + a_1x$

El resultado del modelo obtenido ($a_0 = 1$, $a_1 = 2{,}2$) se muestra también en la **figura 9**. Este modelo no se ajusta demasiado bien, como se puede observar. ¿Deberemos utilizar redes neuronales debido a la no linealidad de los datos?

En muchos casos no hay que llegar tan lejos. Si sospechamos que en los datos hay componentes cuadráticos, podemos añadirlos. La idea es crear un nuevo atributo z = x2. Calculamos este atributo para todos los datos ya ahora volvemos a realizar el modelo de regresión lineal con dos variables de entrada y una de salida como sigue.

$y = a_0 + a_1x + a_2z$

El resultado de este nuevo modelo "lineal" obtenido ($a_0 = 3{,}2$, $a_1 = 3{,}2$, $a_2 = -4{,}3$) se neutra también en la **figura 9**. Como se puede observar el modelo con componentes cuadráticos se ajusta mucho mejor a los datos.

En el ejemplo anterior, un simple atributo adicional, como el cuadrado de un atributo original, ha sido suficiente para ajustar un buen modelo con una técnica relativamente sencilla. En general, sin embargo disponemos de muchos atributos (tanto nominales como numéricos) y de métodos de aprendizaje más sofisticados.

Vamos a dar una serie de pautas para facilitar esta tarea, aunque, en muchos casos, la creación de buenas características o atributos depende no solo de los propios datos, sino también del conocimiento que se tenga de los datos y del método de minería de datos que se vaya a utilizar. De hecho, a veces se categorizar las técnicas según estén guiadas por los datos (data-driven), guiadas por la hipótesis o el modelo (hipótesis-driven), o guiadas por el conocimiento (Knowledge-driven).

En primer lugar, nos podemos preguntar qué tipos de combinaciones podemos hacer. La respuesta no puede ser más tangente: cualquiera que nos imaginemos. Lógicamente, se puede utilizar combinaciones muy complejas. Afortunadamente, es más útil (y más comprensible) el uso de relaciones simples ente los atributos. Dependiendo de si son atributos nominales o numéricos podemos distinguir las técnicas:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Atributos numéricos:** se utilizan, generalmente, operaciones matemáticas básicas de uno o más argumentos: suma, resta, producto, división, máximo, mínimo, media, cuadrado, raíz cuadrada, etc. Todas ellas retornan un valor numérico. También se pueden generar valores nominales a partir de valores numéricos, por ejemplo crear una variable lógica que indique si un determinado valor numérico es mayor o menor que un determinado valor o que otro atributo.

- **Atributos nominales:** se utilizan generalmente operaciones lógicas: conjunción, disyunción, negocio, implicación, igualdad o desigualdad. Todas ellas retornan un valor nominal.

Discretizacion y numerizacion

Uno de los aspectos más importantes, sino el que más, de un atributo es el tipo. El hecho de que un atributo sea nominal o numérico determina en gran medida como va a ser tratado por las herramientas de minería de datos. En algunas situaciones puede ser conveniente un numérico a nominal (discretizacion) o viceversa (numerizacion). Empecemos por la discretizacion.

Discretizacion:

La discretizacion, o cuantizacion es la conversión de un valor numérico en un valor nominal ordenado. Por ejemplo convertir una nota o calificación de 0 a 10 en una serie de valores ordenados { suspenso(0:4,99), aprobado(5,6,99), notable(7:8.49), sobresaliente(8.5,9,99), matrícula de honor (10) } es un ejemplo de discretizacion.

La discretizacion, en general, se realiza cuando el error en la medida puede ser grande o existen ciertos umbrales organizativos. El caso de las notas es, probablemente, el más ilustrativo: la diferencia numérica entre 4.99 y 5 es mínima y equivalente a la diferencia numérica entre 3.99 y 4.00. Sin embargo, la primera diferencia (según el sistema de calificaciones tradicional en España) significa pasar de un suspenso a un aprobado. Otra razón importante de la discretizar es la integración de escalas diferentes. Por ejemplo en ciertos países de latino América como por ejemplo, El Salvador, el aprobado empieza en 6 y no en 5, como en España. En el caso de tener notas de varios países sería preferible integrarlas todas en algún tipo de escala común. Una escala discreta es, en este caso, más sencilla de realizar.

Existen más razones para discretizar como, por ejemplo, el hecho de que las diferencias en ciertas zonas del rango de valores sean más importantes que en otras, o dicho mas técnicamente, cuando la interpretación de la medida no sea lineal. Por ejemplo, si tomamos las medidas de luz en una casa domotica, nos puede interesar más que tener una escala discreta: {Oscuro (0:170), penumbroso (170:200), luminoso (200:1000)}, que puede ser mas practico de cara a la obtención de modelos.

Una última razón y, quizás la más obvia, es que cuando tenemos algunos atributos nominales y otros numéricos, y queremos que todos sean nominales para, por ejemplo, establecer reglas de asociación.

Numerizacion:

La numerizacion es el proceso inverso a la discretizacion. Aunque es menos común que la discretizacion, también existen casos donde puede ser extremadamente útil. Un primer caso obvio donde la numeración es útil es cuando el método de minería de datos que vamos a utilizar no admite datos nominales. Un claro ejemplo es si pretendemos utilizar regresión lineal para obtener la influencia de cada "factor". En general, es preciso para muchos de los métodos de modelización estadística, como la regresión logística (o multidimensional), el análisis ANOVA, los modelos log-linear, los discriminantes parametritos o no parametritos, etc.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En todos estos casos lo que se suele hacer es lo que se denomina numerizacion **"1 a n"** que es la creación de varias variables indicadoras, flan o dummy. Si una variable nominal x tiene posibles valores $\{a_1, a_2,..,a_n\}$ creamos n variables numéricas, con valores 0 a 1 dependiendo de si la variable nominal toma ese valor o no. En general, esta aproximación genera más variables de las necesarias, ya que si tenemos $a_1 =0$, $a_2 = 0,\ldots$, $a_{n-1} =0$ sabemos con certeza que $a_n=1$. Por tanto es suficiente generar n-1 variables numéricas en vez de n. Este se denomina numerizacion "1 a n-1".

En algunos casos es posible identificar un cierto orden o magnitud en los valores de un atributo nominal y entonces podemos realizar una numerizacion denominada numerizacion "1 a 1". Por ejemplo, si tenemos categorías del estilo {niño, joven, adulto, anciano} podemos numerar los valores de 0 a 3, por ejemplo.

Normalización de rango: escalado y centrado

En general, muchos métodos de minería de datos no es necesario centrar los datos ni escalar, es decir normalizar el rango de un valor numérico. De hecho, cuando se usan métodos como los arboles de decisión o los sistemas de reglas, que son comprensibles, escalar previamente hacen que los modelos resultantes sean más difíciles de interpretar (hay que invertir el escalado final).

Sin embargo, en otros casos, es especial para muchas técnicas vistas anteriormente (análisis multivariante, por ejemplo) es necesario normalizar todos los atributos al mismo rango. Por ejemplo, antes se ha visto que es preciso normalizar a la misma medida cuando el mismo atributo proviene de distintas fuentes (euros, pesetas). En otras técnicas, como el análisis de componentes principales, se requiere normalizar el rango entre 0 y 1. También es necesario en la mayoría de los algoritmos basados en distancias, ya que las distancias debidas a diferencias de un atributo que van entre 0 y 100 serán mucho mayores que las distancias debidas a diferencias de un atributo que va entre 0 y 10.

La normalización más común es la **normalización lineal uniforme** y se normaliza a una escala genérica entre 0 y 1 utilizando la siguiente formula.

$$v' = v\text{-}min / (max - min)$$

El resultado de esta normalización es la relación (el cociente) entre valores se mantiene. Para realizar esta normalización solo es necesario conocer el máximo y el mínimo de los valores dados para este atributo.

Una vez los datos están limpios y trasformados se procede a la selección de datos

Selección de datos:

La selección de datos es algo más que decir que tablas (o archivos, en su caso) se va a necesitar para la minería de datos y de qué manera concatenarlas. Esto podría estar ya decidido pero todavía no sabemos qué atributos/variables necesitamos y cuantas instancias (ejemplos) van a ser necesarias. El problema existente es precisamente que si seleccionamos como "vista minable" todo aquello que pueda ser relevante podemos acabar con una vista minable de cientos de columnas/atributos y millones de filas/registros.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El tamaño de una tabla como esta desborda la capacidad de muchas de las técnicas de minería de datos. Por ello hemos de ser capaces de ver si podemos obtener unos primeros modelos (o incluso mejores modelos) con un subconjunto de las instancias y de las variables.

La selección de datos no tiene únicamente como objetivo la reducción del tamaño `para obtener una minería de datos más rápida sino que, si no que en muchos casos puede permitir mejorar el resultado (tanto en precisión o coste).

El proceso de selección de datos muchas veces se engloba dentro de un concepto más amplio, denominado reducción de datos (data reducción), aunque este término puede englobar también la agregación, la generalización o incluso la comprensión de datos.

En general, cuando tratamos de datos del estilo atributo-valor (es decir una tabla) hay dos tipos de selección aplicables: selección horizontal (muestreo), donde se eliminan algunas filas (individuos) y selección vertical (reducción de la dimensionalidad), donde se eliminan características de todos los individuos. Pasemos a ver estos dos tipos de selección:

Técnicas de muestreo:

La manera más directa de reducir el tamaño de una población o conjuntos de individuos es realizar el muestreo. Existen varias razones para realizar un muestreo sobre los datos disponibles (ya sean, a su vez, un muestreo o el total de la población). Entre las razones están, principalmente, la reducción del tamaño (con el objetivo de facilitar y agilizar los algoritmos de minería de datos), pero existen muchas otras (balanceado de clases, cobertura equilibrada del espacio, facilitar visualización…).

Dependiendo de estas razones existen varios tipos de muestreo: aleatorio, estratificado, por grupos o exhaustivo. Veamos con detalle estos tipos de muestreo:

- **Muestreo Aleatorio Simple.** La premisa de este muestreo es que instancia tiene la misma probabilidad de ser extraída en la muestra. Puede ser con o sin reemplazamiento. La manera más sencilla de realizar un muestreo sin reemplazamiento de este estilo es asignar un número aleatorio a cada instancia y después ordenarlas por ese valor. Se seleccionan los n primeros y tenemos una muestra de n instancias. Para realizar un muestreo con reemplazamiento, una manera sencilla es, dado un total de m instancias generar aleatoriamente un valor i entre 1 y m. Con este valor i se elige la i-exima instancia del conjunto. Esta operación se repite hasta tener n valores.

- **Muestreo aleatorio estratificado:** El objetivo de este muestreo es obtener una muestra balanceada con suficientes elementos de todos los estratos, o grupos. Lógicamente, para poder hacerlo es necesario conocer los estratos o grupos de interés. Esto generalmente ocurre con problemas de clasificación, donde estos estratos son precisamente las clases existentes, y puede ser que tengan pocas instancias de unas clases o muchas instancias de otras. No obstante, se puede plantear realizar un agrupamiento previo al muestreo de descubrir los estratos. Sea como sea el muestreo aleatorio estratificado se puede realizar de muchas maneras, dependiendo del número de elementos que se quiera para cada estrato. Por ejemplo, si existen suficientes elemento de todos los estratos y queremos n elementos de cada estrato, realizaremos un muestreo aleatorio simple sin reemplazamiento de cada estrato hasta obtener los n elementos de cada estrato. Finalmente tendremos n*e donde e es el numero de estratos. Sin embargo, muchas veces no tenemos suficientes objetos de todos los estratos. En esta situación lo mejor es realizar un muestreo aleatorio simple con reemplazo de aquellos estratos minoritarios. Como resultado para algunos estratos tendremos más instancias de las que había originalmente en el

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

estrato (sobremostreado) y para otros estratos tendremos menos instancias de las que había originalmente en el estrato (submuestreo). Las ventajas del muestreo estratificado no es solo que se cubren suficientes estancias de todos los estados, sino que además es estadísticamente aconsejable, especialmente cuando se realizan submuestreo, ya que los elementos del mismo estrato suelen ser similares y, por tanto, de alguna manera, se mantiene más diversidad en la muestra. Dicho de otra manera, es preferible pocos elementos, pero de todos los estratos, a tener muchos, pero solo de algunos estratos:

- **Muestreo de grupos:** En cierto modo inverso al anterior, el muestreo de grupos consiste en elegir solo elementos de unos grupos. El objetivo de este muestreo es generalmente descartar ciertos grupos que, por diversas razones, pueden impedir la obtención de buenos modelos.

- **Muestro exhaustivo:** Se trata de una exageración del muestreo estratificado, con motivaciones similares. Se procede de la siguiente manera para los atributos numéricos (normalizados) se genera al azar un valor en el intervalo posible; para los atributos nominales se generara al azar un valor entre los posibles. Con esto obtenemos una instancia ficticia. El quid del procedimiento es que ahora buscamos la instancia real más similar a la ficticia. Se repite este proceso hasta tener n instancias. Nótese que el objetivo de este método es cubrir completamente el espacio de instancias y evitar poner muchos ejemplos de las zonas muy densas. Lógicamente puede haber zonas donde no haya instancias, pero buscaremos de los alrededores la más cercana. Existen variantes de este tipo de muestreo, como por ejemplo el muestreo por cuadricula, en el que las instancias ficticias no se calculan aleatoriamente sino como una cuadricula, en el que las instancias ficticias no se calculan aleatoriamente sino como una cuadricula o grid sobre el espacio de instancias.

Finalmente vistos los tipos de muestreo existentes, no queda la pregunta. ¿Cuántos datos son necesarios? En el caso de validación de hipótesis existen técnicas estadísticas para determinar cuantos datos son necesarios para tener un nivel de confianza preestablecido. No obstante, si el objetivo es determinar cuantos datos son necesarios para el entrenamiento de un modelo, la respuesta es más difícil.

Depende en general, del número de “grados de libertad” que, a su vez, depende del número de atributos. Además, depende del método de aprendizaje y de su expresividad.

Selección de características relevantes: reducción de la dimensionalidad:

La selección de características, variables o atributos (feature selection) tiene cuatro objetivos principales. En primer lugar, permite reducir el tamaño de los datos, al eliminar características o atributos de todos los ejemplos que puedan ser irrelevantes o redundantes. En segundo lugar, una buena selección de características puede mejorar la calidad del modelo, al permitir al método de minería de datos centrarse en las características relevantes. En tercer lugar, una buena selección de características permite expresar el modelo resultante en función de menos variables; esto es especialmente importante cuando se desean modelos comprensibles (árboles de decisión, regresión lineal, etc.). En cuanto lugar, se puede requerir una reducción de dimensionalidad a dos o tres características exclusivamente, con el propósito de representar los datos visualmente.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Existen dos reglas generales para eliminar características:

- **Eliminación de claves candidatas:** la regla general es eliminar cualquier atributo que pueda ser clave primaria de la tabla (o que sea clave candidata o incluso parte de clave candidata, parcial o totalmente).Por ejemplo, hay que eliminar números de documentos de identificación códigos internos, nombres y apellidos, direcciones.

- **Eliminación de atributos dependientes:** en la teoría de la normalización de bases de datos, cuando existen dependencias funcionales entre atributos se intenta normalizar en varias tablas... Por ejemplo, cuando se tiene el código postal, la ciudad, la región, con la región tenemos el país, con lo que podemos establecer una serie de dependencias funcionales que se deben normalizar en cuantas tablas sean necesarias.

Aunque determinar los atributos anteriores (claves candidatas y atributos dependientes) no es sencillo en muchos casos, es mucho más fácil que determinar otras características, que tampoco son relevantes o redundantes. En general una vez detectados los atributos irrelevantes y redundantes, debemos utilizar técnicas más sofisticadas si queremos seguir reduciendo la dimensionalidad, en particular para los atributos numéricos. Existen dos tipos generales de métodos de selección de características (atributos):

- **Métodos de filtro o métodos previos:** se filtran los atributos irrelevantes antes de cualquier proceso de minería de datos y, en cierto modo, son fundamentalmente estadísticas (medidas de información, distancia, dependencia o inconsistencia). El criterio para establecer el subconjunto de características "Optimo" se basa en medidas de calidad previa que se calcula a partir de los mínimos datos.

- **Métodos basados en modelo o métodos envolventes (wrapper):** la bondad de la selección de atributos se evalúa respecto a la calidad de un método de minería de datos o estadístico extraído a partir de los datos (utilizando, lógicamente, algún método de validación). Lógicamente, este tipo de técnicas requieren mucho más tiempo que las otras, ya que para evaluar hay que haber entrenado un modelo. Nótese que el método de minería de datos utilizado para hacer la selección de atributos no tiene porque ser el método de minería de datos que se utilizara finalmente. Por ejemplo se puede utilizar una red neuronal para determinar los atributos más significativos y, una vez determinados, utilizar un árbol de decisión para obtener un modelo comprensible utilizando solo las características seleccionadas para generar las reglas del modelo.

Sea de filtro o basados en modelo, ambos modelos pueden ser iterativos, es decir, se van eliminado atributos y se va viendo el resultado. Se va recuperando o eliminando atributos de una manera iterativa, hasta que se obtiene una combinación que maximiza la calidad. Existen muchas maneras de realizar este procedimiento, por ejemplo, empezando con un atributo y elegir el que de mayor calidad de selección con atributos, después añadir el atributo que de mayor calidad de selección con dos atributos, y así hasta que no se mejore la calidad o se llegue al número deseado de atributos (estrategia forward). La manera inversa (comenzar con todos e ir eliminado) o maneras mixtas pueden también ser utilizadas.

Ejemplo de selección de atributos:

Vamos a realiza un ejemplo de selección de atributos utilizando la herramienta WEKA. Para ello vamos a coger un conjunto de datos que recoge información (16 atributos: ocho nominales y ocho numéricos) sobre acuerdos sindicales, indicando si estos fueran buenos o malos (la clase).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El conjunto solo recoge 57 instancias, por lo que, para poder extraer un modelo fiable sería interesante realizar una selección de atributos previa para aumentar la proporción entre el numero de ejemplos y el numero de atributos. En este ejemplo vamos a analizar la relevancia respecto a la clase, es decir vamos a intentar identificar un subconjunto de atributos que estén fuertemente relacionados con la clase y además poco relacionados entre sí.

El sistema WEKA incorpora una gran cantidad de métodos para estudiar la relevancia de atributos y realizar una selección automática de los mismos. Para utilizar estos métodos, una vez cargados los datos, es necesario elegir uno de los métodos de selección. Existen una serie de evaluadores, entre los que, además de los componentes principales, hay dos tipos: "SubSetEval" (evaluadores de subconjuntos o selectores) y "AttributeEval" (prorrateadotes de atributos). Los primeros necesitan elegir un método o estrategia de búsqueda de los subconjuntos ("Search Method"), como los que hemos vistos antes (forward o backward), además de una estrategia exhaustiva (prueba todos los subconjuntos posibles) y otras. Los segundos solo pueden combinarse con un "Ranker", ya que no seleccionan atributos sino que los ordenan por relevancia.

Veamos cómo se comportan distintos métodos de selección con este conjunto de datos. La siguiente tabla muestra una serie de métodos y los atributos seleccionados por ellos.

Método de selección	Filtro/Modelo	Opciones	Estrategia de búsqueda	Atributos seleccionados
CfsSubSetEval	Filtro	Valores por defecto	"ExhaustiveSearch"	2,3,11,13
WrapperSubSetEval	Modelo	Modelo j4.8.j4.8, valores por defecto	"ExhaustiveSearch"	Muy lento
			"ForwardSelection"	2,11
			"BestFirst"	2,11
			"GeneticSearch"	3,6,7,13
			"RankSearch"	2,3,11
		Modelo LogisticRegression, valores por defecto	"ForwardSelection"	2,6,10,13
			"GeneticSearch"	3,6,8,10,13,14
			"RankSearch"	2,3,11,13
ConsistentSubSetEval	Filtro	Valores por defecto	"ExhaustiveSearch"	2,3,10,16

Tabla 2: Selecciones realizadas con WEKA con métodos "SubSetEval"

Como se puede observar, se han utilizado métodos de filtro y métodos que utilizan modelo (envolventes, wrappers). Entre estos métodos de modelo, hemos elegido como modelo el J4.8 (la versión de WEKA del clasificador de árboles de decisión C 4.5 y la clasificación por regresión logística.

Para analizar mejor las selecciones anteriores vamos a utilizar ahora prorrateadotes de atributos (Para todos ellos hay que elegir un "Ranker" como estrategia de búsqueda). En la **tabla 3** se muestran el orden de los 16 atributos según varios métodos de identificación de la relevancia de atributos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Método de evaluación	Filtro / Modelo	Estrategia de búsqueda	Orden de los atributos
RelieFAttributeEval	Filtro	"Ranker"	14,2,3,12,16,4,7,13,11,9,8,6,10,5,15,1
InfoGainAttributeEval	Filtro	"Ranker"	2,3,11,14,16,12,13,9,7,5,15,4,10,8,1,6
GainRatioAttributeEval	Filtro	"Ranker"	2,3,11,13,14,16,9,15,12,7,5,4,10,8,1,6
SymUncertAttributeEval	Filtro	"Ranker"	2,3,11,14,16,13,12,9,15,7,5,4,10,8,1,6
OneRAttributeEval	Filtro	"Ranker"	13,11,16,7,10,2,3,5,14,9,15,6,12,4,1,8
ChiSquaredAttributeEval	Filtro	"Ranker"	2,3,11,14,16,12,13,9,7,5,15,4,10,8,1,6
WrapperSubSetEval	Modelo j4.8.j4.8	"ForwardSelection" con generateRanking="true"	2,11,1,10,8,4,5,3,15,12,6,9,13,7,16,14
WrapperSubSetEval	Modelo Logistic	"ForwardSelection" con generateRanking="true"	2,13,10,6,15,5,4,3,7,9,14,16,1,8,11,12

Tabla 3: Selecciones realizadas con WEKA con métodos "AttributeEval"

A la vista de las dos tablas tenemos bastante información para tomar una decisión a la hora de seleccionar los atributos más relevantes, Los atributos 2 y 3, que representan los incrementos salariales en los primeros años de contrato son los dos atributos más relevantes para llegar a un buen acuerdo, ya que aparecen en casi todas las selecciones y aparecen en primer lugar en la mayoría de los rankings.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Fase de Minería de datos

La fase de minería de datos es la más característica del KDD y, por esta razón, muchas veces se utiliza esta fase para nombrar todo el proceso. El objetivo de esta fase es producir nuevo conocimiento que pueda ser utilizado por el usuario. Esto se realiza construyendo un modelo basado en los datos recopilados para este efecto, El modelo es una descripción de lo basado en los datos recopilados para este efecto. El modelo es una descripción de los patrones y relaciones entre los datos que pueden usarse para hacer predicciones, para entender mejor los datos o para explicar situaciones pasadas. Para ello es necearlo tomar una serie de decisiones antes de empezar el proceso:

- Determinar que tarea de minería de datos es la más apropiada. Por ejemplo, podrimos utilizar la clasificación para predecir en una entidad bancaria los clientes que dejaran de serlo.

- Elegir el tipo de modelo: Por ejemplo, para una tarea de clasificación podríamos utilizar un árbol de decisión, porque queremos obtener un modelo en forma de reglas.

- Elegir el algoritmo (técnica) de minería que resuelvas la tarea y obtenga el tipo de modelo que estamos buscando. Esta elección es pertinente porque existen muchos métodos para construir los modelos. Por ejemplo, para crear árboles de decisión para clasificación podríamos usar CART o C5.0, entre otros...

Tareas de la minería de datos

En primer lugar para, definir las tareas debemos definir el conjunto de ejemplos con los que se va a tratar. Definamos E como el conjunto de todos los posibles elementos de entrada. Las instancias posibles dentro de E como el conjunto de todos los posibles elementos de entrada. Las instancias posibles dentro de E generalmente se representan como un conjunto de valores para una serie de atributos. Es decir E = $A_1*A_2*A_3..*A_n$ y un ejemplo e es una tupla <a_1, a2,..a_n> tal que a_i pertenece a A_i.

Veamos a continuación las tareas más importantes de la minería de datos.

- **Predictivas:** se trata de problemas y tareas en las que hay uno o más valores para cada uno de loes ejemplo. Los ejemplos en la evidencia van acompañados de una salida (clase, categoría) o un orden entre ellos. Dependiendo de cómo sea la correspondencia entre los ejemplos y los valores de salida y la presentación de los ejemplos podemos definir varias tareas predictivas:

    - **Clasificación:** los ejemplos se presentan como un conjunto de pares de elementos de dos conjuntos, δ= {<e,s>: e pertenece a E, s pertenece a S}, donde S es el conjunto de valores de salida. El objetivo es aprender una función λ: E -> S, denominada clasificador, que represente la correspondencia existente en los ejemplos, es decir, para cada valor de E un único valor de S. Además S, es nominal, es decir, puede tomar un conjunto de valores c_1,c_2, .. , con denominados clases.

    - **Clasificación suave:** la representación del problema es la misma que la de la clasificación. Además de la función λ: E -> S se aprende otra función θ: E -> R que significa el grado de certeza de la predicción hecha por la función λ.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Categorización:** no se trata de aprender una función, sino una correspondencia. Es decir, cada ejemplo de δ = {<e,s>: e pertenece a E, s pertenece a S}, así como la correspondencia a aprender λ: E -> S, puede asignar varias categorías a un mismo e, a diferencia de la clasificación, que solo asigna una y solo una. Dicho de otra manera, un ejemplo podría tener varias categorías asociadas.

- **Regresión:** es quizás la tarea más sencilla de definir. El conjunto de evidencias son correspondencias entre dos conjuntos δ: E -> S, donde S es el conjunto de valores de salida. El objetivo es aprender una función λ: E -> S que represente la correspondencia existente en los ejemplos, es decir, para cada valor de E tenemos un único valor de S. la diferencia con respecto a la clasificación es que S es numérico, es decir puede ser un valor entero o real.

- **Descriptivas:** los ejemplos se presentan como un conjunto δ = {e: e pertenece a E}, sin etiquetar ni ordenar de ninguna manera. El objetivo, por tanto, no es predecir nuevos datos si no describir los existentes. Lógicamente, esto se puede hacer de muchas maneras y la variedad de tareas se dispara. A continuación vamos a ver las tareas descriptivas más importantes.

  - **Agrupamiento (clustering):** la utilidad principal del agrupamiento es averiguar que las instancias se agrupan en varios segmentos es útil porque podemos determinar el comportamiento de una nueva instancia viendo a qué grupo de instancias pertenece. Es decir, si tenemos que una nueva instancia o ejemplo y es etiquetado en el grupo 3 significa que, posiblemente, en muchos aspectos, se comporte como el resto de ejemplos del grupo 3. Generalmente, posteriormente al agrupamiento, se toman decisiones diferentes para cada grupo. Por ejemplo, podemos agrupar los clientes en segmentos diferenciados, estudiar que grupos se comportan mejor ante determinados productos, y después orientar ciertos productos a ciertos grupos.

  - **Correlaciones y factorizaciones:** tratados con el objetivo de ver la relevancia de atributos, detectar atributos redundantes o dependencias entre atributos, o seleccionar un subconjunto, pueden considerarse también, una tarea de minería de datos. Los estudios de correlación y factoriales se centran exclusivamente en los atributos numéricos. El objetivo es ver si dos o más atributos numéricos A_i y A_j están correlacionados linealmente o relacionados de algún modo.

  - **Reglas de asociación:** esta es una tareas reinas de la minería de datos. El objetivo, en cierto modo es similar, a los estudios correlacionados y factoriales, pero para los atributos nominales, muy frecuentes en las bases de datos.

  - **Dependencias funcionales:** El descubrimiento de dependencias funcionales generalmente se suele incluir dentro de la variedad de tareas con el nombre de “reglas de asociación”. En realidad las dependencias funcionales considera todos los posibles valores... Se definen de la siguiente manera “Dados los valores de A_i, A_j,…, A_k puedo determinar el valor A_r” .Es decir el valor Ar depende o es función de los valores de ciertos atributos A_i, A_j,…,A_k.

  - Detección de valores e instancias anómalas: la detección de valores anómalos o atípicos (outlier detection), con el objetivo de hacer limpieza de datos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A continuación vamos a ver la correspondencia entre las tareas de minería de datos y los métodos (técnicas) de minería de datos.

Correspondencia entre tareas y métodos

Cada una de las tareas anteriores como, cualquier problema, requiere métodos, técnicas o algoritmos para resolverlas. Una de las cosas que sorprenden a los recién llegados a la minería de datos es que, además de que, lógicamente, una tarea puede tener muchos métodos diferentes para resolverla, tenemos que el mismo método (o al menos la misma técnica) puede resolver un gran abanico de tareas. Como veremos esto no es casualidad sino que se debe a que, en el fondo, la mayoría de las tareas que acabamos de ver son caras de la misma moneda, el aprendizaje inductivo.

Veamos brevemente los tipos de técnicas existentes para llevar a cabo las tareas anteriores:

- **Técnicas algebraicas y estadísticas:** se basan, generalmente, en expresar modelos y patrones mediante formulas algebraicas, funciones lineales, funciones no lineales, distribuciones o valores agregados estadísticos tales como medias, varianzas y correlaciones, etc. Frecuentemente, estas técnicas, cuando obtienen un patrón, lo hacen a partir de un modelo ya predefinido, se estiman unos coeficientes o parámetros, de ahí el nombre de técnicas paramétricas. Algunos de los algoritmos más conocidos dentro de este grupo de técnicas son la regresión lineal (global o local), la regresión logarítmica, y la regresión logística. Los discriminantes lineales y no lineales, basados en funciones predefinidas, es decir discriminantes paramétricos, entran dentro de esta categoría. No obstante, aunque el término “no parametrito” se utiliza para englobar gran parte de técnicas provenientes del aprendizaje automático, como son las redes neuronales, también existen muchas técnicas de modelización estadística no paramétrica.

- **Técnicas bayesianas:** se basan en estimar la probabilidad de pertenencia (a un grupo o clase), mediante la estimación de las probabilidades condicionales inversas o a priori, utilizando para ello el teorema de bayes. Algunos algoritmos muy populares son el clasificador bayesiano naive, los métodos basados en máxima verisimilitud y el algoritmo EM. Las redes bayesianas generalizan las topologías de las interacciones probabilísticas entre variables y permiten representar gráficamente dichas interacciones.

- **Técnicas basadas en árboles de decisión y sistemas de aprendizaje de reglas:** son técnicas que, además de representación en forma de reglas, se basan en dos tipos de algoritmos: los algoritmos denominados “divide y vencerás”, como el ID3/C4.5 o el CART, y los algoritmos denominados “separa y vencerás”, como el CN2.

- **Técnicas relacionales, declarativas y estructurales:** Las característica principal de este conjunto de técnicas es que representan los modelos mediante lenguajes declarativas, como los lenguajes lógicos, funcionales o lógico-funcionales. Las técnicas de ILP (programación lógica inductiva) son las más representativas y las que ha dado nombre a un conjunto de técnicas denominadas minería de datos relacional.

- **Técnicas basadas en redes neuronales artificiales:** se trata de técnicas que aprenden un modelo mediante el entrenamiento de los pesos que conectan un conjunto de nodos o neuronas. La topología de la red y los pesos de las conexiones determinan el patrón aprendido. Existen innumerables variantes de organización: perceptrón simple, perceptrón multicapa, redes de base radial, redes de Kohonen, etc., con no menos algoritmos diferentes para cada organización; el más conocido es el de retro propagación (backpropagation).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Técnicas basadas en núcleo y maquinas de soporte vectorial:** se tratan de técnicas que intentan maximizar el margen entre los grupos o las clases formadas. Para ello se basan en unas trasformaciones que pueden aumentar la dimensionalidad. Estas transformaciones e llaman núcleos (Kernels). Existen muchísimas variantes, dependiendo del núcleo utilizado y de la manera de trabajar con el margen.

- **Técnicas estocásticas o difusas:** bajo este paraguas se incluyen la mayoría de las técnicas que, junto a las redes neuronales y los algoritmos evolutivos y genéticos forman lo que se denomina computación flexible (soft computing).

- **Técnicas basadas en casos, en densidad o distancia:** son métodos que se basan en distancias al resto de elementos, ya sea directamente, como los vecinos más próximos (los casos más similares), de una manera más sofisticada, mediante la estimación de funciones de densidad. Además de los vecinos más próximos, algunos algoritmos muy conocidos son los jerárquicos, como Two-Step o COBWEB, y los no jerárquicos, como K-medias.

Además de todos los anteriores existen multitud de híbridos que dificultan más aun realizar una taxonomía razonable. En definitiva, es muy difícil realizar una taxonomía óptima que aglutine grupos de técnicas de una manera indiscutible y que, de alguna manera, no confunda tareas con modelos.

A titulo meramente ilustrativo, la siguiente tabla muestra algunas tareas (clasificación, regresión, agrupamiento, reglas de asociación, correlaciones) y algunas técnicas o algoritmos.

Nombre	PREDICTIVO		DESCRIPTIVO		
	Clasificación	Regresión	Agrupamiento	Reglas de asociación	Correlaciones / Factorizaciones
Redes neuronales	✓	✓	✓		
Árboles de decisión ID3, C4.5, C5.0	✓				
Árboles de decisión CART	✓	✓			
Otros árboles de decisión	✓	✓	✓	✓	
Redes de Kohonen			✓		
Regresión lineal y logarítmica		✓			✓
Regresión logística	✓			✓	
Kmeans			✓		
Apriori				✓	
Naive Bayes	✓				
Vecinos más próximos	✓	✓	✓		
Análisis factorial y de comp. ppales.					✓
Twostep, Cobweb			✓		
Algoritmos genéticos y evolutivos	✓	✓	✓	✓	✓
Máquinas de vectores soporte	✓	✓	✓		
CN2 rules (cobertura)	✓			✓	
Análisis discriminante multivariante	✓				

Tabla 4: Correspondencia entre tareas y técnicas:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Como se puede observar, la correspondencia entre tareas y técnicas es muy variada. Algunas tareas pueden ser resueltas por muy diversas técnicas y algunas técnicas pueden aplicarse para tres incluso cuatro tareas.

A continuación vamos a tratar la relación entre la minería de datos y el aprendizaje inductivo:

Minería de datos y el aprendizaje inductivo

Al considerar tantas tareas y métodos parece que estemos tratando problemas inconexos. Un aspecto chocante es que la misma técnica puede utilizarse para varias tareas. Esto quiere decir que hay algunos procesos útiles para una tarea que también lo es, generalmente, para otras, con una ligera adaptación. La pregunta, no obstante, nos asalta ¿qué tiene que ver un agrupamiento con una regresión? O si hablamos de métodos ¿Qué tiene que ver una red neuronal con un árbol de decisión?

La respuesta a estas preguntas consiste en reconocer que todas las tareas (exceptuando las reglas de asociación y correlaciones) y los métodos se centran alrededor de la idea de aprendizaje inductivo. Son, por decirlo de un modo coloquial, presentaciones diferentes del mismo proceso. Para entender esta respuesta debemos, primero, definir lo que es el aprendizaje inductivo. Veamos cuatro definiciones del aprendizaje:

- La visión más genérica define el aprendizaje como la mejora del comportamiento a partir de la experiencia [Mitchell 1997].

- Una visión más externa define el aprendizaje como la capacidad de predecir observaciones futuras con plausibilidad o explicar observaciones pasadas.

- Una visión más estática define el aprendizaje inductivo como la identificación de patrones, de regularidades, existentes en la evidencia.

- Una visión más teórica y matemática define el aprendizaje inductivo como eliminación de redundancia, visto como comprensión de Información [Solomonoff 1966].

Estas cuatro visiones se conjugas perfectamente de la siguiente manera: el aprendizaje nos permite identificar regularidades en un conjunto de observaciones. Estas regularidades son en realidad redundancias que pueden ser representadas por patrones y modelos que los compriman o que los definan. Estos patrones pueden ser utilizados para predecir observaciones futuras o explicar observaciones pasadas.

Las dos primeras definiciones son de aprendizaje, mientras que las dos últimas son de aprendizaje inductivo. El aprendizaje inductivo es un tipo especial de aprendizaje que parte de casos particulares (ejemplos) y obtienen casos generales (reglas o modelos) que abstraen o generalizan evidencias. Existen además otros tipos de aprendizaje, como por ejemplo el aprendizaje obductivo (o explicativo) y el aprendizaje por analogía, que van de casos particulares a casos particulares.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Fase de evaluación e interpretación

Medir la calidad de los patrones descubiertos por un algoritmo de minería de datos no es un problema trivial, ya que esta medida puede atañer varios criterios, algunos de ellos bastante subjetivos. Idealmente, los patrones descubiertos deben de tener tres cualidades: ser precisos, comprensibles (es decir, inteligibles) e interesantes (útiles y novedosos). Según las aplicaciones puede interesar mejorar algún criterio y sacrificar ligeramente otro, como en el caso del diagnostico medico que prefieren patrones comprensibles aunque su precisión no sea muy buena.

Técnicas de evaluación:

Tal y como hemos dicho, para entrenar y probar un modelo se parten los datos en dos conjuntos: el conjunto de entrenamiento (training set) y el conjunto de prueba o test (test set). Esta separación es necesaria para garantizar que la validación de la precisión del método es una medida independiente. Si no se usan conjuntos, diferentes de entrenamiento y prueba, la precisión del modelo será sobreestimada, es decir, tendremos estimaciones muy optimistas.

En los modelos predictivos, el uso de esta separación entre entrenamiento y prueba es fácil de interpretar. Por ejemplo, para cada tarea de clasificación, después de generar el modelo con el conjunto de entrenamiento, este se puede usar para predecir la clase de los datos de prueba (test). Entonces, la razón de precisión (o simplemente precisión), se obtiene dividiendo el número de clasificaciones correctas por el número total de instancias. La precisión es una buena estimación de cómo se comportara el modelo para datos futuros similares a los de test. Esta forma de proceder no garantiza que el modelo sea correcto, sino que simplemente indica que si usamos la misma técnica con una base de datos con datos similares a los de prueba, la precisión media será bastante parecida a la obtenida con estos.

El método de evaluación más básico, la validación simple, reserva un porcentaje de la base de datos como conjunto de prueba, y no lo usa para construir el modelo. Este porcentaje suele variar entre el 5% y el 50%. La división de los datos en estos dos grupos debe de ser aleatoria para que la estimación sea correcta.

Si tenemos una cantidad no muy elevada de datos para construir el modelo, puede que no podamos permitirnos el lujo de reservar parte de los mismos par la etapa de evaluación. En estos casos se utiliza un método llamado validación cruzada (Cross validation). Los datos se dividen aleatoriamente en dos conjuntos equitativos con los que se estima la precisión predicativa del modelo. Para ello, se construye un modelo con el primer conjunto y se usa para predecir los resultados en el segundo conjunto y calcular así un ratio de error (o precisión). A continuación, se construye un modelo con el segundo conjunto y se usa para predecir los resultados del primer conjunto, obteniéndose así su ratio del error. Finalmente, se construye un modelo con todos los datos, se calcula un promedio de los datos de error y se usa para estimar mejor su precisión.

El método que se usa normalmente es la validación cruzada con n pliegues (n-fold cross validation). En este método los datos se dividen aleatoriamente en n grupos. Un grupo se reserva para el conjunto de prueba y con los n-1 restantes (juntando todos sus datos) se construye un modelo y se usa para predecir el resultado de los datos del grupo reservado. Este proceso se repite n veces, dejando cada vez un grupo diferente para la prueba. Esto significa que se calculan n ratios de error independientes. Finalmente, se construye un modelo con todos los datos y se obtienen sus ratos de error y precisión promediando los n ratios de error disponibles.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Otra técnica para estimar el error de un modelo cuando se disponen de pocos datos es la conocida como bootstrapping. Este consiste en construir primero un modelo con todos los datos iniciales. Entonces, se crean numerosos conjuntos de datos, llamados bootstrap simples, haciendo un muestreo de los datos originales con reemplazo, es decir, se van seleccionando instancias del conjunto inicial, pudiendo seleccionar la misma instancia varias veces .Nótese que los conjuntos construidos de esta forma pueden contener datos repetidos. A continuación se construye un modelo con cada conjunto y se calcula su ratio de error sobre el conjunto de test (que son los datos sobrantes de cada muestreo).El error final estimado para cada el modelo construido con todos los datos se calcula promediando los errores obtenidos para cada muestra.

A continuación vamos a tratar las distintas medidas de evaluación de modelos.

Medidas de evaluación de modelos

Dependiendo de la tarea de minería de datos existen diferentes medidas de evaluación de modelos. Por ejemplo, en el contexto de la clasificación, lo normal es evaluar la calidad de los patrones encontrados con respecto a su precisión predictiva, la cual se calcula como el número de instancias del conjunto de prueba clasificadas correctamente dividido por el número de instancias totales en el conjunto de prueba. Tal y como hemos indicado el objetivo es obtener la mayor precisión posible sobre el conjunto de test, ya que obtener un 100% de precisión sobre el conjunto de entrenamiento es trivial, ya que bastaría con generar una regla para cada instancia usando una conjunción de sus variables-valores como antecedentes de una regla y el valor a predecir como consecuente.

En el caso de que la tarea sea de reglas de asociación, se suele evaluar de forma separada cada una de las reglas con el objetivo de restringirnos a aquellas que pueden aplicarse a un mayor número de instancias y que tienen una precisión relativamente alta sobre estas instancias. Esto se hace en base a dos conceptos:

- **Cobertura (también referida como soporte):** numero de instancias a las que la regla se aplica y se predice correctamente.

- **Confianza:** proporción de instancias que la reglase predice correctamente, es decir la cobertura dividida por el número de instancias a las que se le puede aplicar la regla.

Suponiendo el ejemplo de determinar si se juega a un deporte, consideramos:

Instancia	Pronostico	Humedad	Viento	Jugar
1	Soleado	Alta	Débil	No
2	Cubierto	Alta	Débil	Si
3	Lluvioso	Alta	Débil	Si
4	Lluvioso	Normal	Fuerte	No
5	Soleado	Normal	Débil	Si

Tabla 5: Si pronostico=Soleado Y viento=Débil ENTONCES jugar = Si

Tendrá una cobertura de 1, es decir el número de días con viento soleado y con viento débil en los que se recomienda jugar (instancia 5); y una confianza de ½ (ya que la regla también se puede aplicar a la instancia 1)

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Si la tarea es regresión, es decir, la salida del modelo es un valor numérico, la manera más habitual de evaluar un modelo es mediante el error cuadrático medio del valor predicho respecto al valor que se utiliza como validación. Esto promedia los errores y tiene más en cuenta aquellos errores que se desvían más del valor predicho (ponderación cuadrática). Aunque se pueden utilizar otras medidas del error en regresión, esta es quizás la más utilizada.

Para la tarea de agrupamiento, las medidas de evaluación, suelen depender del método utilizado, aunque suelen ser en función de la cohesión de cada grupo y de la separación entre grupos. La cohesión y separación entre grupos se puede formalizar, por ejemplo utilizando la distancia media al centro del grupo de los miembros de un grupo y la distancia media entre grupos, respectivamente. El concepto de distancia y de densidad son dos aspectos cruciales tanto en la construcción de modelos de agrupamiento como en su evaluación.

Interpretación y contextualización

Pese a todas las medidas vistas anteriormente, muchos casos hay que evaluar también el contexto donde el modelo se va a utilizar. Por ejemplo, en el caso de la clasificación y las reglas de asociación, usar la precisión como medida de calidad tiene ciertas desventajas. En primer lugar, no tiene en cuenta el problema de tener distribuciones de clases no balanceadas, es decir, tener muchas instancias de unas clases y muy pocas o ninguna de otras. Esta situación es habitual en la detección de fraudes y en el diagnostico medico. Un simple ejemplo puede ilustrar este punto. El servicio de urgencias de un hospital desea mejorar un sistema de admisión usando técnicas de aprendizaje inductivo. Cuando un paciente acude a las urgencias, una vez evaluado y realizadas las primeras atenciones, se pueden dar tres posibles situaciones que dependen de su estado: ser dado de alta, permanecer hospitalizado en observación o ser ingresado en la UCI. El porcentaje de casos de cada una de estas clases, por ejemplo del 86.5%, 13% y 0.5% respectivamente. Usando un algoritmo de minería sobre los datos referentes a pacientes pasados crearíamos un modelo que siempre diera de alta a todos los pacientes. En términos de precisión, este sería un buen modelo ya que su precisión seria de 86.5 % por ciento. Sin embargo este modelo es inútil. Mas importante todavía puede ser peligroso ya que pacientes cuyas condiciones de salud requieren un ingreso en la UCI serian enviados a sus casas, lo cual es una política desastrosa para el hospital.

El ejemplo anterior pone de manifiesto que necesitamos conocer mejor el tipo de errores y su coste asociado. En los problemas de clasificación se usa una matriz de confusión, la cual muestra el recuento de casos de las clases predichas y sus valores actuales. Si se dispone de información sobre el coste de cada error/acierto en la clasificación, entonces las celdas de la matriz pueden asociarse con el coste de cometer cierto error de clasificación o de efectuar una clasificación correcta. En este caso, la matriz suele denominarse matriz de costes de clasificación. Con estas dos matrices podemos evaluar los modelos con sus costes de error de clasificación, por ejemplo, buscar un modelo que minimice el coste global.

La consideración de que todos los errores no son iguales, puede incluso tenerse en cuenta en situaciones donde los costes de error suelen ser difíciles de estimar o incluso desconocidos para muchas aplicaciones. E estos casos, se usan estrategias alternativas como el análisis ROC (Receiver Operating Characteristics).

Como hemos dicho antes la precisión, de un modelo no garantiza que refleje el mundo real. Normalmente, esta situación se produce cuando al construir el modelo no hemos tenido en cuenta algunos parámetros que implícitamente influyen en el. Por ejemplo, un modelo para predecir el gasto energético de una población puede fallar al no haber considerado que un periodo de recensión económica mundial puede implicar subidas considerables en el precio del petróleo, lo que condiciona el problema para detectar y en su caso resolver los posibles conflictos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Fase de difusión, uso y monitorización

Una vez construido y validado el modelo puede usarse principalmente con dos funcionalidades: para que un analista recomiende acciones basándose en el modelo y sus resultados, o bien para aplicar al modelo a diferentes conjuntos de datos. También incorporarse a otras aplicaciones, como por ejemplo a un sistema de análisis de créditos bancarios, que asista al empleado bancario a la hora de evaluar a los solicitantes los créditos, o incluso automáticamente, como los filtros de spam o la detección de compras con tarjetas de crédito fraudulentas.

Tanto en el caso de una aplicación manual o automática del modelo, es necesario su difusión, es decir que se distribuya y se comunique a los posibles usuarios, ya sea por cauces habituales dentro de la organización, reuniones, Intranet, etc. El nuevo conocimiento extraído debe integrar el Know-how de la organización.

También es importante medir lo bien que en el modelo evoluciona. Aun cuando el modelo funcione bien debemos continuamente comprobar las prestaciones del mismo. Esto se debe principalmente a que los patrones pueden cambiar. Por ejemplo, todos los vendedores saben que las ventas se ven afectadas por factores externos como la tasa de inflación, la cual altérale comportamiento de compra de la gente. Por ello tanto, el modelo deberá ser monitorizado, lo que significa que de tiempo en tiempo el modelo tendrá que ser re-evaluado, re-entrenado y posiblemente reconstruido completamente.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

La decisión de implantar un programa de minería de datos (KDD) y el diseño de un plan del mismo deben preceder a cualquiera de las fases (fases del KDD) vistas anteriormente. De hecho, establecer cuál es el contexto, del negocio, los objetivos del mismo y plasmarlos en objetivos de minería de datos, es previo a pararnos en pensar en recopilar y preparar los datos, realizar modelos, evaluarlos y utilizarlos.

En primer lugar vamos hacer una breve descripción de las claves del éxito de un programa de minería de datos:

Según numerosos autores, experiencias anteriores y cierta reflexión apoyada en el sentido común, se pueden destacar los siguientes aspectos fundamentales para el éxito de la minería de datos en una organización:

- El negocio y sus necesidades han de dirigir el desarrollo del programa. Se han de especificar claramente los problemas y objetivos de negocio. Con estos problemas y objetivos de negocio podremos averiguar qué datos van a ser necesarios y podrán surgir los objetivos y tareas de la minería de datos.

- Una buena especificación de problemas concretos y específicos de minería de datos es otra clave del éxito. Es importante trasladar correctamente los objetivos de negocio a los objetivos concretos de minería de datos.

- La integración del resto de programas de la organización y el apoyo incondicional de los altos ejecutivos de la organización es imprescindible. La existencia de programas previos de calidad de datos y de almacenes de datos puede ser muy interesante.

- La calidad de datos es primordial. La integración de información externa necesaria para los modelos, como campañas de la competencia, evolución de la economía, calendarios, etc, es en muchos casos indispensable.

- El uso de herramientas integradas y de entornos amigables es otro factor destacable.

- La necesidad de un equipo heterogéneo de personal formado no solo en minería de datos, sino también en estadística, bases de datos y en el área de negocio. La continuidad de formación de este personal y de los usuarios es también muy importante.

- Una evaluación de los modelos más realista (teniendo en cuenta costes, comprensibilidad, relevancia, etc.…) y un despliegue de los mismos a todos los niveles permite trasladar los resultados de la minería de datos a resultados positivos en el negocio.

Una vez comentadas las claves del éxito de un programa de minerita de datos cabe preguntarse si es rentable para una organización implantar un programa de minería de datos. Lógicamente, si existen beneficios de su implementación, también habrá inconvenientes y costes. Como la mayoría de las decisiones, se trata de sopesar si los beneficios son mayores que los costes.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

No obstante, aunque la minería de datos funciona en general, no funciona igual de bien en todos los ámbitos y evidentemente existirán organizaciones para las que será mayor el esfuerzo que el beneficio obtenido. No hay la misma posibilidad para una gran empresa de telecomunicaciones, una pequeña empresa de seguros, un equipo de fútbol, que las que existen para una casa rural no integrada en ninguna red de promoción. En muchos casos no es una cuestión de tamaño o incluso de la rama de negocio, sino más bien de dependencia, de que se tomen decisiones importantes diariamente sobre un entorno cambiante y que exista una cierta tradición de informatización y gestión de datos de la organización.

¿Qué necesito saber, por tanto para tomar la decisión de si debo implantar o no minería de datos? Indudablemente para tomar, una buena decisión se debe estudiar el caso en profundidad. Paradójicamente, en muchos casos, esto obliga a diseñar, aunque sea esquemáticamente, el programa de minería de datos, para saber si es factible o no. Es decir para saber si debo implantar un programa de minería de datos debo, en primer lugar, diseñar y tener entre las manos un programa de minería de datos. Esta situación se muestra en la **figura 10**:

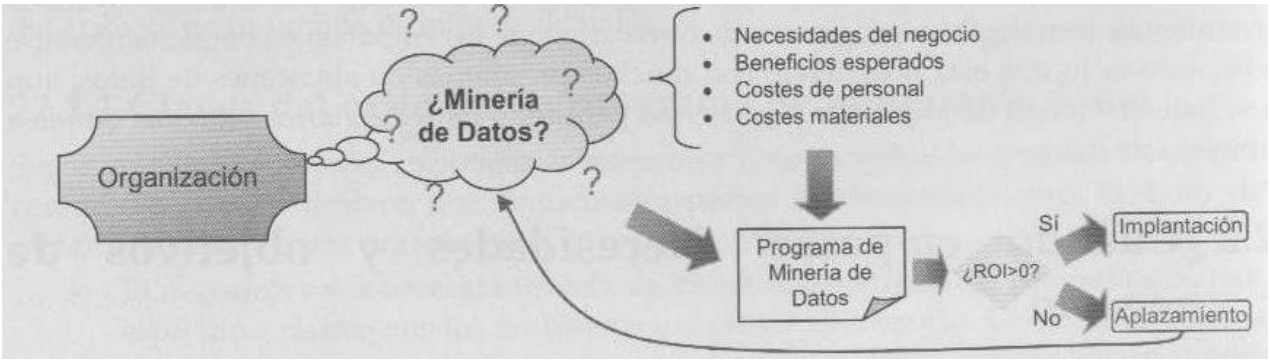


Figura 10: La necesidad de un programa de minería de datos sobre la mesa

Con un (esquema de) programa de minería de datos podemos estimar cual va a ser el rendimiento o ROI (Return of Investement), evaluando los beneficios y evaluando de la manera más ajustada posible los costes, ya que implementar minería de datos puede requerir una inversión considerable en formación, herramientas y personal. Se debe preservar en todo momento la duda sanísima de si, en su caso, la minería de datos le va a proporcionar, en su conjunto, mas beneficio que costes.

Un aspecto a considerar para determinar si es necesario un programa de minería de datos es (como no podría ser de otra manera) la identificación de necesidades que puedan ser cubiertas con la minería de datos. Y s que en cualquier caso la necesidad debe crear el programa y no el programa la necesidad. Las necesidades pueden ser clásicas: identificar razones de subidas o bajadas de ventas, adecuar la producción a ventas futuras, determinar tendencias de mercado o la competencia, realizar una campaña de marketing selectiva de clientes ante cierto producto, reducir las listas de espera en un hospital. Estas son las necesidades del negocio y van a marcar los objetivos de negocio.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Fases de implantación

Anteriormente hemos visto que la elaboración de un programa de minería de datos es crucial para tomar la una decisión definitiva sobre si hay que embarcarse en el o no en el. Pasemos, por tanto a formular dicho programa. ¿Qué debo incluir en un programa? ¿Qué estructura debo seguir? ¿Qué extensión debe de tener el programa?

Lógicamente, el programa, y en especial su extensión, será muy diferente si se trata de una gran empresa o si lo desarrollamos en una pequeña empresa. Es importante comprender que mientras una pequeña empresa tiene menos recursos (materiales y personales) para llevar a cabo un programa de minería de datos en una gran empresa es mucho más complejo que en una pequeña empresa. Esta es la razón por la cual hoy en día plantearse un programa de minería de datos es igualmente razonable en pequeñas o grandes empresas. Para ambos casos existen soluciones adecuadas.

Sobre los aspectos a tratar en el programa de minería de datos, podemos, en primer lugar, destacar el contexto y las necesidades del negocio, para poder formular los objetivos de negocio. A partir de ellos, podríamos destacar las fases del proceso de KDD: recopilación de datos, integración, limpieza, selección, minería de datos, evaluación y uso. Quizás aquí la importancia radica en la formulación de los objetivos de negocio, a partir de los cuales se va dando forma a las fases. Evidentemente, cuanto mejor formulados estén los problemas más sencillo será realizar el resto del proceso.

Para formular el programa es muy conveniente establecer no solo las fases, sino también sus interconexiones y las salidas (el subproducto) de cada fase. Esto puede realizarse utilizando herramientas clásicas de gestión de proyectos o puede llevarse a cabo bajo las nuevas herramientas de “workflow management” (gestión del flujo de trabajo).

Empezar de cero a establecer estas fases puede ser complicado la primera vez que se realiza un programa de minería de datos. La mayoría de las organizaciones desearían tener un patrón o metodología para llevarlo a cabo. Afortunadamente, este patrón existe: se dispone de una metodología estándar para el proceso de minería de datos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

CRISP-DM El modelo y guía de referencia

CRISP-DM (Cross-Industry Standard Process for Data Mining) es un consorcio de empresas (Inicialmente bajo la supervisión de la comisión Europea), incluyendo SPSS, NCR, DaimlerChrysler. La difusión de este estándar ha sido altísima y, al ser independiente de la plataforma o herramienta, está siendo utilizado por cientos de organizaciones en todo el mundo.

El estándar 1.0 incluye un modelo de referencia y una guía para llevar a cabo un proyecto de minería de datos. La guía puede ser muy útil como referencia a la hora de establecer una formulación o planificación de un programa de minería de datos adaptado a las necesidades de cada organización. El modelo y la guía se estructuran en seis fases principales, como se muestra en la **figura 11**. Como se puede observar, existen realimentación bidireccional entre algunas fases, es decir, algunas fases pueden obligar a revisar parcial o totalmente fases anteriores.

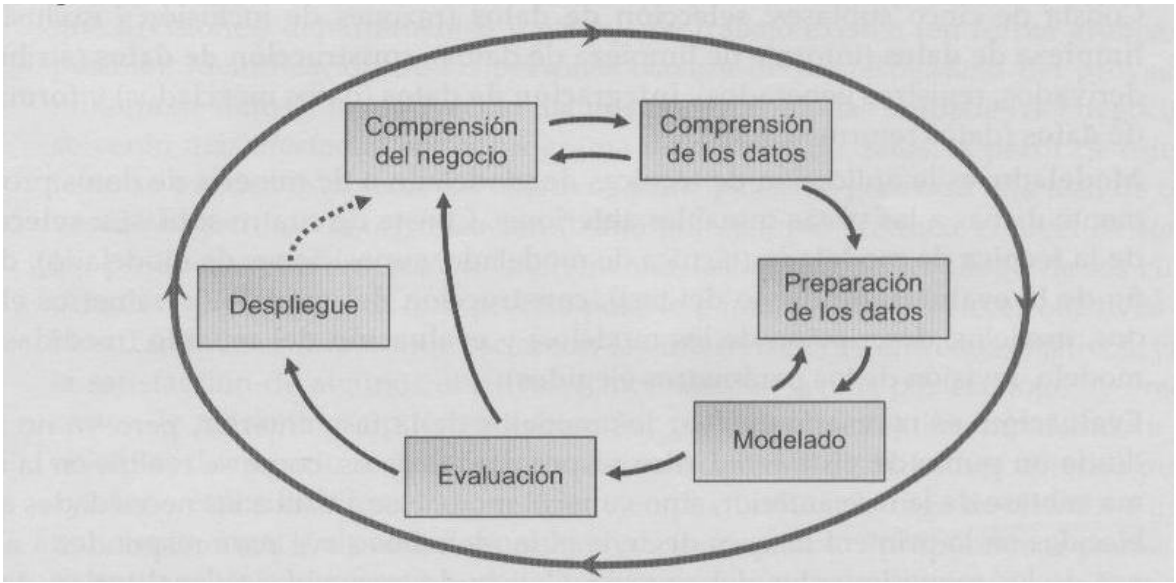


Figura 11: Fases del modelo de referencia CRISP-DM

Pasemos a describir brevemente las fases, las subfases (mostradas en negritas) y los subproductos de cada fase (que se muestran entre paréntesis).

- **Comprensión del negocio:** la primera fase se centra en entender los objetivos y requerimientos del proyecto desde una perspectiva de negocio, plasmando todo esto en una definición del problema de minería de datos y un plan preliminar. diseñado a obtener los objetos. Consta de cuatro subfases: **establecimiento de los objetivos de negocio** (contexto inicial, objetivos y criterios de éxito), **evaluación de la situación** (inventario de recursos, requerimientos, suposiciones y restricciones, riesgos y contingencias, terminología y costes y beneficios), **establecimiento de los objetivos de minería de datos** (objetivos de minería de datos y criterios de éxito) y **generación del plan de proyecto** (plan del proyecto y evaluación inicial de las herramientas y técnicas).

- **Comprensión de los datos:** según este estándar, se trata de recopilar y familiarizarse con los datos, identificar los problemas de la calidad de datos y ver las primeras potencialidades o subconjunto de datos que puede ser interesante analizar (para ello es muy importante haber establecido los objetivos de negocio en la fase anterior). La fase de comprensión de los datos consta de cuatro subfases: **recopilación inicial de los datos** (informe de recopilación), **descripción de datos** (informe de descripción), **exploración de datos** (informe de exploración) y **verificación de la calidad de los datos** (información de calidad).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Preparación de datos:** el objetivo de esta fase es obtener es obtener la vista minable aunque el estándar no use esta terminología. Aquí se incluye la integración, sección, limpieza y transformación. Consta de cinco subfases: **selección de datos** (razones de inclusión/selección), **limpieza de datos** (informe de limpieza de datos), **construcción de datos** (atributos derivados, registros generados), **integración de datos** (datos mezclados) y **formateo de datos** (información de calidad)

- **Modelado:** es la aplicación de técnicas de modelado o de minería de datos propiamente dichas a las vistas minables anteriores. Consta de cuatro subfases: **selección de la técnica de modelado** (técnica de modelado, suposiciones de modelado), **diseño de la evaluación** (diseño del test), **construcción del modelo** (parámetros elegidos, modelos, descripción de los modelos) y **evaluación del modelo** (medidas del modelo, revisión de los parámetros elegidos).

- **Evaluación:** es necesario evaluar los modelos de la fase anterior, pero ya no solo desde el punto de vista estadístico respecto a los datos, como se realiza en la ultima subfase de la fase anterior, sino ver si el modelo nos sirve para responder a algunos de los requerimientos del negocio. Consta de tres subfases: evaluación de resultados (evaluación de los resultados de minería de datos, modelos aprobados), revisar el proceso (revisión del proceso) y establecimiento de los siguientes pasos (lista de posibles acciones, decisión).

- **Despliegue:** se trata de explotar la potencialidad de los modelos, integrarlos en los procesos de toma de decisiones de la organización, difundir informes sobre el conocimiento extraído, etc. Consta de cuatro subfases: **planificación del despliegue** (plan del despliegue), **planificación de la monitorización y del mantenimiento** (plan de la monitorización y el despliegue), **generación del informe final** (informe final, presentación final) y **revisión del proyecto** (documentación de la experiencia).

Casi todas estas fases corresponden con fases ya vistos en el KDD (Aunque no exactamente en el mismo orden y agrupadas de la misma manera). Lo destacable del estándar CRISP-DM es, en primer lugar, el detalle de las fases y la especificación de los subproductos de cada subfase.

Pero quizás la fase más novedosa y relevantes de las incluidas en el CRISP-DM y que queremos recalcar es la primera, la fase de “Comprensión del negocio”. Es cierto que cada subfase de la “Comprensión del negocio” recuerda a las fases de especificación de requerimientos y análisis de cualquier gestión de proyectos. Quizás uno de los aspectos más importantes es la jerarquía de objetos, es decir, se marcan unos objetivos en términos del negocio y después se plasman o convierten en objetos, generalmente más concretos, en términos de minería de datos.

La especificación del CRISP-DM, en especial la guía, detalla el resto de fases. Como hemos dicho, no es mala opción obtener una copia de la guía (gratuitamente desde la Web del consorcio) y adaptarlo a nuestras necesidades. Además de servirnos como una buena referencia, permite que el personal externo se acople más fácilmente (si está familiarizado con las fases y terminología de CRISP-DM) y permite comparar también unos proyectos con otros. Existen otras guías, como por ejemplo la guía SEMMA del SAS Enterprise Miner, pero son menos generales, menos completas o menos independientes del fabricante.

Como hemos visto, el CRISP-DM se basa en las fases de extracción de conocimiento (KDD), englobándolas con una series de subfases previas de “comprensión del negocio” y unas subfases finales de evaluación del proyecto.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Herramienta WEKA

A la hora de realizar el proyecto he utilizado la herramienta WEKA para llevar a cabo todas las fases (excepto la primera y la ultima) de la metodología CRISP-DM. A continuación voy explicar que es esta herramienta.

Weka (Waikato Environment for Knowledge Analysis - Entorno para Análisis del Conocimiento de la Universidad de Waikato) es un conocido software para el aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato. WEKA es un software libre distribuido bajo licencia GNU-GPL.

Breve historia

- En 1993, la Universidad de Waikato en Nueva Zelanda inició el desarrollo de la versión original de Weka (en TCL/TK y C).
- En 1997, se decidió reescribir el código en Java incluyendo implementaciones de algoritmos de modelado.
- En 2005, Weka recibe de SIGKDD (*Special Interest Group on Knowledge Discovery and Data Mining*) el galardón "***Data Mining and Knowledge Discovery Service***".

Descripción:

El paquete Weka contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. La versión original de Weka fue un *front-end* en TCL/TK para modelar algoritmos implementados en otros lenguajes de programación, más unas utilidades para pre procesamiento de datos desarrolladas en C para hacer experimentos de aprendizaje automático.

Esta versión original se diseñó inicialmente como herramienta para analizar datos procedentes del dominio de la agricultura, pero la versión más reciente basada en Java (WEKA 3), que empezó a desarrollarse en 1997, se utiliza en muchas y muy diferentes áreas, en particular con finalidades docentes y de investigación.

Razones a favor de WEKA

Los puntos fuertes de Weka son:

- Está disponible libremente bajo la licencia pública general de GNU.
- Es muy portable porque está completamente implementado en Java y puede correr en casi cualquier plataforma.
- Contiene una extensa colección de técnicas para pre procesamiento de datos y modelado.
- Es fácil de utilizar por un principiante gracias a su interfaz gráfica de usuario.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Weka soporta varias tareas estándar de minería de datos, especialmente, pre procesamiento de datos, clustering, clasificación, regresión, visualización, y selección. Todas las técnicas de Weka se fundamentan en la asunción de que los datos están disponibles en un fichero plano (*flat file*) o una relación, en la que cada registro de datos está descrito por un número fijo de atributos (normalmente numéricos o nominales, aunque también se soportan otros tipos). Weka también proporciona acceso a bases de datos vía SQL gracias a la conexión JDBC (*Java Database Connectivity*) y puede procesar el resultado devuelto por una consulta hecha a la base de datos. No puede realizar minería de datos multi-relacional, pero existen aplicaciones que pueden convertir una colección de tablas relacionadas de una base de datos en una única tabla que ya puede ser procesada con Weka.

Carencias de Weka

Un área importante que actualmente no cubren los algoritmos incluidos en Weka es el modelado de secuencias.

La interfaz de usuario

Al ejecutar la aplicación nos aparece el **selector de interfaz de Weka** (*Weka GUI Chooser*) que da la opción de seleccionar entre cuatro posibles interfaces de usuario para acceder a las funcionalidades del programa, éstas son "*Simple CLI*", "*Explorer*", "*Experimenter*" y "*Knowledge Flow*".

Simple CLI:

Simple CLI es la abreviatura de *Simple Command-Line Interface* (Interfaz Simple de Línea de Comandos); se trata de una consola que permite acceder a todas las opciones de Weka desde línea de comandos.

Explorer:

La interfaz **Explorer** (Explorador) dispone de varios paneles que dan acceso a los componentes principales del banco de trabajo:

El panel "*Preprocess*" dispone de opciones para importar datos de una base de datos, de un fichero CSV, etc., y para pre procesar estos datos utilizando los denominados algoritmos de *filtrado*. Estos filtros se pueden utilizar para transformar los datos (por ejemplo convirtiendo datos numéricos en valores discretos) y para eliminar registros o atributos según ciertos criterios previamente especificados.

El panel "*Classify*" permite al usuario aplicar algoritmos de clasificación estadística y análisis de regresión (denominados todos *clasificadores* en Weka) a los conjuntos de datos resultantes, para estimar la exactitud del modelo predictivo resultante, y para visualizar predicciones erróneas, curvas ROC, etc., o el propio modelo (si este es susceptible de ser visualizado, como por ejemplo un árbol de decisión).

El panel "*Associate*" proporciona acceso a las reglas de asociación aprendidas que intentan identificar todas las interrelaciones importantes entre los atributos de los datos

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El panel "*Cluster*" da acceso a las técnicas de *clustering* o agrupamiento de Weka como por ejemplo el algoritmo K-Means. Este es sólo una implementación del algoritmo expectación-maximización para aprender una mezcla de distribuciones normales.

El panel "*Selected attributes*" proporciona algoritmos para identificar los atributos más predictivos en un conjunto de datos.

El panel "*Visualize*" muestra una matriz de puntos dispersos donde cada punto individual puede seleccionarse y agrandarse para ser analizados en detalle usando varios operadores de selección.

Experimenter:

La interfaz **Experimenter** (Experimentador) permite la comparación sistemática de una ejecución de los algoritmos predictivos de Weka sobre una colección de conjuntos de datos.

Knowledge Flow:

Knowledge Flow (Flujo de Conocimiento) es una interfaz que soporta esencialmente las mismas funciones que el *Explorer* pero con una interfaz que permite "*arrastrar y soltar*". Una ventaja es que ofrece soporte para el aprendizaje incremental.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En este proyecto se pretende realizar predicciones meteorológicas de la temperatura (máxima, media, mínima), precipitación y velocidad media del viento a 24 horas. Realizaremos estas predicciones utilizando datos públicos reales de tres estaciones meteorológicas navarras que aparecen incluidos en la página web del Gobierno de Navarra: http://meteo.navarra.es/. Concretamente, extraeremos la información de una estación localizada en el sur (Tudela montes de cierzo), otra en el centro (Pamplona (Larrabide)) y la última en el norte (Gorramendi).

Se utilizara como guía de referencia la metodología CRISP-DM para crear los mejores sistemas de minería de datos que predigan los datos meteorológicos (temperatura, precipitación, velocidad del viento) en cada una de las estaciones navarras seleccionadas.

Se pueden enumerar los siguientes objetivos principales:

- Comprender que es el proceso de KDD con todas sus fases y aplicaciones.
- Entender las diferentes tareas de minería de datos (predictivas y descriptivas) así como los correspondientes métodos (regresiones, redes neuronales etc...) para la resolución de estas tareas.
- Aplicar estándar CRISP-DM para desarrollar un sistema de minería de datos.
- Desarrollar un sistema de minería de datos para predecir la temperatura (máxima, media, mínima), velocidad media del viento y precipitación (discreteada) a las 24 horas.
- Comparar y validar los diferentes métodos y algoritmos de minería de datos para establecer los mejores.
- Comparar los resultados obtenidos para cada estación meteorológica y sacar conclusiones de ello.
- Realizar el despliegue (difusión) del sistema de minería de datos mediante una pequeña aplicación grafica en java.

Para la realización de este proyecto no me he basado en ningún antecedente ya que este proyecto es una propuesta mía. Sin embargo he encontrado diferentes proyectos de minería de datos aplicada sobre las predicciones meteorológicas y que me han servido de ayuda.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Recopilación y presentación de datos.

Para la realización de los distintos modelos (clasificaciones y regresiones) de minería de datos para predecir la temperatura (máxima, media, mínima), velocidad media del viento y precipitación (discreteada) a las 24 horas, hemos partido de los datos meteorológicos que nos proporciona el gobierno de Navarra para cada una de las estaciones a trabes de la web: http://meteo.navarra.es/ .

Para la realización de los distintos modelos de minería de datos he partido de los datos meteorológicos de las tres estaciones de Larrabide, Motes de cierzo y Gorramendi. De cada una de las estaciones he recopilado la información meteorológica de los años 2000 a 2008 (incluidos) con los cuales entrenare los distintos modelos de minería de datos y a continuación evaluare con estos mismos datos mediante un método de evaluación cruzada con 10 pliegues.

Todos los datos recopilados para este proyecto se encuentran en la carpeta llamada “datos” adjunta a esta documentación.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Desarrollo del trabajo.

En este apartado voy a mostrar todo el trabajo realizado en mi proyecto.

Para ello en primer lugar mostrare para cada una de las estaciones los resultados que obtenido a la hora de aplicar las distintas fases del KDD (todas las Fases de CRISP-DM menos la primera) con el objetivo de predecir la temperatura (máxima, media, mínima), velocidad media del viento y precipitación (discreteada) a las 24 horas. Para ello aplicare distintos métodos y algoritmos de minería de datos.

A continuación mostrare un análisis comparativo en base a los resultados que he obtenido para cada una de las estaciones. En esta comparativa se compararan los distintos algoritmos de minería de datos y los resultados obtenidos para las distintas estaciones con el fin de sacar conclusiones.

Por último mostrare la aplicación que he desarrollado en java para la predecir la temperatura (máxima, media, mínima), velocidad media del viento y precipitación (discreteada) a las 24 horas. Esta aplicación seria el despliegue del sistema de minería de datos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Estudio de las 3 estaciones:

En este apartado voy a mostrar para cada una de las estaciones los resultados que he obtenido a la hora de aplicar las distintas fases del KDD (todas las Fases de CRISP-DM menos la primera) con el objetivo de predecir la temperatura (máxima, media, mínima), velocidad media del viento y precipitación (discreteada) a las 24 horas. Para ello aplicare distintos métodos y algoritmos de minería de datos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Estación de Montes de Cierzo:

A continuación voy a mostrar el trabajo realizado en cada una de las fases del proceso del descubrimiento del conocimiento KDD, con el objetivo de poder predecir las temperaturas, velocidad media del viento y precipitación acumulada a 24h.

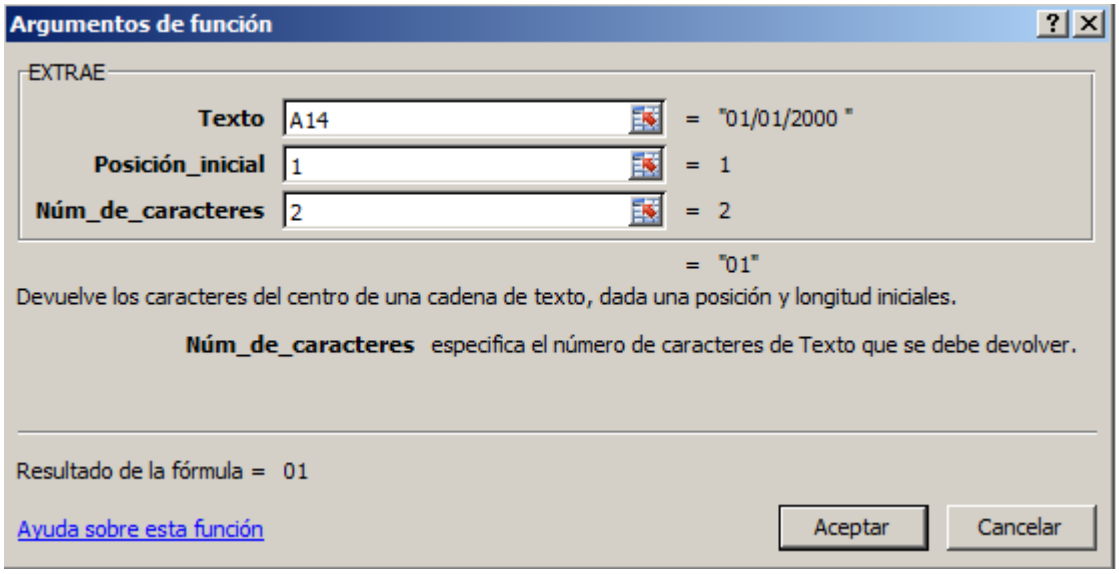
Integración y recopilación de los datos:

1. Descargarse los ficheros Excel con los datos meteorológicos de las estaciones en los años 2000-2008 (http://meteo.navarra.es/). A continuación hay que realizar los siguientes pasos para los datos de cada una de las estaciones.

2. Una vez descargados los ocho ficheros (uno por cada año) manualmente uniremos los ficheros manualmente para tener uno con toda la información.

3. A continuación añadiremos los nuevos atributos al archivo Excel del paso anterior:

   - **Día:** Atributo numérico con valores entre 1 y 31 que refleja el día de la medición. Lo obtenemos a través del atributo fecha realizando la extracción de los dos primeros caracteres.



   - **Mes:** atributo numérico con valores entre 1 y 12 que identifica el mes en el cual se realizo la medición. Lo obtenemos atreves del atributo fecha realizando la extracción de los caracteres 4 a 5.

   - **Año:** atributo numérico con valores entre 2000 y 2008 que muestra el año en el cual se realizo la medición. Lo obtenemos atreves del atributo fecha realizando la extracción de los caracteres 7 a 10.

A continuación añadiremos los atributos que vamos a predecir:

- **Temp med a 24h:** atributo numérico que contiene los valores de la temperatura media al día siguiente. Se obtiene creado un nuevo atributo y dando los valores de la temperatura

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

media desde la segunda medición hasta la última. El último valor de este atributo se obtiene de la medición del 1 de enero del 2009.

- **Temp max a 24h:** se obtiene igual que el anterior atributo pero en este caso se obtiene a partir de la temperatura máxima.

- **Temp min a 24h:** se obtiene igual que el anterior atributo pero en este caso se obtiene a partir de la temperatura mínima.

- **velocidad media viento a 24h:** se obtiene igual que los anteriores atributos pero en este caso se obtiene a partir de la velocidad media del viento.

- **precipitación acumulada a 24h** se obtiene igual que los anteriores atributos pero en este caso se obtiene a partir de la precipitación acumulada. Pero debido a que los valores de las precipitaciones acumuladas son mayoritariamente 0 o están entre valores entre 0 y 10 l/m2, decidí discretizar este atributo en 3 clases para facilitar a la hora de realizar la predicción (clasificación).

    - Clase 0: no llueve, es decir la precipitación acumulada es 0.

    - Clase 1: llueve pero débilmente, es decir la precipitación acumulada es mayor que 0 y menor que 15 l/m2.

    - Clase 2: llueve fuertemente es decir la precipitación acumulado es mayor o igual a 15 l/m2. (=SI(F3=0;0;SI(Y(F3 > 0;F3 < 15);1;SI(F3 >= 15; 2;-1))))

Con este cambio no tendré que predecir exactamente los que llueve sin aproximar si no llueve (0), llueve poco (1), llueve mucho (2).

4. A continuación cambiaremos los valores nulos del archivo Excel los cuales se representan por (- -) por los valores nulos (?) que los acepta Weka. Para ello en Excel pulsaremos edición/remplazar a continuación pondremos en buscar los caracteres "- -"y en remplazar el carácter "?" y a continuación pulsamos el botón en remplazar todo.

5. Ahora guardaremos el archivo del paso actual en formato .csv ya que este formato separa los campos por comas y esta es la forma de meter valores a Weka.

6. Una vez tenemos los datos en un bloc de notas crearemos la estructura del archivo .arff (formato de weka) la cual está compuesta por las relación y sus atributos. En nuestro caso la relación se va llamar meteorología y todos sus atributos son reales.

7. Por último se añaden los datos archivo arff. Para ello se copian los datos del fichero .csv del paso 5. Una vez realizado este paso el archivo resultante se guarda como .arff el cual es un formato soportado por Weka y por tanto lo único que faltaría es cargar este archivo en Weka.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
@relation metereologia

@attribute dia                      real
@attribute mes                      real
@attribute ano                      real
@attribute tem_med                  real
@attribute tem_max                  real
@attribute tem_min                  real
@attribute humedad_med              real
@attribute humedad_max              real
@attribute humedad_min              real
@attribute precip_acumulada         real
@attribute vel_media_viento         real
@attribute direc_viento             real
@attribute vel_racha_max            real
@attribute direc_racha_max          real
@attribute radiacion_global         real
@attribute insolacion               real
@attribute Temperatura_med_150mm    real
@attribute Temperatura_med_suelo_50 real
@attribute Temperatura_med_suel_200 real
@attribute Velocidad_med_vi_2m_MODA real
@attribute Velocidad_racha_máx_2m   real
@attribute Radiación_reflej_med     real
@attribute Dirección_racha_máx_2m   real
@attribute tem_med_24h              real
@attribute tem_max_24h              real
@attribute tem_min_24h              real
@attribute precip_acumulada_24h     {0,1,2}
@attribute vel_media_viento_24h     real


@data
1,1,2000,11.8,1.9,77,89,60,0.0,4.1,6,9.2,326,75.7,17114,6.6,6.6,7.1,1
.8,6,6.8,15.4,314,4.7,8.5,1.0,0,2.3
2,1,2000,8.5,1.0,83,98,70,0.0,2.3,3,5.2,230,31.5,110,5.9,7.0,7.4,1.2,
```

Formato del archivo .arff resultante

Una vez los datos han sido cargados en Weka se procede al pre proceso de los datos con la limpieza, trasformación y selección de los datos. Como se observa se pueden diferenciar dos sub fases una relacionada con la limpieza y trasformación y otra relacionada con la selección.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección, Limpieza y transformación de los datos:

En esta sub fase se tiene en cuenta diferentes aspectos como:

Limpieza de datos:

Tratamiento de valores perdidos (missing values)

- Aplicando el filtro ReplaceMissingValues el cual sustituye todos los valores no encontrados (missing) de un atributo por la moda y la media de ese atributo.

Tratamiento de valores anómalos (outlier values):

- Mediante el filtro IntercuatileRange detectamos los valores outlier (anómalos) así como los valores extremos. Pero debido a que al eliminar estos valores los métodos obtienen peores resultados hemos decidido no aplicar este filtro.

Trasformación de datos:

Discretizacion y numeración:

- Como he comentado anteriormente he discretizado el atributo numérico precipitación acumulada a 24 h en tres valores nominales (1, 2,3).

Normalización de rango (escalado):

- Al ser todos mis atributos numéricos y cada uno de ellos con un rango de valores muy diferentes he decidido normalizarlos para comprender mejor los atributos. Para ello he utilizado el filtro llamado Normalize el cual ha normalizado los valores entre 0 y 1.

Reducción de la dimensión:

- He eliminado dos atributos llamados **velocidad media viento km/h** y dirección **de la racha máxima km/h** ya que tenía otros dos atributos iguales pero medidos en m/s. Con los atributos y muestras que dispongo no es necesaria en mi caso ya que Weka puede tratarlos bien.

Aumento de la dimensión:

- Probé a introducir nuevos atributos derivados de los ya existentes como la diferencia entre la temperatura máxima y la mínima pero sin embargo los resultados no mejoraban por lo que no introduje nuevos atributos derivados. A demás al introducir nuevos atributos aumentaba el tiempo de cómputo y los resultados no mejoraban.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de datos:

Selección de las características (atributos) más importantes:

Weka incorpora una gran cantidad de métodos para estudiar la relevancia de atributos y realizar una selección automática de los mismos. Para utilizar estos métodos una vez cargados los datos es necesario elegir uno de estos métodos selección en la pestaña del explorador llamada select atributes. Hay dos tipos de evaluadores principales: los SubsetEval (evaluadores de subconjuntos o selectores) y AtributteEval (Prorreteadores de atributos). Los primeros tienen que elegir una estrategia o método de búsqueda de subconjuntos (Search Method) para encontrar el mejor subconjunto de atributos que nos proporcione los mejores resultados para un modelo de clasificación. Como resultado tendremos aquellos atributos con los cuales se ha realizado una mejor clasificación. El segundo tipo de evaluadores solo pueden combinarse con un Ranker ya que no seleccionan los atributos sino que los ordenan por su relevancia a la hora de clasificar el atributo objetivo.

A continuación muestro los resultados obtenidos del proceso de selección de atributos para los datos de la estación de Tudela (Montes de cierzo):

En primer lugar voy a enumerar todos los atributos que dispongo para realizar las predicciones/clasificaciones:

Atributo	Numero	Atributo	Numero
día	1	Velocidad de la racha máxima	13
mes	2	Dirección de la racha máxima	14
año	3	Radiación global	15
Temperatura media	4	insolación	16
Temperatura máxima	5	Temperatura media 150m	17
Temperatura mínima	6	Temperatura media 50m_suelo	18
Humedad media	7	Temperatura media 200m_suelo	19
Humedad máxima	8	Velocidad media viento 2m	20
Humedad mínima	9	Velocidad racha max 2m	21
Precipitación acumulada	10	Radiación reflect med 2m	22
Velocidad media del viento	11	Dirección racha máxima 2m	23
Dirección del viento	12		

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura máxima:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Temperatura máxima a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	2,4,6,7,8,9,12,14,16,17,18,20,23
Temperatura máxima a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,4,9,12,14,15,16,17,21,22
Temperatura máxima a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	1,4,6,7,8,9,10,12,14,15,16,17,18,19,20, 21
Temperatura máxima a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	4,6,7,8,10,12,14,15, 16,17,18,20, 21

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son bastante parecidos salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 4(temperatura media), 6(temperatura mínima), 12(Dirección del viento), 14(Dirección de la racha máxima), 16(insolación) ,17(Temperatura media a 150m).

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

 0.010265    2 mes
 0.006925   16 insolacion
 0.006787    4 tem_med
 0.006645    3 ano
 0.004444    5 tem_max
 0.004413   17 Temperatura_med_150mm
 0.00417    18 Temperatura_med_suelo_50
 0.003269    1 dia
 0.002002   20 Velocidad_med_vi_2m_MODA
 0.001784   11 vel_media_viento
 0.001168   14 direc_racha_max
 0.000806    9 humedad_min
 0.000388   22 Radiación_reflej_med
-0.000456    7 humedad_med
-0.000896    8 humedad_max
-0.000925    6 tem_min
-0.001042   12 direc_viento

Selected attributes:
2,16,4,3,5,17,18,1,20,11,14,9,22,7,8,6,12,10,21,15,19,13,23 : 23
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura máxima a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el mes, la insolación, las temperatura media, el año y la temperatura máxima.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura mínima:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Temperatura mínima a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	2,3,4,6,8,9,11,13,14,15,16,18,19,22,23
Temperatura mínima a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,4,7,8,13,15,16,22,23
Temperatura mínima a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,3,4,6,7,8,9,11,13,14,15,16,17,18,19,21,22
Temperatura mínima a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,3,4,7,8,9,13,14,15,16,17,18,19,21,23

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son prácticamente bastante parecidos salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 2,4, 8, 13, 15,16.

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo RelieF que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

Ranked attributes:

0.01173 16 insolacion
0.010124 5 tem_max
0.009083 2 mes
0.007583 4 tem_med
0.006002 18 Temperatura_med_suelo_50
0.004311 17 Temperatura_med_150mm
0.003216 3 ano
0.00117 1 dia
0.000864 8 humedad_max
0.000247 6 tem_min
-0.00028 14 direc_racha_max
-0.000458 9 humedad_min
-0.000664 20 Velocidad_med_vi_2m_MODA
-0.000907 7 humedad_med
-0.001144 11 vel_media_viento
-0.001335 10 precip_acumulada

Selected attributes:

16,5,2,4,18,17,3,1,8,6,14,9,20,7,11,10,12,21,19,22,15,13,23 : 23

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura mínima a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son la insolación, la temperatura máxima, el mes y la temperatura media.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura media:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Temperatura media a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	2,3,4,6,7,8,9,12,14, 15,16,17,19,20
Temperatura media a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,3,7,10,12,14,15,1 6,17,20,22,23
Temperatura media a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,4,5,6,8,9,10,12,13 ,14,15,16,17,18,20, 23
Temperatura media a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,3,4,5,6,8,9,10,12, 13,14,15,16,17,20

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son bastante parecidos salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 2, 12, 14, 15, 16, 17, 20.

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

 0.012388   2 mes
 0.01179   16 insolacion
 0.00893    4 tem_med
 0.00866    5 tem_max
 0.00644    3 ano
 0.006422  18 Temperatura_med_suelo_50
 0.005992  17 Temperatura_med_150mm
 0.002287   1 dia
 0.00037   14 direc_racha_max
 0.000299   9 humedad_min
 0.000295  20 Velocidad_med_vi_2m_MODA
-0.000203  11 vel_media_viento
-0.00066    8 humedad_max
-0.000755   6 tem_min
-0.000929  22 Radiación_reflej_med
-0.001431   7 humedad_med

Selected attributes:
2,16,4,5,3,18,17,1,14,9,20,11,8,6,22,7,12,10,21,19,15,13,23 : 23
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura media a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el mes, la insolación, la temperatura media, la temperatura máxima y el año.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Velocidad media del viento a 24h:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Velocidad media del viento a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	4,5,7,10,12,13,15,16,22,23
Velocidad media del viento a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,3,4,5,6,10,12,14,15,16,17,19
Velocidad media del viento a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,4,5,10,11,12,15,16,18
Velocidad media del viento a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,3,4,5,10,12,15,18

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son parecidos. Además todos estos métodos de selección comparten como atributos seleccionados los atributos (4, 5, 10, 12,15).

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

 0.005276   10 precip_acumulada
 0.004112   12 direc_viento
 0.004104    1 dia
 0.003502   19 Temperatura_med_suel_200
 0.003395   18 Temperatura_med_suelo_50
 0.003171    2 mes
 0.002659    3 ano
 0.002627    8 humedad_max
 0.002616    5 tem_max
 0.00214    16 insolacion
 0.002135   21 Velocidad_racha_máx_2m
 0.001654    4 tem_med
 0.001368    7 humedad_med
 0.000889    6 tem_min
 0.00071     9 humedad_min

Selected attributes:
10,12,1,19,18,2,3,8,5,16,21,4,7,6,9,14,17,15,22,11,13,23,20 : 23
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (velocidad media del viento a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son la precipitación acumulada, la dirección del viento, el día, la temperatura media a 200m y la temperatura media a 50m.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Precipitación acumulada a 24h:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Precipitación acumulada a 24h	WrapperSubsetEval	J48	GeneticSearch	8,9,22
Precipitación acumulada a 24h	WrapperSubsetEval	J48	BestFirst	5,6,9
Precipitación acumulada a 24h	WrapperSubsetEval	Logistic	GeneticSearch	5,8,9,10,12,14,15,16,18,23
Precipitación acumulada a 24h	WrapperSubsetEval	Logistic	BestFirst	1,3,4,6,7,8,9,10,11,12,13,16,17,18,19,23

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos no coinciden ya que solo tienen un atributo en común el numero 9. Se observa que con la selección de atributos se disminuye drásticamente el número de atributos seleccionados.

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:
 0.03581    3 ano
 0.02359   23 Dirección_racha_máx_2m
 0.0226     1 dia
 0.02181    2 mes
 0.01882   11 vel_media_viento
 0.01793   13 vel_racha_max
 0.0172    15 radiacion_global
 0.01565   20 Velocidad_med_vi_2m_MODA
 0.01303    7 humedad_med
 0.0114    10 precip_acumulada
 0.01131    6 tem_min
 0.01116   14 direc_racha_max
 0.01023   22 Radiación_reflej_med
 0.00976   12 direc_viento
 0.00975   19 Temperatura_med_suel_200
 0.0096    17 Temperatura_med_150mm

Selected attributes:
3,23,1,2,11,13,15,20,7,10,6,14,22,12,19,17,18,16,5,8,4,21,9 : 23
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (precipitación acumulada a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el año, dirección de la racha máxima, día, mes y velocidad media del viento.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En esta fase voy a mostrar los resultados obtenidos de aplicar los diferentes métodos de minería de datos a la tarea de predicción de los atributos. Como método de evaluación he utilizado una validación cruzada con 10 pliegues. En este método los datos se dividen aleatoriamente en n (pliegues) grupos. Un grupo se reserva para el conjunto de prueba y los n-1 restantes (juntando sus datos) se construye un modelo y se usa para predecir el resultado de los datos del grupo reservado. Este proceso se repite n veces dejando cada vez un grupo diferente para la prueba. Esto quiere decir que se calculan n ratios de error independientes. Finalmente se construye un modelo con todos los datos y se obtienen sus ratios de error y precisión promediando los n ratios de error disponibles.

Además en esta fase se aplicara las técnicas de minería de datos a todos los atributos de entrada y a los atributos seleccionados en la fase anterior y así realizar una comparativa de los resultados obtenidos.

A la hora de realizar la predicción de las temperaturas, la velocidad y la precipitación acumulada hemos utilizado estos métodos ya que son los que mejores resultados nos han ofrecido:

- **M5P:**

  Es un método basado en arboles de decisión para realizar predicciones.

  OPCIONES:

  - **minNumInstances** -- Mínimo numero de instancias para permitir que un nodo sea hoja.
  - **saveInstances** -- si desea guardar los datos de instancia en cada nodo del árbol
  - **unpruned –** Si poda de árboles / reglas deben ser generados.
  - **useUnsmoothed** -- si se deben usar las predicciones alterna.

- **LinearRegresion (Regresión lineal):**

Regresión lineal que utiliza el criterio de Akaike para la selección de atributos, y es capaz de hacer frente a casos ponderados.

OPCIONES

- **attributeSelectionMethod --** Establece el método utilizado para seleccionar atributos para el uso en la regresión lineal. Los métodos disponibles son: no hay selección de atributos, selección de atributos utilizando el método M5 (eliminar el atributo con el menor coeficiente normalizado hasta que no se observa mejoría en la estimación del error dada por el criterio de información Akaike), selección codiciosa utilizando la información de las métricas Akaike.
- **eliminateColinearAttributes --** Eliminar los atributos coloniales.
- **ridge –** El valor del parámetro Ridge.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **IsotonicRegresion:**

  Modelo de regresión isotónica.

- **LeastaMedSq:**

  Implementa la regresión lineal con menor error cuadrático medio utilizando el clasificador de Weka LinearRegresion. Esta función de regresión es generada a partir de un subconjunto aleatorio de muestras de datos.

OPCIONES

- **randomSeed** – Conjunto de semillas para la selección de un conjunto aleatorio de sub muestras de los datos de entrenamiento.
- **sampleSize** – Tamaño del conjunto de muestras aleatorias usadas para generar la regresión lineal con menor error cuadrático medio.

- **Bagging:**

  Es un clasificador para reducir la varianza. Puede ser utilizado sobre clasificadores o regresiones.

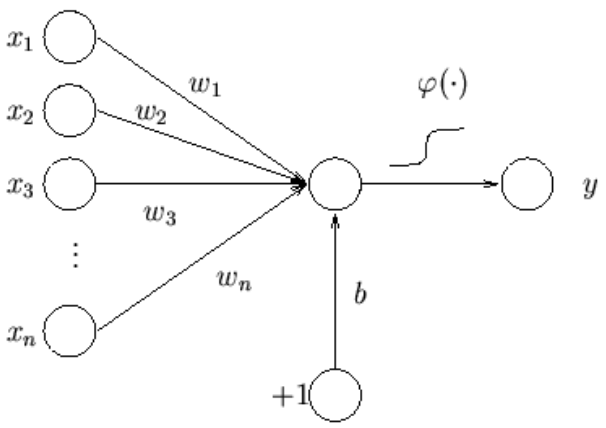
OPCIONES

- **bagSizePercent** -- Tamaño de cada bolsa (bag), como un porcentaje del tamaño del conjunto de entrenamiento.
- **calcOutOfBag** -- si fuera de la bolsa el error se calcula.
- **classifier** –el clasificador base que es utilizado.
- **numIterations** -- El número de iteraciones a realizar.
- **seed** –el número aleatorio de semillas que son utilizados.

- **MultilayerPerceptron (Perceptrón Multicapa):**

Este método crea una red multicapa de neuronas de tipo perceptrón que mediante un aprendizaje por backpropagation (propagación hacia atrás) es capaz de realizar clasificaciones muy exactas. Este método de aprendizaje llamado backpropagation ajusta el peso de todas las neuronas desde las neuronas del final hasta las del principio para que la salida de la red neuronal se ajuste lo máximo (minimice el error) con la salida esperada (datos de salida del entrenamiento). Por tanto este método consiste en entrenar la red neuronal aplicando continuamente (iteración a iteración) este proceso de actualización de los pesos (backpropagation) hasta que se minimice el error a la hora de clasificar.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira



Perceptrón

OPCIONES

- **GUI** – Proporciona una GUI para modificar o alterar la red neuronal durante el entrenamiento.

- **autoBuild** – Con esta opción los nodos de la red se generan automáticamente.

- **hiddenLayers** – Este parámetro define el número de neuronas en cada capa oculta de la red neuronal. Si no se tienen capas ocultas el valor de este parámetro vale 0. En el caso de estar activa la opción autoBuild se tiene unos valores comodín para este atributo 'a' = (atributos + clases) / 2,'i' = atributos, 'o' = clases, 't' = + atributos + clases.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error). Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – Es el numero de épocas (iteraciones) con las cuales se va a entrenar la red neuronal. La red neuronal terminara cuando se hayan completado todas las iteraciones.

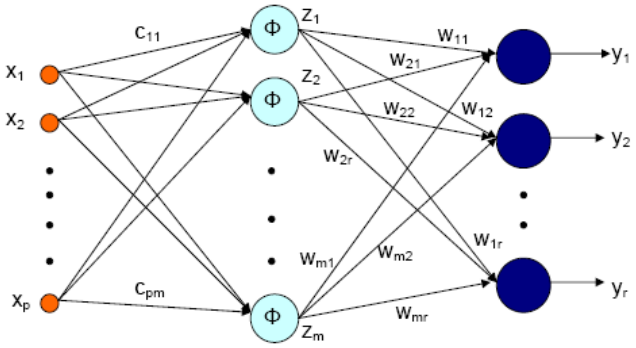
- **validationSetSize** – Tamaño en tanto por ciento del conjunto de validación. El entrenamiento de la red neuronal continuara hasta que se observe que el error en el conjunto de validación sea constantemente peor o se acaben el número de iteraciones (épocas) del entrenamiento. Si el valor de este parámetro es 0 entonces no se utilizara un conjunto de validación por lo que el entrenamiento se terminara cuando se ejecute la última época (iteración).

- **validationThreshold** – Es usado para determinar cuándo se termina la validación del test. Este valor dicta cuantas veces el error del conjunto de validación puede empeorar al ejecutar una iteración más.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **RBFNetwork (Red de base Radial):**

Este método crea una red de base radial la cual tiene la siguiente arquitectura:



Esta figura representa la arquitectura de una RBFN

Una red RBF, a diferencia de una MLP, Está conformada de tres capas.

- **La capa de entrada:** reciben las señales del exterior, no realizan ningún pre procesado.
- **La capa oculta:** (tantos nodos en la capa oculta como clases a clasificar) Está completamente interconectada entre todos sus nodos con la capa de entrada y activada a través de la función radial (gaussiana).
- **La capa de salida:** Se realiza una combinación lineal de las activaciones de las neuronas de la capa oculta y actúa como salida de la red. Hay tontos nodos en la capa de salida como clases a clasificar.

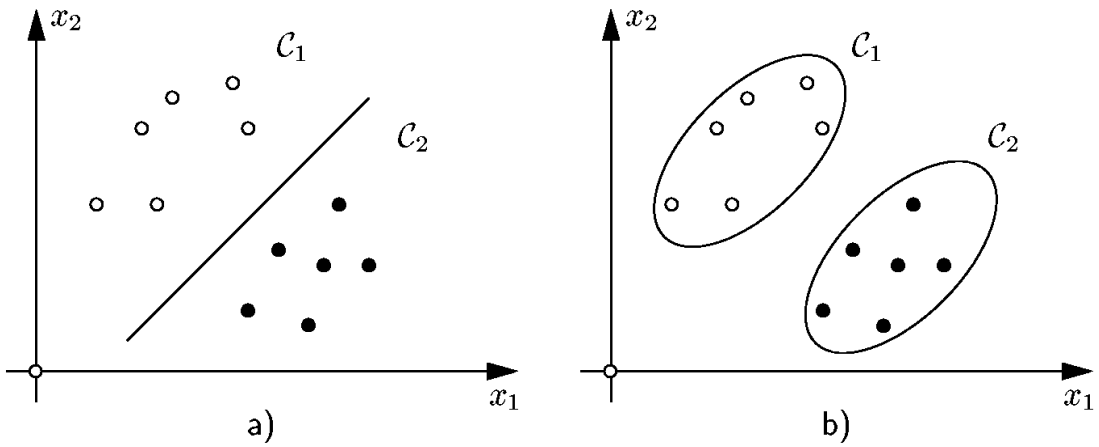


Figura: Espacio de Patrones bidimensionales: a) El perceptrón representa limites de decisión (lineales), b) Las RBF-NN representan clases (a través de las funciones base).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

OPCIONES

- **clusteringSeed**– Número aleatorio de semillas para aplicar el algoritmo de las K-means para el cálculo de los centros de las funciones radiales.
- **maxIts**– Número máximo de iteraciones del aprendizaje.
- **minStdDev**–Establece la desviación estándar mínima de los clusters.
- **numClusters**– Establece el numero de clusters para el algoritmo K-means.
- **ridge**– Establece el anillo para la regresión logística o lineal

- **NBTree:**

  Genera un árbol de decisión con clasificadores naive Bayes en las hojas.

- **DecisionTable:**

  Crea una tabla de decisión.

OPCIONES

- **crossVal** – Establece el numero de pliegues para la validación cruzada.
- **displayRules** – Muestra las reglas generadas.
- **evaluationMeasure** – Establece la medida usada para evaluar los resultados de la combinación de atributos usados en la tabla de decisión.
- **search** – Método de búsqueda usado para encontrar buenas combinaciones de atributos para la tabla de decisión.

- **Logistic:**

Genera una regresión logística multidimensional con estimador cadena (Ridge)

OPCIONES

- **maxIts** –Máximo número de iteraciones.
- **ridge** – Establece el valor del Ridge.

- **BayesNet:**

OPCIONES

- **estimator** – Algoritmo de selección para la búsqueda de tablas de condiciones de probabilidad en la red Bayesiana.
- **searchAlgorithm** – Método usado para la búsqueda de estructuras de red.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la temperatura máxima a 24h:

Con todos los atributos de entrada:

Resultados obtenidos con los diferentes métodos:

- **LeastMedSq (-S 1 -G 0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.
Los valores obtenidos son:

Linear Regression Model

tem_max_24h = 0.0782 * dia +-0.2658 * mes +0.0237 * ano +22.7261 * tem_med + -2.2633 * tem_max +-3.6147 * tem_min +-0.3167 * humedad_med + 5.2551 * humedad_max +1.2954 * humedad_min + 5.3674 * precip_acumulada +-0.0246 * vel_media_viento +-8.5772 * direc_viento +-0.194 * vel_racha_max +5.2824 * direc_racha_max +-2.144 * radiacion_global +11.2288 * insolacion +4.6952 * Temperatura_med_150mm +1.9557 * Temperatura_med_suelo_50 +-2.6386 * Temperatura_med_suel_200 +0.6272 * Velocidad_med_vi_2m_MODA +2.345 * Velocidad_racha_máx_2m +1.0799 * Radiación_reflej_med +-0.338 * Dirección_racha_máx_2m +0.1014

Time taken to build model: 2.43 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.9301
Mean absolute error	2.3475
Root mean squared error	3.0371
Relative absolute error	33.6513 %
Root relative squared error	36.7181 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.
Los valores obtenidos son:

Linear Regression Model

tem_max_24h = 23.0553 * tem_med +-1.9824 * tem_max +-4.04 * tem_min +5.4794 * humedad_max +1.3281 * humedad_min +5.2449 * precip_acumulada + -8.7185 * direc_viento +5.679 * direc_racha_max +-2.3282 * radiacion_global +10.7013 * insolacion +4.3902 * Temperatura_med_150mm +1.8392 * Temperatura_med_suelo_50 +-2.5216 * Temperatura_med_suel_200 +0.566 * Velocidad_med_vi_2m_MODA +2.5295 * Velocidad_racha_máx_2m +1.0911 * Radiación_reflej_med +-0.3856 * Dirección_racha_máx_2m +-0.1462

Time taken to build model: 0.07 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9301
Mean absolute error	2.3474
Root mean squared error	3.0376
Relative absolute error	33.6492 %
Root relative squared error	36.7238 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

Los valores obtenidos son:

Time taken to build model: 5.95 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9153
Mean absolute error	2.5923
Root mean squared error	3.3308
Relative absolute error	37.16 %
Root relative squared error	40.2683 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
LM1 (3220/36.461%)

LM num: 1

tem_max_24h = 23.0553 * tem_med - 1.9824 * tem_max - 4.04 * tem_min + 5.4794 * humedad_max + 1.3281 * humedad_min + 5.2449 * precip_acumulada - 8.7185 * direc_viento + 5.679 * direc_racha_max - 2.3282 * radiacion_global + 10.7013 * insolacion + 4.3902 * Temperatura_med_150mm + 1.8392 * Temperatura_med_suelo_50 - 2.5216 * Temperatura_med_suel_200 + 0.566 * Velocidad_med_vi_2m_MODA + 2.5295 * Velocidad_racha_máx_2m + 1.0911 * Radiación_reflej_med - 0.3856 * Dirección_racha_máx_2m - 0.1462

Number of Rules : 1

Time taken to build model: 2.98 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9303
Mean absolute error	2.3487
Root mean squared error	3.0346
Relative absolute error	33.6689 %
Root relative squared error	36.6869 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

LM1 (3220/36.033%)

LM num: 1

tem_max_24h = -0.4095 * mes + 23.7013 * tem_med - 3.1626 * tem_min + 4.7029 * humedad_max + 1.5451 * humedad_min + 5.2297 * precip_acumulada - 8.8947 * direc_viento - 0.5618 * vel_racha_max + 5.8136 * direc_racha_max - 1.4677 * radiacion_global + 8.4551 * insolacion + 4.7774 * Temperatura_med_150mm + 1.8321 * Temperatura_med_suelo_50 - 2.6817 * Temperatura_med_suel_200 + 0.7171 * Velocidad_med_vi_2m_MODA + 3.5264 * Velocidad_racha_máx_2m - 0.7831

Number of Rules: 1

Time taken to build model: 26.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9324
Mean absolute error	2.3112
Root mean squared error	2.99
Relative absolute error	33.1311 %
Root relative squared error	36.1477 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers --** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – número de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	27.55 s	0.9192	2.5861	3.281
500	0.2	10	0.01	35.15 s	0.9224	2.5359	3.2147
500	0.2	15	0.01	40.98 s	0.9219	2.5304	3.2199
500	0.01	15	0.01	42.16 s	0.93	2.3646	3.0448
3000	0.01	15	0.01	255.11 s	0.9355	2.3005	2.9352

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 5.000

Time taken to build model: 454.15 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9357
Mean absolute error	2.2887
Root mean squared error	2.9284
Relative absolute error	32.8085 %
Root relative squared error	35.4036 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_max_24h =-0.4458 * mes +19.9085 * tem_med +-4.9574 * tem_min +-0.2653 * humedad_med +6.1265 * humedad_max +1.2555 * humedad_min +-4.5342 * direc_viento +4.5703 * direc_racha_max +10.8758 * insolacion + 4.7532 * Temperatura_med_150mm +2.3228 * Temperatura_med_suelo_50 +0.6689 * Velocidad_med_vi_2m_MODA +-0.4576 * Dirección_racha_máx_2m + 0.6161

Time taken to build model: 2.29 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.93
Mean absolute error	2.3536
Root mean squared error	3.0409
Relative absolute error	33.7386 %
Root relative squared error	36.7637 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_max_24h =-0.4604 * mes +20.0555 * tem_med +-5.2312 * tem_min + 6.4008 * humedad_max +-4.422 * direc_viento + 4.4951 * direc_racha_max + 10.7565 * insolacion + 4.848 * Temperatura_med_150mm + 2.3756 * Temperatura_med_suelo_50 +0.6722 * Velocidad_med_vi_2m_MODA + -0.4685 * Dirección_racha_máx_2m + 0.4052

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9298
Mean absolute error	2.356
Root mean squared error	3.0436
Relative absolute error	33.7736 %
Root relative squared error	36.7965 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

Los valores obtenidos son:

Time taken to build model: 3.35 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9153
Mean absolute error	2.5923
Root mean squared error	3.3308
Relative absolute error	37.16 %
Root relative squared error	40.2683 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Number of Rules: 9

Time taken to build model: 2.25 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9317
Mean absolute error	2.3374
Root mean squared error	3.004
Relative absolute error	33.5068 %
Root relative squared error	36.3172 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
(using smoothed linear models)
LM1 (3220/36.208%)

LM num: 1

tem_max_24h = -0.557 * mes + 20.1775 * tem_med - 4.4239 * tem_min + 5.6422 * humedad_max + 1.5271 * humedad_min - 3.9403 * direc_viento + 4.6163 * direc_racha_max + 10.9631 * insolacion + 4.8756 * Temperatura_med_150mm + 2.1242 * Temperatura_med_suelo_50 + 0.7069 * Velocidad_med_vi_2m_MODA - 0.5015 * Dirección_racha_máx_2m - 0.0092

Number of Rules: 1

Time taken to build model: 18.93 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9326
Mean absolute error	2.3156
Root mean squared error	2.9857
Relative absolute error	33.1938 %
Root relative squared error	36.0965 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	17 s	0.9206	2.5304	3.2392
500	0.2	10	0.01	16.99 s	0.9243	2.4659	3.1633
500	0.2	15	0.01	25.36 s	0.9246	2.4735	3.1637
500	0.01	15	0.01	24.64 s	0.9295	2.3768	3.0554
3000	0.01	15	0.01	151.73 s	0.9361	2.2723	2.919

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15 (1 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 4.500

Los valores obtenidos son:

Time taken to build model: 217.34 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9363
Mean absolute error	2.2701
Root mean squared error	2.9142
Relative absolute error	32.5421 %
Root relative squared error	35.2323 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la temperatura media a 24h:

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =-0.0527 * dia +0.8595 * mes +0.6776 * ano +6.8291 * tem_med +-0.7413 * tem_max +0.8228 * tem_min +-1.0216 * humedad_med +4.9661 * humedad_max +1.698 * humedad_min +-0.6541 * precip_acumulada +0.2685 * vel_media_viento +0.0829 * direc_viento +-0.4089 * vel_racha_max +3.6401 * direc_racha_max +-2.5455 * radiacion_global +21.5846 * insolacion +2.9635 * Temperatura_med_150mm +0.8398 * Temperatura_med_suelo_50 +2.5007 * Temperatura_med_suel_200 +0.003 * Velocidad_med_vi_2m_MODA +-0.8742 * Velocidad_racha_máx_2m +-0.4968 * Radiación_reflej_med +-0.2474 * Dirección_racha_máx_2m +-10.4622

Time taken to build model: 2.44 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9395
Mean absolute error	1.6318
Root mean squared error	2.0871
Relative absolute error	31.7089 %
Root relative squared error	34.234 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =0.8315 * mes +0.684 * ano +7.0326 * tem_med +-0.4209 * tem_max +0.7693 * tem_min +-1.0328 * humedad_med +5.0594 * humedad_max +1.6972 * humedad_min +-0.685 * precip_acumulada +0.2623 * vel_media_viento +-0.4024 * vel_racha_max +3.6094 * direc_racha_max +-2.4948 * radiacion_global +21.1364 * insolacion +2.9865 * Temperatura_med_150mm +0.8317 * Temperatura_med_suelo_50 +2.4574 * Temperatura_med_suel_200 + -0.7778 * Velocidad_racha_máx_2m +-0.4893 * Radiación_reflej_med +-0.2471 * Dirección_racha_máx_2m +-10.5341

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9395
Mean absolute error	1.631
Root mean squared error	2.0868
Relative absolute error	31.6939 %
Root relative squared error	34.2286 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 4.39 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9276
Mean absolute error	1.7877
Root mean squared error	2.2772
Relative absolute error	34.7385 %
Root relative squared error	37.3518 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
insolacion <= 0.566 : LM1 (1717/38.969%)
insolacion > 0.566 : LM2 (1503/25.483%)

LM num: 1

tem_min_24h = 0.6688 * mes + 0.9836 * ano + 7.6648 * tem_med + 2.412 * tem_max + 0.0067 * tem_min - 0.987 * humedad_med + 5.308 * humedad_max + 1.6126 * humedad_min - 1.1297 * precip_acumulada + 0.0023 * vel_media_viento - 0.4943 * vel_racha_max + 2.783 * direc_racha_max - 3.2092 * radiacion_global + 15.7726 * insolacion + 3.0525 * Temperatura_med_150mm + 0.0072 * Temperatura_med_suelo_50 + 2.5968 * Temperatura_med_suel_200 - 0.0067 * Velocidad_racha_máx_2m + 1.8662 * Radiación_reflej_med - 0.0021 * Dirección_racha_máx_2m - 9.7417

LM num: 2

tem_min_24h = 0.2596 * dia + 2.9805 * mes + 0.287 * ano + 4.8515 * tem_med - 6.087 * tem_max + 3.0522 * tem_min - 1.6217 * humedad_med + 4.6725 * humedad_max + 1.7403 * humedad_min - 0.6822 * precip_acumulada + 0.5439 * vel_media_viento - 0.4691 * vel_racha_max + 4.4955 * direc_racha_max - 2.6522 * radiacion_global + 29.2867 * insolacion + 2.3805 * Temperatura_med_150mm + 0.0082 * Temperatura_med_suelo_50 + 2.0738 * Temperatura_med_suel_200 - 0.7846 * Velocidad_racha_máx_2m - 0.8808 * Radiación_reflej_med - 0.3462 * Dirección_racha_máx_2m - 11.7453

Time taken to build model: 2.66 seconds

=== Cross-validation ===

Correlation coefficient	0.94
Mean absolute error	1.626
Root mean squared error	2.0792
Relative absolute error	31.5958 %
Root relative squared error	34.1044 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
LM1 (3220/33.185%)

LM num: 1

tem_min_24h = 0.7975 * mes + 0.588 * ano + 6.5126 * tem_med + 1.3031 * tem_min - 1.4345 * humedad_med + 5.3352 * humedad_max + 2.3728 * humedad_min - 0.6226 * precip_acumulada + 0.7285 * direc_viento - 0.4107 * vel_racha_max + 3.5992 * direc_racha_max - 2.1393 * radiacion_global + 21.859 * insolacion + 3.0802 * Temperatura_med_150mm + 2.2051 * Temperatura_med_suel_200 - 1.1417 * Velocidad_racha_máx_2m - 0.3857 * Radiación_reflej_med - 0.2446 * Dirección_racha_máx_2m - 10.8149

Number of Rules: 1

Time taken to build model: 21.5 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9411
Mean absolute error	1.6064
Root mean squared error	2.0604
Relative absolute error	31.2153 %
Root relative squared error	33.7959 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers--** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	24.96 s	0.9324	1.7542	2.2161
500	0.2	10	0.01	24.39 s	0.9361	1.710	2.1571
500	0.2	15	0.01	42.12 s	0.9343	1.7272	2.189
500	0.01	15	0.01	37.67 s	0.941	1.6061	2.0617
3000	0.01	15	0.01	227.98 s	0.946	1.547	1.9764

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3.000

Los valores obtenidos son:

Time taken to build model: 601.14 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9463
Mean absolute error	1.5452
Root mean squared error	1.9712
Relative absolute error	30.027 %
Root relative squared error	32.3322 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.4335 * mes +0.3491 * ano +12.5683 * tem_med +-2.4385 * tem_min +-0.1807 * humedad_med +5.3345 * humedad_max +1.1317 * humedad_min + -3.297 * direc_viento +6.0263 * direc_racha_max +-2.8317 * radiacion_global + 16.7138 * insolacion + 3.9325 * Temperatura_med_150mm + 1.6993 * Temperatura_med_suel_200 + 0.3101 * Velocidad_med_vi_2m_MODA + -5.3814

Time taken to build model: 2.5 seconds
=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.9552
Mean absolute error	1.5563
Root mean squared error	2.0487
Relative absolute error	26.4414 %
Root relative squared error	29.5999 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
=== Classifier model (full training set) ===

Linear Regression Model

tem_med_24h =0.4364 * mes +0.3569 * ano +12.6524 * tem_med +-2.7095 * tem_min +5.5038 * humedad_max + 1.1299 * humedad_min +-3.2888 * direc_viento +6.0121 * direc_racha_max +-2.8262 * radiacion_global +16.6306 * insolacion +3.9492 * Temperatura_med_150mm +1.7065 * Temperatura_med_suel_200 +0.3187 * Velocidad_med_vi_2m_MODA +-5.4598

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9551
Mean absolute error                      1.5575
Root mean squared error                  2.0505
Relative absolute error                 26.4625 %
Root relative squared error             29.6259 %
Total Number of Instances                3220
Ignored Class Unknown Instances            68
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 3.14 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9481
Mean absolute error	1.6866
Root mean squared error	2.1995
Relative absolute error	28.6552 %
Root relative squared error	31.7783 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

Number of Rules: 9

Time taken to build model: 2.29 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9576
Mean absolute error	1.5325
Root mean squared error	1.9932
Relative absolute error	26.0377 %
Root relative squared error	28.7981 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

Number of Rules: 11

Time taken to build model: 18.16 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9581
Mean absolute error	1.5198
Root mean squared error	1.9824
Relative absolute error	25.8217 %
Root relative squared error	28.6419 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared errr
500	0.2	10	0.2	17.08 s	0.9519	1.6364	2.1239
500	0.2	10	0.01	15.3 s	0.9529	1.616	2.1002
500	0.2	15	0.01	23.15 s	0.9535	1.6073	2.0891
500	0.01	15	0.01	26.91 s	0.9558	1.5508	2.0355
3000	0.01	15	0.01	144.98 s	0.9598	1.4948	1.9455

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3500

Los valores obtenidos son:

Time taken to build model: 377.88 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9603
Mean absolute error	1.4843
Root mean squared error	1.9349
Relative absolute error	25.219 %
Root relative squared error	27.9548 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la temperatura mínima a 24h:

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =-0.0527 * dia +0.8595 * mes +0.6776 * ano +6.8291 * tem_med +-0.7413 * tem_max +0.8228 * tem_min +-1.0216 * humedad_med +4.9661 * humedad_max +1.698 * humedad_min +-0.6541 * precip_acumulada +0.2685 * vel_media_viento +0.0829 * direc_viento +-0.4089 * vel_racha_max +3.6401 * direc_racha_max +-2.5455 * radiacion_global +21.5846 * insolacion +2.9635 * Temperatura_med_150mm +0.8398 * Temperatura_med_suelo_50 +2.5007 * Temperatura_med_suel_200 +0.003 * Velocidad_med_vi_2m_MODA +-0.8742 * Velocidad_racha_máx_2m +-0.4968 * Radiación_reflej_med +-0.2474 * Dirección_racha_máx_2m +-10.4622

Time taken to build model: 2.16 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9395
Mean absolute error	1.6318
Root mean squared error	2.0871
Relative absolute error	31.7089 %
Root relative squared error	34.234 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =0.8315 * mes +0.684 * ano +7.0326 * tem_med +-0.4209 * tem_max +0.7693 * tem_min +-1.0328 * humedad_med +5.0594 * humedad_max +1.6972 * humedad_min +-0.685 * precip_acumulada +0.2623 * vel_media_viento +-0.4024 * vel_racha_max +3.6094 * direc_racha_max +-2.4948 * radiacion_global +21.1364 * insolacion +2.9865 * Temperatura_med_150mm +0.8317 * Temperatura_med_suelo_50 +2.4574 * Temperatura_med_suel_200 +-0.7778 * Velocidad_racha_máx_2m +-0.4893 * Radiación_reflej_med +-0.2471 * Dirección_racha_máx_2m +-10.5341

Time taken to build model: 0.03 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9395
Mean absolute error	1.631
Root mean squared error	2.0868
Relative absolute error	31.6939 %
Root relative squared error	34.2286 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 4.35 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9276
Mean absolute error	1.7877
Root mean squared error	2.2772
Relative absolute error	34.7385 %
Root relative squared error	37.3518 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
insolacion <= 0.566 : LM1 (1717/38.969%)
insolacion > 0.566 : LM2 (1503/25.483%)

LM num: 1

tem_min_24h = 0.6688 * mes + 0.9836 * ano + 7.6648 * tem_med + 2.412 * tem_max + 0.0067 * tem_min - 0.987 * humedad_med + 5.308 * humedad_max + 1.6126 * humedad_min - 1.1297 * precip_acumulada + 0.0023 * vel_media_viento - 0.4943 * vel_racha_max + 2.783 * direc_racha_max - 3.2092 * radiacion_global + 15.7726 * insolacion + 3.0525 * Temperatura_med_150mm + 0.0072 * Temperatura_med_suelo_50 + 2.5968 * Temperatura_med_suel_200 - 0.0067 * Velocidad_racha_máx_2m + 1.8662 * Radiación_reflej_med - 0.0021 * Dirección_racha_máx_2m - 9.7417

LM num: 2

tem_min_24h = 0.2596 * dia + 2.9805 * mes + 0.287 * ano + 4.8515 * tem_med - 6.087 * tem_max + 3.0522 * tem_min - 1.6217 * humedad_med + 4.6725 * humedad_max + 1.7403 * humedad_min - 0.6822 * precip_acumulada + 0.5439 * vel_media_viento - 0.4691 * vel_racha_max + 4.4955 * direc_racha_max - 2.6522 * radiacion_global + 29.2867 * insolacion + 2.3805 * Temperatura_med_150mm + 0.0082 * Temperatura_med_suelo_50 + 2.0738 * Temperatura_med_suel_200 - 0.7846 * Velocidad_racha_máx_2m - 0.8808 * Radiación_reflej_med - 0.3462 * Dirección_racha_máx_2m - 11.7453

Number of Rules : 2 , Time taken to build model: 2.51 seconds

=== Cross-validation ===

Correlation coefficient	0.94
Mean absolute error	1.626
Root mean squared error	2.0792
Relative absolute error	31.5958 %
Root relative squared error	34.1044 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Number of Rules : 1

M5 pruned model tree:
(using smoothed linear models)
LM1 (3220/33.185%)
LM num: 1

tem_min_24h = 0.7975 * mes + 0.588 * ano + 6.5126 * tem_med + 1.3031 * tem_min - 1.4345 * humedad_med + 5.3352 * humedad_max + 2.3728 * humedad_min - 0.6226 * precip_acumulada + 0.7285 * direc_viento - 0.4107 * vel_racha_max + 3.5992 * direc_racha_max - 2.1393 * radiacion_global + 21.859 * insolacion + 3.0802 * Temperatura_med_150mm + 2.2051 * Temperatura_med_suel_200 - 1.1417 * Velocidad_racha_máx_2m - 0.3857 * Radiación_reflej_med - 0.2446 * Dirección_racha_máx_2m - 10.8149

Number of Rules : 1

Time taken to build model: 22.45 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9411
Mean absolute error	1.6064
Root mean squared error	2.0604
Relative absolute error	31.2153 %
Root relative squared error	33.7959 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers --** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	26.16 s	0.9324	1.7542	2.2161
500	0.2	10	0.01	26.07 s	0.9361	1.7101	2.1571
500	0.2	15	0.01	38.69 s	0.9343	1.7272	2.189
500	0.01	15	0.01	37.58 s	0.941	1.6061	2.0617
3000	0.01	15	0.01	225.95 s	0.946	1.547	1.9764

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3.000

Los valores obtenidos son:

Time taken to build model: 394.65 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9463
Mean absolute error	1.5452
Root mean squared error	1.9712
Relative absolute error	30.027 %
Root relative squared error	32.3322 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =1.0044 * mes +0.5405 * ano + 7.6979 * tem_med +-0.6816 * tem_min +6.0183 * humedad_max +1.7627 * humedad_min +0.2786 * vel_media_viento +-0.409 * vel_racha_max +4.0219 * direc_racha_max +-2.7815 * radiacion_global +21.6112 * insolacion +1.5173 * Temperatura_med_suelo_50 +1.2514 * Temperatura_med_suel_200 +-0.2783 * Radiación_reflej_med +-0.2992 * Dirección_racha_máx_2m +-11.0804

Time taken to build model: 2.66 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9393
Mean absolute error	1.6347
Root mean squared error	2.0913
Relative absolute error	31.7649 %
Root relative squared error	34.3032 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =0.9816 * mes +0.5658 * ano +7.7799 * tem_med +-0.7002 * tem_min +6.1095 * humedad_max +1.7607 * humedad_min +0.2739 * vel_media_viento +-0.4087 * vel_racha_max +3.7662 * direc_racha_max +-2.7616 * radiacion_global +21.6098 * insolacion +1.518 * Temperatura_med_suelo_50 +1.2137 * Temperatura_med_suel_200 +-0.3029 * Dirección_racha_máx_2m +-11.1405

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9392
Mean absolute error	1.6356
Root mean squared error	2.0922
Relative absolute error	31.7838 %
Root relative squared error	34.3181 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 3.35 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9276
Mean absolute error	1.7877
Root mean squared error	2.2772
Relative absolute error	34.7385 %
Root relative squared error	37.3518 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
insolacion <= 0.566 : LM1 (1717/39.144%)
insolacion > 0.566 : LM2 (1503/25.838%)

LM num: 1

tem_min_24h = 0.868 * mes + 0.8257 * ano + 7.2575 * tem_med - 0.9456 * tem_min + 5.987 * humedad_max + 1.9813 * humedad_min + 0.0024 * vel_media_viento - 0.5747 * vel_racha_max + 3.9591 * direc_racha_max - 3.9414 * radiacion_global + 20.1876 * insolacion + 0.0131 * Temperatura_med_suelo_50 + 1.7019 * Temperatura_med_suel_200 + 1.8201 * Radiación_reflej_med - 0.0026 * Dirección_racha_máx_2m - 10.0615

LM num: 2

tem_min_24h = 2.8755 * mes + 0.2356 * ano + 9.4964 * tem_med - 0.0069 * tem_min + 6.5663 * humedad_max + 1.4952 * humedad_min + 0.474 * vel_media_viento - 0.5246 * vel_racha_max + 4.3152 * direc_racha_max - 2.1712 * radiacion_global + 20.2099 * insolacion + 1.188 * Temperatura_med_suelo_50 + 0.5485 * Temperatura_med_suel_200 - 0.6969 * Radiación_reflej_med - 0.4293 * Dirección_racha_máx_2m - 12.5917

Number of Rules: 2, Time taken to build model: 2.5 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9404
Mean absolute error	1.6141
Root mean squared error	2.0724
Relative absolute error	31.3656 %
Root relative squared error	33.9929 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

M5 pruned model tree:
insolacion <= 0.598 : LM1 (1886/37.432%)
insolacion > 0.598 : LM2 (1334/24.928%)

LM num: 1

tem_min_24h = 0.8911 * mes + 0.772 * ano + 6.8623 * tem_med - 1.3155 * tem_min + 6.9245 * humedad_max + 3.0031 * humedad_min + 0.002 * vel_media_viento - 0.6597 * vel_racha_max + 3.4738 * direc_racha_max - 3.2796 * radiacion_global + 21.3574 * insolacion + 0.0092 * Temperatura_med_suelo_50 + 1.7819 * Temperatura_med_suel_200 + 2.1036 * Radiación_reflej_med - 0.0022 * Dirección_racha_máx_2m - 10.7108

LM num: 2

tem_min_24h = 2.156 * mes + 0.214 * ano + 8.6422 * tem_med - 0.0086 * tem_min + 6.8729 * humedad_max + 1.9052 * humedad_min + 0.3076 * vel_media_viento - 0.4009 * vel_racha_max + 2.2697 * direc_racha_max - 0.7293 * radiacion_global + 20.7256 * insolacion + 1.0494 * Temperatura_med_suelo_50 + 0.015 * Temperatura_med_suel_200 - 0.7635 * Radiación_reflej_med - 0.2796 * Dirección_racha_máx_2m - 11.4857

Number of Rules: 2

Time taken to build model: 19.77 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9416
Mean absolute error	1.6033
Root mean squared error	2.0513
Relative absolute error	31.1554 %
Root relative squared error	33.6475 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	16.75 s	0.9335	1.7329	2.191
500	0.2	10	0.01	16.2 s	0.9364	1.6919	2.1432
500	0.2	15	0.01	25.98 s	0.9361	1.6875	2.1481
500	0.01	15	0.01	23.76 s	0.9403	1.6203	2.0733
3000	0.01	15	0.01	145.54 s	0.9433	1.5854	2.0245

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3.000

Los valores obtenidos son:

Time taken to build model: 398.6 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9437
Mean absolute error	1.5797
Root mean squared error	2.0154
Relative absolute error	30.6967 %
Root relative squared error	33.0572 %
Total Number of Instances	3220
Ignored Class Unknown Instances	68

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-0.0793 * dia +-0.1474 * mes +-0.0732 * ano +0.9838 * tem_med +-3.4556 * tem_max +0.0857 * tem_min +-0.1546 * humedad_med +0.3105 * humedad_max +-0.4596 * humedad_min + 4.1789 * precip_acumulada +0.0088 * vel_media_viento +1.4952 * direc_viento +-0.0521 * vel_racha_max + -0.132 * direc_racha_max +-0.9411 * radiacion_global +2.3086 * insolacion +0.5928 * Temperatura_med_150mm +0.2642 * Temperatura_med_suelo_50 +-0.1989 * Temperatura_med_suel_200 +0.0871 * Velocidad_med_vi_2m_MODA +-0.2425 * Velocidad_racha_máx_2m +0.7773 * Radiación_reflej_med +0.1327 * Dirección_racha_máx_2m +2.3896

Time taken to build model: 2.75 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5266
Mean absolute error	1.2712
Root mean squared error	1.5782
Relative absolute error	85.8675 %
Root relative squared error	85.9868 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-0.257 * mes +-0.2095 * ano +0.8919 * tem_med +-3.0524 * tem_max + 5.3026 * precip_acumulada + 2.5903 * direc_viento +-0.5165 * radiacion_global + 1.5415 * insolacion + 0.9568 * Temperatura_med_suelo_50 +-1.129 * Velocidad_racha_máx_2m + 2.5395

Time taken to build model: 0.11 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5316
Mean absolute error	1.258
Root mean squared error	1.5541
Relative absolute error	84.9769 %
Root relative squared error	84.6749 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 2.5 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5223
Mean absolute error	1.2653
Root mean squared error	1.5649
Relative absolute error	85.4665 %
Root relative squared error	85.2628 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0)** :

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
precip_acumulada <= 0.379 : LM1 (2266/76.852%)
precip_acumulada > 0.379 : LM2 (942/95.51%)

LM num: 1

vel_media_viento_24h = -0.1844 * mes - 0.2389 * ano + 0.0059 * tem_med - 5.525 * tem_max + 3.0409 * precip_acumulada + 2.0457 * direc_viento - 0.9381 * radiacion_global + 5.1667 * insolacion + 0.6957 * Temperatura_med_suelo_50 - 0.0074 * Velocidad_racha_máx_2m + 0.2932 * Dirección_racha_máx_2m + 2.9166

LM num: 2

vel_media_viento_24h = -0.3921 * dia - 0.392 * mes - 0.0033 * ano + 6.5286 * tem_med + 2.2642 * tem_max + 1.8442 * tem_min - 2.3158 * humedad_min + 8.7764 * precip_acumulada + 1.81 * direc_viento - 0.0081 * radiacion_global - 8.133 * insolacion + 1.5163 * Temperatura_med_150mm + 0.015 * Temperatura_med_suelo_50 - 1.3121 * Temperatura_med_suel_200 - 0.0177 * Velocidad_racha_máx_2m - 0.917

Number of Rules : 2
Time taken to build model: 3.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5367
Mean absolute error	1.2529
Root mean squared error	1.5487
Relative absolute error	84.629 %
Root relative squared error	84.3781 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

M5 pruned model tree:

precip_acumulada <= 0.438 : LM1 (2542/78.761%)
precip_acumulada > 0.438 : LM2 (666/96.785%)

LM num: 1

vel_media_viento_24h = -0.0009 * mes - 0.3239 * ano + 1.4582 * tem_med - 4.6069 * tem_max + 0.9382 * tem_min - 0.5972 * humedad_med - 0.4439 * humedad_max + 3.088 * precip_acumulada - 0.0012 * vel_media_viento + 1.473 * direc_viento - 0.0031 * direc_racha_max - 1.0785 * radiacion_global + 2.8004 * insolacion - 0.4564 * Temperatura_med_150mm + 1.2 * Temperatura_med_suelo_50 + 0.8631 * Temperatura_med_suel_200 + 0.2664 * Velocidad_med_vi_2m_MODA - 0.0034 * Velocidad_racha_máx_2m + 0.5515 * Radiación_reflej_med + 0.226 * Dirección_racha_máx_2m +2.6869

LM num: 2

vel_media_viento_24h = -0.5615 * dia - 0.0035 * mes - 0.0074 * ano + 7.8068 * tem_med + 0.7536 * tem_max + 0.0146 * tem_min - 0.0093 * humedad_med + 2.4174 * humedad_max - 3.6705 * humedad_min + 9.7268 * precip_acumulada - 0.4295 * vel_media_viento + 2.9225 * direc_viento - 2.2535 * direc_racha_max - 0.0176 * radiacion_global - 9.7412 * insolacion + 2.3531 * Temperatura_med_150mm + 1.5544 * Temperatura_med_suelo_50 - 2.3274 * Temperatura_med_suel_200 + 0.0054 * Velocidad_med_vi_2m_MODA - 0.0129 * Velocidad_racha_máx_2m + 1.697 * Radiación_reflej_med - 0.7952

Number of Rules: 2, Time taken to build model: 23.27 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5406
Mean absolute error	1.2493
Root mean squared error	1.5437
Relative absolute error	84.389 %
Root relative squared error	84.1047 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

  A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

  - **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

  - **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

  - **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

  - **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

  - **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	25.37 s	0.4329	1.3671	1.7585
500	0.2	10	0.01	26.18	0.4328	1.3456	1.7271
500	0.2	15	0.01	43.22 s	0.4458	1.3522	1.7297
500	0.01	15	0.01	40.74 s	0.5335	1.2547	1.5537
2000	0.01	15	0.01	242.78 s	0.4877	1.2925	1.6322

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 1500

Los valores obtenidos son:

Time taken to build model: 230.5 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.536
Mean absolute error	1.2524
Root mean squared error	1.5515
Relative absolute error	84.5994 %
Root relative squared error	84.5296 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos de seleccionados:

- **LeastMedSq (-S 1 -G 0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h = 0.4749 * tem_med +-3.0072 * tem_max + -0.0378 * humedad_med +3.9761 * precip_acumulada + 1.204 * direc_viento + 0.0203 * vel_racha_max + -1.0563 * radiacion_global + 2.6329 * insolacion + 0.9641 * Radiación_reflej_med +0.1735 * Dirección_racha_máx_2m +2.436

Time taken to build model: 2.35 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5298
Mean absolute error	1.2675
Root mean squared error	1.5763
Relative absolute error	85.6151 %
Root relative squared error	85.8803 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =0.9971 * tem_med +-2.8962 * tem_max +5.1814 * precip_acumulada +1.7191 * direc_viento +-0.8031 * radiacion_global + 1.6818 * insolacion +0.6763 * Radiación_reflej_med +0.1814 * Dirección_racha_máx_2m +2.1653

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.533
Mean absolute error	1 .2543
Root mean squared error	1.5524
Relative absolute error	84.7289 %
Root relative squared error	84.5802 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 1.1 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5223
Mean absolute error	1.2653
Root mean squared error	1.5649
Relative absolute error	85.4665 %
Root relative squared error	85.2628 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

M5 pruned model tree:

precip_acumulada <= 0.379 : LM1 (2266/77.074%)
precip_acumulada > 0.379 : LM2 (942/96.41%)

LM num: 1

vel_media_viento_24h = 0.0066 * tem_med - 5.752 * tem_max + 3.1789 * precip_acumulada + 2.062 * direc_viento - 0.9573 * radiacion_global + 5.866 * insolacion + 0.0044 * Radiación_reflej_med + 0.3085 * Dirección_racha_máx_2m + 2.6479

LM num: 2

vel_media_viento_24h = 5.2859 * tem_med + 4.3903 * tem_max + 0.8741 * humedad_med + 7.0779 * precip_acumulada + 1.8433 * direc_viento - 0.0126 * radiacion_global - 9.0285 * insolacion + 0.7648 * Radiación_reflej_med + 0.0028 * Dirección_racha_máx_2m - 0.5864

Number of Rules : 2

Time taken to build model: 2.1 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5407
Mean absolute error	1.2496
Root mean squared error	1.5436
Relative absolute error	84.4058 %
Root relative squared error	84.1009 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

M5 pruned model tree:
precip_acumulada <= 0.438: LM1 (2542/79.259%)
precip_acumulada > 0.438 : LM2 (666/99.45%)

LM num: 1

vel_media_viento_24h = 1.9691 * tem_med - 4.0227 * tem_max + 3.9337 * precip_acumulada + 1.379 * direc_viento - 1.3593 * radiacion_global + 2.2585 * insolacion + 0.9019 * Radiación_reflej_med + 0.2442 * Dirección_racha_máx_2m + 2.5093

LM num: 2

vel_media_viento_24h = 2.4018 * tem_med + 3.5185 * tem_max + 6.9222 * precip_acumulada + 2.7607 * direc_viento - 0.0244 * radiacion_global - 5.9016 * insolacion + 0.0201 * Radiación_reflej_med + 0.0049 * Dirección_racha_máx_2m + 0.2118

Number of Rules : 2

Time taken to build model: 15.69 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5421
Mean absolute error	1.2461
Root mean squared error	1.5419
Relative absolute error	84.1691 %
Root relative squared error	84.0063 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	13.65 s	0.5051	1.287	1.5935
500	0.2	10	0.01	13.55 s	0.5135	1.2842	1.5862
500	0.2	15	0.01	19.74 s	0.508	1.2863	1.5951
500	0.01	15	0.01	19.39 s	0.5367	1.255	1.549
2000	0.01	15	0.01	81.24 s	0.5473	1.2428	1.5369

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15
learningRate = 0.01
momentum = 0.01
trainingTime = 1.750

Los valores obtenidos son:

Time taken to build model: 69.74 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5474
Mean absolute error	1.2421
Root mean squared error	1.5366
Relative absolute error	83.9039 %
Root relative squared error	83.7183 %
Total Number of Instances	3208
Ignored Class Unknown Instances	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la precipitación acumulada a 24h:

Con todos los atributos de entrada:

- **NBTree:**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Time taken to build model: 4.68 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2318	71.8092 %
Incorrectly Classified Instances	910	28.1908 %
Kappa statistic	0.245	
Mean absolute error	0.2447	
Root mean squared error	0.3625	
Relative absolute error	86.2806 %	
Root relative squared error	96.3053 %	
Total Number of Instances	3228	
Ignored Class Unknown Instances	60	

=== Confusion Matrix ===

a	b	c	<-- classified as
2016	264	0	a = 0
585	302	1	b = 1
31	29	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **DecisionTable (-X 1 -S "weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1"):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

```
Time taken to build model: 1.6 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2327               72.088 %
Incorrectly Classified Instances      901                27.912 %
Kappa statistic                       0.1977
Mean absolute error                   0.2571
Root mean squared error               0.3581
Relative absolute error              90.6831 %
Root relative squared error          95.1424 %
Total Number of Instances             3228
Ignored Class Unknown Instances       60

=== Confusion Matrix ===
   a    b    c   <-- classified as
2114  166    0 |    a = 0
 675  213    0 |    b = 1
  33   27    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Logistic (-R 1.0E-8 -M -1):**

  Los valores obtenidos son:

Time taken to build model: 0.57 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2368	73.3581 %
Incorrectly Classified Instances	860	26.6419 %
Kappa statistic	0.2529	
Mean absolute error	0.2402	
Root mean squared error	0.3483	
Relative absolute error	84.6923 %	
Root relative squared error	92.5279 %	
Total Number of Instances	3228	
Ignored Class Unknown Instances	60	

=== Confusion Matrix ===

a	b	c	<-- classified as
2101	179	0	a = 0
621	267	0	b = 1
35	25	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **BayesNet (-D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

```
Time taken to build model: 0.08 seconds
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2109               65.3346 %
Incorrectly Classified Instances      1119               34.6654 %
Kappa statistic                          0.2525
Mean absolute error                      0.2407
Root mean squared error                  0.4304
Relative absolute error                 84.9015 %
Root relative squared error            114.3433 %
Total Number of Instances             3228
Ignored Class Unknown Instances         60

=== Confusion Matrix ===

   a   b   c   <-- classified as
1581 697   2 |   a = 0
 355 528   5 |   b = 1
  23  37   0 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **RBFNetwork (-B 2 -S 1 -R 1.0E-8 -M -1 -W 0.1):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

```
Time taken to build model: 0.66 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2343               72.5836 %
Incorrectly Classified Instances      885                27.4164 %
Kappa statistic                        0.2274
Mean absolute error                    0.2531
Root mean squared error                0.3565
Relative absolute error               89.2744 %
Root relative squared error           94.6975 %
Total Number of Instances              3228
Ignored Class Unknown Instances          60

=== Confusion Matrix ===

   a    b    c   <-- classified as
2094  186    0 |    a = 0
 639  249    0 |    b = 1
  37   23    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

  A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

  - **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

  - **hiddenLayers --** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

  - **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

  - **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

  - **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error
500	0.2	10	0.2	31.3 s	2225 (68.9281 %)	1003 (31.0719%)	0.2342
500	02	10	0.01	29.99 s	2272 (70.3841 %)	956 (29.6159%)	0.2325
500	0.2	15	0.01	47.66 s	2233 (69.176 %)	995 (30.824 %)	0.2271
500	0.01	15	0.01	47.33 s	2374 (73.544 %)	854 (26.456 %)	0.2422
700	0.01	15	0.01	52.79 s	2365 (73.2652 %)	863 (26.7348%)	0.2394

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15
learningRate = 0.01
momentum = 0.01
trainingTime = 625

Los valores obtenidos son:

```
Time taken to build model: 47.81 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2380               73.7299 %
Incorrectly Classified Instances       848               26.2701 %
Kappa statistic                          0.2899
Mean absolute error                      0.2403
Root mean squared error                  0.351
Relative absolute error                 84.7496 %
Root relative squared error             93.2324 %
Total Number of Instances             3228
Ignored Class Unknown Instances         60

=== Confusion Matrix ===
   a    b    c   <-- classified as
2060  220    0 |    a = 0
 568  320    0 |    b = 1
  29   31    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos seleccionados de entrada:

- **NBTree:**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
Time taken to build model: 0.41 seconds

=== Stratified cross-validation ===
=== Summary ==

Correctly Classified Instances        2314               71.6853 %
Incorrectly Classified Instances      914                28.3147 %
Kappa statistic                       0.2399
Mean absolute error                   0.2454
Root mean squared error               0.3629
Relative absolute error               86.5405 %
Root relative squared error           96.4135 %
Total Number of Instances             3228
Ignored Class Unknown Instances       60

=== Confusion Matrix ===
   a    b    c   <-- classified as
2016  264    0 |    a = 0
 590  298    0 |    b = 1
  32   28    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **DecisionTable(-X 1 -S "weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1"):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

```
Time taken to build model: 0.14 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2349               72.7695 %
Incorrectly Classified Instances       879               27.2305 %
Kappa statistic                          0.2185
Mean absolute error                      0.2548
Root mean squared error                  0.3574
Relative absolute error                 89.8687 %
Root relative squared error             94.9463 %
Total Number of Instances             3228
Ignored Class Unknown Instances         60

=== Confusion Matrix ===
   a    b    c   <-- classified as
 2122  158    0 |    a = 0
  661  227    0 |    b = 1
   35   25    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Logistic (-R 1.0E-8 -M -1):**

  Los valores obtenidos son:

Time taken to build model: 0.19 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2350	72.8005 %
Incorrectly Classified Instances	878	27.1995 %
Kappa statistic	0.1657	
Mean absolute error	0.2565	
Root mean squared error	0.358	
Relative absolute error	90.457 %	
Root relative squared error	95.1151 %	
Total Number of Instances	3228	
Ignored Class Unknown Instances	60	

=== Confusion Matrix ===

a	b	c	<-- classified as
2202	78	0	a = 0
740	148	0	b = 1
44	16	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **BayesNet ((-D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2287	70.8488 %
Incorrectly Classified Instances	941	29.1512 %
Kappa statistic	0.2533	
Mean absolute error	0.2404	
Root mean squared error	0.3677	
Relative absolute error	84.7953 %	
Root relative squared error	97.6705 %	
Total Number of Instances	3228	
Ignored Class Unknown Instances	60	

=== Confusion Matrix ===

a	b	c	<-- classified as
1937	343	0	a = 0
538	350	0	b = 1
29	31	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **RBFNetwork (-B 1 -S 1 -R 1.0E-8 -M -1 -W 0.1):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

```
Time taken to build model: 0.48 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2344               72.6146 %
Incorrectly Classified Instances      884                27.3854 %
Kappa statistic                        0.1934
Mean absolute error                    0.2566
Root mean squared error                0.3589
Relative absolute error               90.4986 %
Root relative squared error           95.3406 %
Total Number of Instances              3228
Ignored Class Unknown Instances        60

=== Confusion Matrix ===

   a    b    c   <-- classified as
2150  130    0 |    a = 0
 694  194    0 |    b = 1
  41   19    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- define el número de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error
500	0.2	10	0.2	10.12 s	2361 (73.1413 %)	867 (26.8587%)	0.3577
500	0.2	10	0.01	10.09 s	2364 (73.2342 %)	864 (26.7658%)	0.2528
500	0.2	15	0.01	14.66 s	2360 (73.1103 %)	868 (26.8897%)	0.2528
500	0.01	15	0.01	13.76 s	2348 (72.7385 %)	880 (27.2615%)	0.2571
700	0.01	15	0.01	19.49 s	2343 (72.5836 %)	885 (27.4164%)	0.2562

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 2
learningRate = 0.2
momentum = 0.2
trainingTime = 600

Los valores obtenidos son:

Time taken to build model: 3.86 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2369	73.3891 %
Incorrectly Classified Instances	859	26.6109 %
Kappa statistic	0.2424	
Mean absolute error	0.2536	
Root mean squared error	0.3573	
Relative absolute error	89.4344 %	
Root relative squared error	94.9163 %	
Total Number of Instances	3228	
Ignored Class Unknown Instances	60	

=== Confusion Matrix ===

a	b	c	<-- classified as
2121	159	0	a = 0
640	248	0	b = 1
36	24	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Evaluación e interpretación

A la hora de evaluar los modelos que he construido en la fase de minería de datos me voy a centrar en analizar los resultados de la evaluación de estos modelos con la validación cruzada con 10 pliegues. Este método de evaluación es muy preciso ya que entrena con todos los pliegues menos uno y luego evalúa el modelo creado con el pliegue que no se ha utilizado para entrenar. Este proceso se repite tantas veces como pliegues halla, escogiendo siempre un conjunto de pliegues distintos para entrenar y otro para evaluar.

Predicción de la temperatura máxima a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlationcoefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.98 s	0.9303	2.3487	3.0346	33.668%	36.686%	3220	68
LinearRegression	0.07 s	0.9301	2.3474	3.0376	33.649%	36.723%	3220	68
Isotonic regression	5.95 s	0.9153	2.5923	3.3308	37.16 %	40.268%	3220	68
LeastMedSq	2.43 s	0.9301	2.3475	3.0371	33.651%	36.718%	3220	68
MultilayerPerceptron	454.15 s	0.9357	2.2887	2.284	32.8%	35.4%	3220	68
Bagging (M5P -M 1.0) 10 iteraciones	26.02 s	0.9324	2.3112	2.99	33.131%	36.147%	3220	68

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.25 s	0.9317	2.3374	3.004	33.5068 %	36.317%	3220	68
LinearRegression	0.02 s	0.9298	2.356	3.0436	33.7736 %	36.796%	3220	68
Isotonic regression	3.35 s	0.9153	2.5923	3.3308	37.16 %	40.268 %	3220	68
LeastMedSq	2.29 s	0.93	2.3536	3.0409	33.7386 %	36.763%	3220	68
MultilayerPerceptron	217.34 s	0.9363	2.2701	2.9142	32.5421 %	35.232%	3220	68
Bagging (M5P -M 1.0) 10 iteraciones	18.93s	0.9326	2.3156	2.9857	33.193%	36.05 %	3220	68

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura máxima al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 2.27 grados y un error relativo de un 32.54 %. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 (bagging) y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye un poco y que los resultados mejoran (M5P, MultilayerPerceptron) con unos modelos y empeoran (Isotonic regression, LinearRegression, LeastMedSq) con otros pero no varían mucho frente a los obtenidos con todos los atributos.

Predicción de la temperatura media a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.66 s	0.94	1.626	2.0792	31.595%	34.104%	3220	68
LinearRegression	0.04 s	0.9395	1.631	2.0868	31.693%	34.228%	3220	68
Isotonic regression	4.39 s	0.9276	1.7877	2.2772	34.738%	37.351%	3220	68
LeastMedSq	2.44 s	0.9395	1.6318	2.871	31.708%	34.234 %	3220	68
MultilayerPerceptron	601.14 s	0.9463	1.5452	1.9712	30.027 %	32.332%	3220	68
Bagging (M5P -M 1.0) 10 iteraciones	21.5 s	0.9411	1.6064	2.0604	31.215%	33.795%	3220	68

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.29 s	0.9576	1.5325	1.9932	26.0377 %	28.798%	3220	68
LinearRegression	0.02 s	0.9551	1.557	2.050	26.4625 %	29.625%	3220	68
Isotonic regression	3.14 s	0.9481	1.6866	2.1995	28.6552 %	31.778%	3220	68
LeastMedSq	2.5 s	0.9552	1.5563	2.0487	26.4414 %	29.599%	3220	68
MultilayerPerceptron	377.88 s	0.9603	1.4843	1.9349	25.219 %	27.954%	3220	68
Bagging (M5P -M 1.0) 10 iteraciones	18.16 s	0.9581	1.5198	1.9824	25.8217 %	28.641%	3220	68

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura media al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 1.48 grados y un error relativo de un 25.21%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 (bagging) y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye con y que los resultados mejoran (en todos los modelos).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Predicción de la temperatura mínima a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.51 s	0.94	1.626	2.0792	31.595%	34.104%	3220	68
LinearRegression	0.03 s	0.9395	1.631	2.0868	31.693%	34.228%	3220	68
Isotonic regression	4.35 s	0.9276	1.7877	2.2772	34.738%	37.351%	3220	68
LeastMedSq	2.16 s	0.9395	1.6318	2.0871	31.708%	34.234%	3220	68
MultilayerPerceptron	394.65 s	0.9463	1.5452	1.9712	30.027 %	32.332%	3220	68
Bagging (M5P -M 1.0) 10 iteraciones	22.45 s	0.9411	1.6064	2.0604	31.215%	33.795%	3220	68

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.5 s	0.9404	1.6141	2.0724	31.3656 %	33.992%	3220	68
LinearRegression	0.02 s	0.9392	1.6356	2.0922	31.7838 %	34.318%	3220	68
Isotonic regression	3.35 s	0.9276	1.7877	2.2772	34.7385 %	37.351%	3220	68
LeastMedSq	2.66 s	0.9393	1.6347	2.0913	31.7649 %	34.303%	3220	68
MultilayerPerceptron	398.6 s	0.9437	1.5797	2.0154	30.6967 %	33.057%	3220	68
Bagging (M5P -M 1.0) 10 iteraciones	19.77 s	0.9416	1.6033	2.0513	31.1554 %	33.647%	3220	68

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura mínima al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 1.54 grados y un error relativo de un 30 %. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 (Bagging) y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye con la mayoría de los modelos un poco (M5P, LinearRegression, Bagging, Isotonic regression, Bagging) y que los resultados mejoran (M5P, Bagging) con unos modelos y empeoran (Isotonic regression, LinearRegression, LeastMedSq , MultilayerPerceptron) con otros pero prácticamente no varían frente a los obtenidos con todos los atributos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa de los mejores resultados para la predicción de las temperaturas:

Temperatura	Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error
Temperatura máxima	**MultilayerPerceptron**	217.34 s	0.9363	2.2701	2.9142	32.5421 %	35.232%
Temperatura media	**MultilayerPerceptron**	377.88 s	0.9603	1.4843	1.9349	25.219 %	27.954%
Temperatura mínima	**MultilayerPerceptron**	394.65 s	0.9463	1.5452	1.9712	30.027 %	32.332%

Como se observa las mejores predicciones las he obtenido mediante el método perceptrón multicapa. A demás se aprecia claramente que la predicción de la temperatura media es la más precisa (error relativo de un 25,2%) seguida de la predicción de la temperatura mínima (30%) y ya más atrás la temperatura máxima (32%). El hecho de que la temperatura media es la mejor predecible es en gran parte al hecho de que la temperatura media es menos variable que las temperaturas máximas y mínimas.

Predicción de la velocidad media del viento a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	3.04 s	0.5367	1.2529	1.5487	84.6%	84.3%	3208	80
LinearRegression	0.11 s	0.5316	1.258	1.5541	84.976%	84.674%	3208	80
Isotonic regression	2.5 s	0.5223	1.2653	1.5649	85.466%	85.262%	3208	80
LeastMedSq	2.75 s	0.5266	1.2712	1.5782	85.867%	85.986%	3208	80
MultilayerPerceptron	230.5 s	0.536	1.2524	1.5515	84.599%	84.529%	3208	80
Bagging (M5P -M 1.0) 10 iteraciones	23.27 s	0.5406	1.2493	1.5437	84.389 %	84.104%	3208	80

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.1 s	0.5407	1.2496	1.5436	84.4058 %	84.100%	3208	80
LinearRegression	0.02 s	0.533	1 .2543	1.5524	84.7289 %	84.580%	3208	80
Isotonic regression	1.1 s	0.5223	1.2653	1.5649	85.4665 %	85.262%	3208	80
LeastMedSq	2.35 s	0.5298	1.2675	1.5763	85.6151 %	85.880%	3208	80
MultilayerPerceptron	69.74 s	0.5474	1.2421	1.5366	83.9039 %	83.718%	3208	80
Bagging (M5P -M 1.0) 10 iteraciones	15.69 s	0.5421	1.2461	1.5419	84.1691 %	84.006%	3208	80

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la velocidad media del viento al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 1.2421 m/s y un error relativo de un 83.71%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye bastante y que los resultados mejoran (LinearRegression, M5P, LeastMedSq, MultilayerPerceptron).

Predicción de la precipitación acumulada a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
NBTree	4.68 s	2318 (71.8092 %)	910 (28.1908%)	0.2447	0.3625	86.280%	96.3053 %	3228	60
DecisionTable	1.6 s	2327 (72.088 %)	901 (27.912 %)	0.2571	0.3581	90.683%	95.1424 %	3228	60
Logistic	0.57 s	2368 (73.3581 %)	860 (26.6419%)	0.2402	0.3483	84.692%	92.5279 %	3228	60
BayesNet	0.08 s	2109 (65.3346 %)	1119 (34.6654%)	0.2407	0.4304	84.901%	114.3433 %	3228	60
RBFNetwork	0.66 s	2343 (72.5836 %)	885 (27.4164%)	0.2531	0.3565	89.274%	94.6975 %	3228	60
MultilayerPerceptron	47.81 s	2380 (73.7299 %)	848 (26.2701%)	0.2403	0.351	84.749%	93.2324 %	3228	60

Con los atributos seleccionados:

Método	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
NBTree	0.41 s	2314 (71.6853 %)	914 (28.3147%)	0.2454	0.3629	86.540%	96.4135 %	3228	60
DecisionTable	0.14 s	2349 (72.7695 %)	879 (27.2305%)	0.2548	0.3574	89.868%	94.9463 %	3228	60
Logistic	0.19 s	2350 (72.8005 %)	878 (27.1995%)	0.2565	0.358	90.457 %	95.1151 %	3288	60
BayesNet	0.05 s	2287 (70.8488 %)	941 (29.1512%)	0.2404	0.3677	84.795%	97.6705 %	3228	60
RBFNetwork	0.48 s	2344 (72.6146 %)	884 (27.3854%)	0.2566	0.3589	90.498%	95.3406 %	3228	60
MultilayerPerceptron	3.86 s	2369 (73.389%)	859 (26.61%)	0.2536	0.3573	89.4%	94.916%	3228	60

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura mínima al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se clasifican correctamente un 73.72 % de las instancias. El segundo mejor resultado se obtiene mediante una regresión Logística con un 73.35% de las instancias clasificadas correctamente. En los resultados de todos las clasificaciones se observa que la tasa de acierto de las instancias en la clase 1 (no llueve) es de 90%, la tasa de acierto de las instancias de la clase 2 (llueve débilmente) es de un 36% y la tasa de acierto de las instancias de la clase 3 (llueve fuerte) es de un 0%.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Dado que la mayoría de las instancias (el 70%) pertenecían a la clase 1 (no hay precipitación acumulada) y luego a la clase 2 (hay poca precipitación acumulada) al final la tasa de acierto general es de un 73.72%. De estos resultados se deduce que clasifica bien las instancias de la clase 1, clasifica mal las de la clase 2 y clasifica fatal las de la clase 3. Ha y que tener en cuenta que el 70% de las instancias son de la clase 1 (no hay precipitación acumulada) con lo que prediciendo siempre que no llueve se acertaría el 70% mientras que con redes neuronales conseguimos un acierto del 73.72% por lo que mejoramos solo un 3.72 %. Por tanto los resultados no son muy valiosos.

Otro aspecto a destacar es que al realizar la clasificación sobre los atributos seleccionados (3 atributos) y no sobre el total (23 atributos) de atributos se puede ver que el tiempo de construcción de los modelos disminuye bastante en y que los resultados empeoran frente a los obtenidos con todos los atributos excepto con los modelos RBFNetwork y DecisionTable.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Estación de Larrabide

A continuación voy a mostrar el trabajo realizado en cada una de las fases del proceso del descubrimiento del conocimiento KDD, con el objetivo de poder predecir las temperaturas, velocidad media del viento y precipitación acumulada a 24h.

Integración y recopilación de los datos:

La integración de los datos se ha realizado de la misma forma que para la estación de Montes de Cierzo pero con los datos de la estación meteorológica de Larrabide(Pamplona).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección, Limpieza y transformación de los datos:

En esta sub fase se tiene en cuenta diferentes aspectos como:

Limpieza de datos:

Tratamiento de valores perdidos (missing values)

- Aplicando el filtro ReplaceMissingValues el cual sustituye todos los valores no encontrados (missing) de un atributo por la moda y la media de ese atributo.

Tratamiento de valores anómalos (outlier values):

- Mediante el filtro IntercuatileRange detectamos los valores outlier (anómalos) así como los valores extremos. Pero debido a que al eliminar estos valores los métodos obtienen peores resultados hemos decidido no aplicar este filtro.

Trasformación de datos:

Discretizacion y numeración:

- Como he comentado anteriormente he discretizado el atributo numérico precipitación acumulada a 24 h en tres valores nominales (1, 2,3).

Normalización de rango (escalado):

- Al ser todos mis atributos numéricos y cada uno de ellos con un rango de valores muy diferentes he decidido normalizarlos para comprender mejor los atributos. Para ello he utilizado el filtro llamado Normalize el cual ha normalizado los valores entre 0 y 1.

Reducción de la dimensión:

- He eliminado dos atributos llamados **velocidad media viento km/h** y dirección **de la racha máxima km/h** ya que tenía otros dos atributos iguales pero medidos en m/s. Con los atributos y muestras que dispongo no es necesaria en mi caso ya que Weka puede tratarlos bien.

Aumento de la dimensión:

- Probé a introducir nuevos atributos derivados de los ya existentes como la diferencia entre la temperatura máxima y la mínima pero sin embargo los resultados no mejoraban por lo que no introduje nuevos atributos derivados. A demás al introducir nuevos atributos aumentaba el tiempo de cómputo y los resultados no mejoraban.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de datos:

Selección de las características (atributos) más importantes:

Weka incorpora una gran cantidad de métodos para estudiar la relevancia de atributos y realizar una selección automática de los mismos. Para utilizar estos métodos una vez cargados los datos es necesario elegir uno de estos métodos selección en la pestaña del explorador llamada select atributes. Hay dos tipos de evaluadores principales: los SubsetEval (evaluadores de subconjuntos o selectores) y AtributteEval (Prorreteadores de atributos). Los primeros tienen que elegir una estrategia o método de búsqueda de subconjuntos (Search Method) para encontrar el mejor subconjunto de atributos que nos proporcione los mejores resultados para un modelo de clasificación. Como resultado tendremos aquellos atributos con los cuales se ha realizado una mejor clasificación. El segundo tipo de evaluadores solo pueden combinarse con un Ranker ya que no seleccionan los atributos sino que los ordenan por su relevancia a la hora de clasificar el atributo objetivo.

A continuación muestro los resultados obtenidos del proceso de selección de atributos para los datos de la estación de Pamplona (Larrabide):

En primer lugar voy a enumerar todos los atributos que dispongo para realizar las predicciones/clasificaciones:

Atributo	Numero	Atributo	Numero
día	**1**	**Precipitación acumulada**	**10**
mes	**2**	**Velocidad media del viento**	**11**
año	**3**	**Dirección del viento**	**12**
Temperatura media	**4**	**Velocidad de la racha máxima**	**13**
Temperatura máxima	**5**	**Dirección de la racha máxima**	**14**
Temperatura mínima	**6**	**Radiación global**	**15**
Humedad media	**7**	**insolación**	**16**
Humedad máxima	**8**	**Presión atmosférica media**	**17**
Humedad mínima	**9**		

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura máxima:

Aplicando evaluadores de subconjuntos:

Atributo	**Evaluador de atributos**	**Modelo**	**Estrategia de búsqueda**	**Atributos seleccionados**
Temperatura máxima a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	1,2,4,5,6,9,11,13,15 ,16, 17
Temperatura máxima a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,4,5,6,9,11,13, 15,16,17
Temperatura máxima a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,4,6,9,10,13,14,15, 16,17
Temperatura máxima a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,4,6,9,10,12,13,15, 16,17

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son prácticamente iguales salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 2(mes), 4(temperatura media), 6(temperatura mínima), 9(Humedad Mínima), 13(Velocidad de la racha máxima), 15(Radiación global), 16(Insolación), 17(Presión atmosférica).

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

ReliefAttributeEval:

Ranked attributes:

0.010691 2 mes
0.004089 5 tem_max
0.00405 4 tem_med
0.00268 6 tem_min
0.002303 3 ano
0.001468 1 dia
0.000884 15 radiacion_global
0.000246 10 precip_acumulada
-0.000155 14 direc_racha_max
-0.000499 13 vel_racha_max
-0.000692 17 presion_ats_media
-0.001216 8 humedad_max
-0.001732 9 humedad_min
-0.001746 7 humedad_med
-0.00189 16 insolacion
-0.001899 11 vel_media_viento
-0.00367 12 direc_viento

Selected attributes: 2,5,4,6,3,1,15,10,14,13,17,8,9,7,16,11,12 : 17

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura máxima a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el mes, las temperaturas y el año.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura mínima:

Aplicando evaluadores de subconjuntos:

Atributo	**Evaluador de atributos**	**Modelo**	**Estrategia de búsqueda**	**Atributos seleccionados**
Temperatura mínima a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	2,3,4,6,7,8,9,12,15, 16
Temperatura mínima a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,4,6,7,8,9,12,15,16 ,17
Temperatura mínima a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	1,2,4,5,6,7,9,10,11, 12,13,14,15, 16,17
Temperatura mínima a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,3,4,6,7,9,11,12,13 ,15,16,17

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son prácticamente iguales salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 2, 4, 6, 7, 9, 12, 15,16.

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

 0.0092634   4 tem_med
 0.0083826   6 tem_min
 0.0081206   2 mes
 0.0057843   5 tem_max
 0.004287    3 ano
 0.0029983  17 presion_ats_media
 0.0029284   1 dia
 0.001081   10 precip_acumulada
 0.0010397  13 vel_racha_max
 0.0009309  11 vel_media_viento
 0.0007254   8 humedad_max
-0.0000841   7 humedad_med
-0.0002532  14 direc_racha_max
-0.0005902   9 humedad_min
-0.0008529  16 insolacion
-0.0014878  12 direc_viento
-0.0015166  15 radiacion_global

Selected attributes: 4,6,2,5,3,17,1,10,13,11,8,7,14,9,16,12,15 : 17
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura mínima a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son las temperaturas media y mínima, el mes, la temperatura máxima el año y la presión.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura media:

Aplicando evaluadores de subconjuntos:

Atributo	**Evaluador de atributos**	**Modelo**	**Estrategia de búsqueda**	**Atributos seleccionados**
Temperatura media a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	2,4,5,6,7,8,9,11,13, 15,16,17
Temperatura media a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,4,5,6,7,8,9,11,13, 15,16,17
Temperatura media a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,4,5,6,9,13,15, 16,17
Temperatura media a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,4,5,6,9,13,15, 16,17

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son prácticamente iguales salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 2, 4, 5, 6, 9, 13, 15, 16,17.

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

Ranked attributes:

0.0118613 2 mes
0.0086337 4 tem_med
0.0067844 6 tem_min
0.0061183 5 tem_max
0.0044153 3 ano
0.0030759 1 dia
0.0015906 10 precip_acumulada
0.0012667 17 presion_ats_media
0.0005632 13 vel_racha_max
0.0000879 15 radiacion_global
-0.0003146 8 humedad_max
-0.0003403 14 direc_racha_max
-0.0009418 7 humedad_med
-0.0009778 11 vel_media_viento
-0.0016635 16 insolacion
-0.0016958 9 humedad_min
-0.0018523 12 direc_viento

Selected attributes: 2,4,6,5,3,1,10,17,13,15,8,14,7,11,16,9,12 : 17

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura media a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el mes, las temperaturas, el año, humedad mínima y radiación.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Velocidad media del viento a 24h:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Velocidad media del viento a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	3,11,12,13,17
Velocidad media del viento a 24h	WrapperSubsetEval	M5Rules	BestFirst	3,11,12,13,17
Velocidad media del viento a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,4,5,6,7,8,11,12,13,16,17
Velocidad media del viento a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	11,13

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son parecidos. Además todos estos métodos de selección comparten como atributos seleccionados los atributos (11,13).

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

 0.013285  11 vel_media_viento
 0.009581  17 presion_ats_media
 0.007852  13 vel_racha_max
 0.005591  10 precip_acumulada
 0.005103   1 dia
 0.004233   3 ano
 0.003044   9 humedad_min
 0.002506   6 tem_min
 0.002307   8 humedad_max
 0.002165   2 mes
 0.002112  16 insolacion
 0.002001   4 tem_med
 0.001731   5 tem_max
 0.001486   7 humedad_med
 0.001118  14 direc_racha_max
 0.000674  15 radiacion_global
-0.000123  12 direc_viento

Selected attributes: 11,17,13,10,1,3,9,6,8,2,16,4,5,7,14,15,12 : 17
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (velocidad media del viento a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son la velocidad media del viento, la presión atmosférica, velocidad de la racha máxima y la precipitación acumulada.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Precipitación acumulada a 24h:

Aplicando evaluadores de subconjuntos:

Atributo	**Evaluador de atributos**	**Modelo**	**Estrategia de búsqueda**	**Atributos seleccionados**
Precipitación acumulada a 24h	WrapperSubsetEval	J48	GeneticSearch	10,13,14,17
Precipitación acumulada a 24h	WrapperSubsetEval	J48	BestFirst	6,10,14,17
Precipitación acumulada a 24h	WrapperSubsetEval	Logistic	GeneticSearch	1,2,5,6,10,11,13,14, 15,17
Precipitación acumulada a 24h	WrapperSubsetEval	Logistic	BestFirst	1,3,5,6,8,9,10,11,13 ,14,15,17

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son parecidos. Además todos estos métodos de selección comparten como atributos seleccionados los atributos (10,14, 17).

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:
 0.02508   3 ano
 0.01896   1 dia
 0.01679   2 mes
 0.01462  17 presion_ats_media
 0.01234  12 direc_viento
 0.01063  16 insolacion
 0.00828  15 radiacion_global
 0.00742   4 tem_med
 0.00729   6 tem_min
 0.00667   8 humedad_max
 0.00649   5 tem_max
 0.00603  13 vel_racha_max
 0.00549   7 humedad_med
 0.0054   11 vel_media_viento
 0.00414   9 humedad_min
 0.00382  10 precip_acumulada
 0.00182  14 direc_racha_max

Selected attributes: 3,1,2,17,12,16,15,4,6,8,5,13,7,11,9,10,14 : 17
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (precipitación acumulada a 24h). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el año día, mes, presión atmosférica, dirección del viento e insolación.

Tras la aplicación de diferentes métodos para seleccionar estos atributos he decidido quedar con los atributos seleccionados mediante el evaluador de atributos WrapperSubsetEval (evaluador de subconjuntos envolvente) basado en el clasificador M5Rules y con un método de búsqueda genético (GeneticSearch). Esta decisión es debido a los algoritmos WrapperSubsetEval son capaces de escoger mejor aquellos los atributos claves para la clasificación (el punto negativo es que tardan mucho tiempo).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En esta fase voy a mostrar los resultados obtenidos de aplicar los diferentes métodos de minería de datos a la tarea de predicción de los atributos. Como método de evaluación he utilizado una validación cruzada con 10 pliegues. En este método los datos se dividen aleatoriamente en n (pliegues) grupos. Un grupo se reserva para el conjunto de prueba y los n-1 restantes (juntando sus datos) se construye un modelo y se usa para predecir el resultado de los datos del grupo reservado. Este proceso se repite n veces dejando cada vez un grupo diferente para la prueba. Esto quiere decir que se calculan n ratios de error independientes. Finalmente se construye un modelo con todos los datos y se obtienen sus ratios de error y precisión promediando los n ratios de error disponibles.

Además en esta fase se aplicara las técnicas de minería de datos a todos los atributos de entrada y a los atributos seleccionados en la fase anterior y así realizar una comparativa de los resultados obtenidos.

A la hora de realizar la predicción de las temperaturas, la velocidad y la precipitación acumulada hemos utilizado estos métodos ya que son los que mejores resultados nos han ofrecido:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos de entrada:

Resultados obtenidos con los diferentes métodos:

- **LeastMedSq (-S 1 -G 0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_max_24h = 0.2613 * dia + 0.4641 * mes +-0.1384 * ano +45.0931 * tem_med +-3.8334 * tem_max +-9.2194 * tem_min +3.0682 * humedad_med +-2.2387 * humedad_max +-0.1006 * humedad_min +2.4638 * precip_acumulada +-0.3139 * vel_media_viento +-0.4621 * direc_viento +-5.984 * vel_racha_max +0.4589 * direc_racha_max +9.0565 * radiacion_global +-2.8529 * insolacion +5.7314 * presion_ats_media +-1.5401

Time taken to build model: 1.97 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.916
Mean absolute error	2.6413
Root mean squared error	3.3601
Relative absolute error	37.5014 %
Root relative squared error	40.1073 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_max_24h = 0.4617 * mes +44.8668 * tem_med + -3.3258 * tem_max +-9.3301 * tem_min +2.8867 * humedad_med +-2.2048 * humedad_max +2.5361 * precip_acumulada +-0.4254 * direc_viento +-6.2373 * vel_racha_max +8.9581 * radiacion_global +-2.8698 * insolacion +5.6696 * presion_ats_media +-1.4159

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9159
Mean absolute error	2.6443
Root mean squared error	3.3628
Relative absolute error	37.5439 %
Root relative squared error	40.1398 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Linear Regression Model

tem_max_24h = 0.0072 * dia + 0.0114 * mes + 1.0731 * tem_med + -0.0816 * tem_max +-0.2131 * tem_min + 0.0627 * humedad_med + -0.0528 * humedad_max + 0.0646 * precip_acumulada + -0.0106 * direc_viento +-0.1537 * vel_racha_max +0.2183 * radiacion_global +-0.0701 * insolacion +0.1374 * presion_ats_media + -0.0066

Time taken to build model: 4.79 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.914
Mean absolute error	2.65
Root mean squared error	3.38
Relative absolute error	37.901 %
Root relative squared error	40.5772 %
Total Number of Instances	3288

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

M5 pruned model tree:
tem_med <= 0.464: LM1 (1526/33.437%)
tem_med > 0.464 : LM2 (1742/43.416%)

LM num: 1

tem_max_24h = 0.0045 * mes + 34.7618 * tem_med + 6.8226 * tem_max - 10.1334 * tem_min + 1.4931 * humedad_med - 2.9172 * humedad_max + 2.5366 * precip_acumulada - 0.0041 * direc_viento - 4.8938 * vel_racha_max + 10.2856 * radiacion_global - 5.4841 * insolacion + 3.7083 * presion_ats_media + 1.9045

LM num: 2

tem_max_24h = 0.4716 * dia + 2.356 * mes + 51.2935 * tem_med - 9.9984 * tem_max - 9.6288 * tem_min + 2.2502 * humedad_med - 1.167 * humedad_max + 2.1122 * humedad_min + 0.0217 * precip_acumulada - 3.9857 * vel_media_viento - 0.0036 * direc_viento - 4.7189 * vel_racha_max + 8.7983 * radiacion_global - 2.0307 * insolacion + 12.0545 * presion_ats_media - 7.7511

Number of Rules: 2

Time taken to build model: 3.84 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9185
Mean absolute error	2.6133
Root mean squared error	3.312
Relative absolute error	37.1043 %
Root relative squared error	39.5337 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

tem_med <= 0.464: LM1 (1532/33.901%)
tem_med > 0.464 : LM2 (1736/43.601%)

LM num: 1

tem_max_24h = 0.0041 * mes + 25.9907 * tem_med + 14.2151 * tem_max - 8.0498 * tem_min + 0.043 * humedad_med - 2.8649 * humedad_max + 1.8668 * humedad_min + 1.8899 * precip_acumulada + 1.7477 * vel_media_viento - 5.9036 * vel_racha_max + 9.6139 * radiacion_global - 5.6769 * insolacion + 3.1744 * presion_ats_media + 1.8194

LM num: 2

tem_max_24h = 1.6922 * mes + 58.1032 * tem_med - 15.6707 * tem_max - 11.6401 * tem_min + 5.6767 * humedad_med - 2.5984 * humedad_max + 0.0141 * precip_acumulada - 4.3127 * vel_media_viento - 3.9737 * vel_racha_max + 7.3498 * radiacion_global - 0.0195 * insolacion + 12.1641 * presion_ats_media - 7.0363

Number of Rules: 2

Time taken to build model: 22.97 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9195
Mean absolute error	2.5986
Root mean squared error	3.293
Relative absolute error	36.8955 %
Root relative squared error	39.3072 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

  A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

  - **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

  - **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

  - **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

  - **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

  - **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	25.18 s	0.908	2.8199	3.5668
500	0.2	10	0.01	28.86 s	0.9116	2.7387	3.4854
500	0.2	15	0.01	29.43 s	0.9136	2.7174	3.443
500	0.01	15	0.01	29.65 s	0.921	2.5671	3.2666
3000	0.01	15	0.01	180.66 s	0.9246	2.519	3.1937

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 5.000

Time taken to build model: 760.89 seconds
=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9257
Mean absolute error	2.5019
Root mean squared error	3.1714
Relative absolute error	35.5228 %
Root relative squared error	37.8558 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_max_24h =0.2604 * dia +0.488 * mes + 39.921 * tem_med +-0.7161 * tem_max +-6.7872 * tem_min +1.675 * humedad_min +-0.1921 * vel_media_viento +-5.7227 * vel_racha_max +9.2176 * radiacion_global +-3.0216 * insolacion +5.8392 * presion_ats_media +-2.72

Time taken to build model: 1.49 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient	0.9165
Mean absolute error	2.6356
Root mean squared error	3.3505
Relative absolute error	37.4267 %
Root relative squared error	39.9977 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_max_24h =0.4786 * mes +39.3832 * tem_med +-6.7636 * tem_min +1.7389 * humedad_min +-5.8519 * vel_racha_max +9.1199 * radiacion_global +-3.0043 * insolacion +5.7567 * presion_ats_media +-2.6292

Time taken to build model: 0.03 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient	0.9165
Mean absolute error	2.636
Root mean squared error	3.3512
Relative absolute error	37.4316 %
Root relative squared error	40.0069 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 2.91 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient	0.9
Mean absolute error	2.9269
Root mean squared error	3.6514
Relative absolute error	41.5631 %
Root relative squared error	43.5907 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

tem_med <= 0.464: LM1 (1526/33.567%)
tem_med > 0.464 : LM2 (1742/43.425%)

LM num: 1

tem_max_24h = 0.0047 * mes + 32.9104 * tem_med + 7.001 * tem_max - 8.7395 * tem_min + 0.0169 * humedad_min - 4.5085 * vel_racha_max + 10.2438 * radiacion_global - 5.5632 * insolacion + 3.7211 * presion_ats_media + 0.6004

LM num: 2

tem_max_24h = 0.4797 * dia + 2.4193 * mes + 47.7762 * tem_med - 7.9696 * tem_max - 7.9835 * tem_min + 3.4068 * humedad_min - 4.05 * vel_media_viento - 4.799 * vel_racha_max + 8.881 * radiacion_global - 2.0945 * insolacion + 12.0622 * presion_ats_media - 7.9946

Number of Rules: 2
Time taken to build model: 2.6 seconds

=== Evaluation on training set ===

=== Summary ===

Correlation coefficient	0.9203
Mean absolute error	2.5893
Root mean squared error	3.278
Relative absolute error	36.7681 %
Root relative squared error	39.132 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

tem_med <= 0.464: LM1 (1532/34.051%)
tem_med > 0.464 : | tem_max <= 0.648 : LM2 (1005/40.007%)| tem_max > 0.648 : LM3 (731/46.349%)

LM num: 1

tem_max_24h = 0.0042 * mes + 26.1034 * tem_med + 12.3187 * tem_max - 6.8231 * tem_min + 0.0218 * humedad_min + 2.3602 * vel_media_viento - 6.2667 * vel_racha_max + 9.5083 * radiacion_global - 5.8873 * insolacion + 3.312 * presion_ats_media + 0.6891

LM num: 2

tem_max_24h = 0.7634 * dia + 1.4744 * mes + 47.4199 * tem_med - 11.9539 * tem_max - 8.1762 * tem_min + 4.2381 * humedad_min - 5.3736 * vel_media_viento - 0.1088 * vel_racha_max + 10.0583 * radiacion_global - 2.1925 * insolacion + 12.5119 * presion_ats_media - 7.3294

LM num: 3

tem_max_24h = 7.3952 * mes + 43.2836 * tem_med - 15.1355 * tem_max - 0.2018 * tem_min + 0.0858 * humedad_min - 11.7794 * vel_media_viento - 5.7724 * vel_racha_max + 9.2723 * radiacion_global - 0.0207 * insolacion + 14.856 * presion_ats_media - 7.9255

Number of Rules: 3
Time taken to build model: 20.8 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient	0.9232
Mean absolute error	2.5419
Root mean squared error	3.2195
Relative absolute error	36.0954 %
Root relative squared error	38.4339 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

  A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

  - **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

  - **hiddenLayers** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

  - **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

  - **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

  - **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	16.12 s	0.9163	2.6809	3.3817
500	0.2	10	0.01	16.04 s	0.9188	2.6297	3.3288
500	0.2	15	0.01	24.43 s	0.9168	2.6701	3.3754
500	0.01	15	0.01	20.06 s	0.9164	2.64	3.3592
3000	0.01	15	0.01	125.55 s	0.9262	2.4972	3.1594

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 5.000

Los valores obtenidos son:

Time taken to build model: 548.69 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9274
Mean absolute error	2.4778
Root mean squared error	3.1349
Relative absolute error	35.1808 %
Root relative squared error	37.4198 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la temperatura media a 24h:

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.0441 * dia +0.5005 * mes +-0.2222 * ano +45.3834 * tem_med + - 6.8615 * tem_max + - 9.0511 * tem_min +2.8218 * humedad_med +-1.3373 * humedad_max +1.2101 * humedad_min +0.7733 * precip_acumulada +0.4289 * vel_media_viento +-0.0328 * direc_viento +-3.2434 * vel_racha_max +0.4865 * direc_racha_max +5.9235 * radiacion_global +-2.6396 * insolacion +2.2754 * presion_ats_media + - 4.3066

Time taken to build model: 1.69 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9493
Mean absolute error	1.6249
Root mean squared error	2.0946
Relative absolute error	28.8204 %
Root relative squared error	31.4153 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.4966 * mes +-0.2152 * ano +45.3715 * tem_med +-7.0188 * tem_max + -8.892 * tem_min +2.8574 * humedad_med +-1.3288 * humedad_max +1.2495 * humedad_min +-2.8159 * vel_racha_max +5.9422 * radiacion_global +-2.6274 * insolacion +2.218 * presion_ats_media +-4.2798

Time taken to build model: 0.1 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9494
Mean absolute error	1.6248
Root mean squared error	2.0941
Relative absolute error	28.8181 %
Root relative squared error	31.4077 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 3.96 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0 9424
Mean absolute error	1.7557
Root mean squared error	2.2305
Relative absolute error	31.1404 %
Root relative squared error	33.4532 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:

tem_med <= 0.504: LM1 (1727/30.433%)
tem_med > 0.504 : LM2 (1541/30.711%)

LM num: 1

tem_med_24h = -0.2625 * dia + 0.2987 * mes - 0.0019 * ano + 39.1892 * tem_med - 2.3255 * tem_max - 8.2868 * tem_min + 0.0246 * humedad_med - 0.0114 * humedad_max + 1.7707 * humedad_min - 2.4167 * vel_racha_max + 6.3543 * radiacion_global - 4.0278 * insolacion + 1.0913 * presion_ats_media - 2.3718

LM num: 2

tem_med_24h = 0.4343 * dia + 2.3142 * mes - 0.0021 * ano + 50.267 * tem_med - 11.2005 * tem_max - 11.2048 * tem_min + 4.5253 * humedad_med - 2.4261 * humedad_max + 1.0361 * humedad_min - 1.5033 * vel_media_viento - 2.5456 * vel_racha_max + 5.7881 * radiacion_global - 2.2664 * insolacion + 6.7063 * presion_ats_media - 7.4215

Number of Rules: 2
Time taken to build model: 3.87 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9509
Mean absolute error	1.6108
Root mean squared error	2.0642
Relative absolute error	28.5692 %
Root relative squared error	30.9586 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
tem_med <= 0.481: LM1 (1621/29.016%)
tem_med > 0.481 : LM2 (1647/32.381%)

LM num: 1

tem_med_24h = 0.221 * mes - 0.2264 * ano + 31.3191 * tem_med + 2.6887 * tem_max - 4.5614 * tem_min - 0.9059 * humedad_med - 0.0185 * humedad_max + 2.7818 * humedad_min + 0.6655 * vel_media_viento - 3.0871 * vel_racha_max + 6.2037 * radiacion_global - 3.6853 * insolacion + 0.0176 * presion_ats_media - 2.1147

LM num: 2

tem_med_24h = 2.1256 * mes - 0.0016 * ano + 52.9294 * tem_med - 12.8002 * tem_max - 12.1129 * tem_min + 6.3042 * humedad_med - 3.19 * humedad_max + 0.0076 * humedad_min - 1.3504 * vel_media_viento - 2.2503 * vel_racha_max + 5.6616 * radiacion_global - 1.9014 * insolacion + 6.7782 * presion_ats_media - 7.5408

Number of Rules: 2

Time taken to build model: 22.07 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9514
Mean absolute error	1.6042
Root mean squared error	2.0531
Relative absolute error	28.4524 %
Root relative squared error	30.7921 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	21.27 s	0.9464	1.7079	2.1819
500	0.2	10	0.01	20.88 s	0.9475	1.6834	2.157
500	0.2	15	0.01	31.24 s	0.9454	1.7117	2.2106
500	0.01	15	0.01	30.19 s	0.9503	1.6191	2.0775
3000	0.01	15	0.01	176.72s	0.9542	1.5366	1.9944

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 5.000

Los valores obtenidos son:

Time taken to build model: 621.33 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9554
Mean absolute error	1.5341
Root mean squared error	1.9704
Relative absolute error	27.2102 %
Root relative squared error	29.552 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.4991 * mes +45.3251 * tem_med +-6.7931 * tem_max +-9.025 * tem_min +2.8401 * humedad_med +-1.3557 * humedad_max +1.1801 * humedad_min +0.3299 * vel_media_viento +-3.073 * vel_racha_max +5.924 * radiacion_global +-2.7276 * insolacion +2.2291 * presion_ats_media +-4.3108

Time taken to build model: 1.42 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9494
Mean absolute error	1.623
Root mean squared error	2.093
Relative absolute error	28.786 %
Root relative squared error	31.3906 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.5042 * mes +45.3553 * tem_med +-7.0016 * tem_max +-8.8983 * tem_min +2.7836 * humedad_med +-1.3837 * humedad_max +1.2018 * humedad_min +-2.8341 * vel_racha_max +5.9538 * radiacion_global +-2.713 * insolacion +2.2641 * presion_ats_media +-4.2744

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9494
Mean absolute error	1.6242
Root mean squared error	2.0942
Relative absolute error	28.8082 %
Root relative squared error	31.4093 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 3.96 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0 9424
Mean absolute error	1.7557
Root mean squared error	2.2305
Relative absolute error	31.1404 %
Root relative squared error	33.4532 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

M5 pruned model tree:
tem_med <= 0.504: LM1 (1727/30.455%)
tem_med > 0.504 : LM2 (1541/30.767%)

LM num: 1

tem_med_24h = 0.2839 * mes + 39.3446 * tem_med - 2.3888 * tem_max - 8.2942 * tem_min + 0.024 * humedad_med - 0.0119 * humedad_max + 1.7813 * humedad_min - 2.4139 * vel_racha_max + 6.3324 * radiacion_global - 4.0097 * insolacion + 1.1532 * presion_ats_media - 2.5685

LM num: 2

tem_med_24h = 2.0412 * mes + 50.5875 * tem_med - 11.1669 * tem_max - 11.2672 * tem_min + 4.7747 * humedad_med - 2.4096 * humedad_max + 0.9848 * humedad_min - 1.2795 * vel_media_viento - 2.6564 * vel_racha_max + 5.3955 * radiacion_global - 1.995 * insolacion + 6.6973 * presion_ats_media - 7.3091

Number of Rules: 2
Time taken to build model: 3.45 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9507
Mean absolute error	1.6126
Root mean squared error	2.0676
Relative absolute error	28.6018 %
Root relative squared error	31.0106 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

tem_med <= 0.481: LM1 (1621/29.036%)
tem_med > 0.481 : | tem_med <= 0.663 : LM2 (984/29.769%)|
tem_med > 0.663 : LM3 (663/34.548%)

LM num: 1

tem_med_24h = 0.2214 * mes + 31.1394 * tem_med + 2.796 * tem_max - 4.4431 * tem_min - 1.1522 * humedad_med - 0.0195 * humedad_max + 2.8009 * humedad_min + 0.6428 * vel_media_viento - 3.1128 * vel_racha_max + 6.1797 * radiacion_global - 3.7806 * insolacion + 0.0183 * presion_ats_media - 2.0463

LM num: 2

tem_med_24h = 2.3322 * mes + 49.346 * tem_med - 9.6721 * tem_max - 12.2285 * tem_min + 0.1283 * humedad_med - 0.0669 * humedad_max + 3.3321 * humedad_min - 0.0204 * vel_media_viento - 2.5428 * vel_racha_max + 6.6599 * radiacion_global - 2.9305 * insolacion + 5.7804 * presion_ats_media - 7.0776

LM num: 3

tem_med_24h = 2.8714 * mes + 53.9318 * tem_med - 14.9107 * tem_max - 11.2916 * tem_min + 9.0613 * humedad_med - 4.9018 * humedad_max + 0.0077 * humedad_min - 5.5389 * vel_media_viento - 2.3738 * vel_racha_max + 3.3352 * radiacion_global - 0.0623 * insolacion + 8.9263 * presion_ats_media - 7.8832

Time taken to build model:
17.33 seconds

=== Cross-validation ===

Correlation coefficient	0.9513
Mean absolute error	1.6042
Root mean squared error	2.0553
Relative absolute error	28.4524 %
Root relative squared error	30.825 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	15.94s	0.9469	1.7046	2.1736
500	0.2	10	0.01	2.1736 s	0.9484	1.6813	2.1473
500	0.2	15	0.01	34.22 s	0.9483	1.6845	2.1493
500	0.01	15	0.01	31.97 s	0.9489	1.6405	2.1048
3000	0.01	15	0.01	201.13 s	0.9546	1.5456	1.9854

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 5.000

Los valores obtenidos son:

Time taken to build model: 537.73 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9549
Mean absolute error	1.5448
Root mean squared error	1.9786
Relative absolute error	27.3998 %
Root relative squared error	29.675 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la temperatura mínima a 24h:

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =-0.1129 * dia +0.588 * mes +-0.2172 * ano +34.7346 * tem_med +-1.8769 * tem_max +-3.5485 * tem_min +3.8837 * humedad_med +-0.8644 * humedad_max +1.9962 * humedad_min +-0.4496 * precip_acumulada +1.5509 * vel_media_viento +0.3179 * direc_viento +-1.0514 * vel_racha_max +0.7311 * direc_racha_max +2.7784 * radiacion_global +-2.4351 * insolacion +-0.4018 * presion_ats_media +-8.0821

Time taken to build model: 3.93 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9392
Mean absolute error	1.5628
Root mean squared error	1.9834
Relative absolute error	32.3301 %
Root relative squared error	34.3299 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =0.5439 * mes +-0.1722 * ano +34.6541 * tem_med +-1.9595 * tem_max +-3.2251 * tem_min +3.6633 * humedad_med +-0.7419 * humedad_max +2.0681 * humedad_min +1.4163 * vel_media_viento +0.3844 * direc_viento +-0.9898 * vel_racha_max +2.8374 * radiacion_global +-2.5129 * insolacion +-8.4658

Time taken to build model: 0.1 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9391
Mean absolute error	1.5635
Root mean squared error	1.9848
Relative absolute error	32.3456 %
Root relative squared error	34.3532 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 4.95 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9284
Mean absolute error	1.714
Root mean squared error	2.1457
Relative absolute error	35.458 %
Root relative squared error	37.1389 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:

tem_med <= 0.522: LM1 (1814/37.179%)
tem_med > 0.522 : LM2 (1454/28.284%)

LM num: 1

tem_min_24h = -0.3994 * dia + 0.4009 * mes - 0.257 * ano + 34.4219 * tem_med - 4.6077 * tem_max - 3.1506 * tem_min - 1.1142 * humedad_med + 1.9354 * humedad_max + 3.5762 * humedad_min + 0.0116 * vel_media_viento + 0.58 * direc_viento - 0.0081 * vel_racha_max + 2.2725 * direc_racha_max + 3.4656 * radiacion_global - 2.9803 * insolacion - 7.132

LM num: 2

tem_min_24h = 0.3268 * dia + 2.6108 * mes - 0.0018 * ano + 35.8469 * tem_med - 2.0705 * tem_max - 4.8057 * tem_min + 5.9059 * humedad_med - 2.3988 * humedad_max + 1.7147 * humedad_min + 0.0145 * vel_media_viento + 0.0039 * direc_viento - 0.7192 * vel_racha_max + 3.6265 * radiacion_global - 2.9712 * insolacion + 1.4277 * presion_ats_media - 10.1802

Number of Rules: 2
Time taken to build model: 3.7 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9405
Mean absolute error	1.5412
Root mean squared error	1.9618
Relative absolute error	31.8827 %
Root relative squared error	33.9557 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
tem_med <= 0.522: LM1 (1820/36.748%)
tem_med > 0.522 : LM2 (1448/29.161%)

LM num: 1

tem_min_24h = 0.3734 * mes - 0.2575 * ano + 29.7951 * tem_med - 1.4589 * tem_max - 0.8952 * tem_min - 0.8732 * humedad_med + 1.7445 * humedad_max + 3.7254 * humedad_min + 1.1859 * vel_media_viento + 0.3792 * direc_viento - 1.4063 * vel_racha_max + 2.5926 * direc_racha_max + 3.6574 * radiacion_global - 2.9894 * insolacion - 0.7561 * presion_ats_media - 7.2393

LM num: 2

tem_min_24h = 2.4521 * mes + 0.2059 * ano + 37.2474 * tem_med - 2.1532 * tem_max - 6.3129 * tem_min + 6.6686 * humedad_med - 3.1382 * humedad_max + 1.6199 * humedad_min + 0.0127 * vel_media_viento + 0.0028 * direc_viento - 0.0106 * vel_racha_max - 1.2399 * direc_racha_max + 3.4913 * radiacion_global - 2.8846 * insolacion + 2.2642 * presion_ats_media - 10.4574

Number of Rules: 2
Time taken to build model: 36.32 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9406
Mean absolute error	1.5435
Root mean squared error	1.9611
Relative absolute error	31.931 %
Root relative squared error	33.944 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	24.03 s	0.9341	1.6118	2.0736
500	0.2	10	0.01	24.1 s	0.9354	1.5982	2.0555
500	0.2	15	0.01	24.7 s	0.93.43	1.5792	2.0734
500	0.001	15	0.01	33.97 s	0.03.89	156.54	1.98.75
3000	0.01	15	0.01	207.77s	0.9427	1.4803	1.9275

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 5.000

Los valores obtenidos son:

Time taken to build model: 561.95 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9458
Mean absolute error	1.4702
Root mean squared error	1.8753
Relative absolute error	30.4137 %
Root relative squared error	32.4585 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos seleccionados

- **LeastMedSq (-S 1 -G 0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =0.6289 * mes +-0.1935 * ano +30.7936 * tem_med +-1.4052 * tem_min +2.3146 * humedad_med +-0.5637 * humedad_max +2.9781 * humedad_min +0.2912 * direc_viento +3.0095 * radiacion_global +-2.5915 * insolacion +-8.1694

Time taken to build model: 1.71 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9392
Mean absolute error	1.5631
Root mean squared error	1.9827
Relative absolute error	32.3357 %
Root relative squared error	34.3175 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =0.5686 * mes +-0.1741 * ano +30.8047 * tem_med +-1.2219 * tem_min +2.2426 * humedad_med +-0.4599 * humedad_max +3.0735 * humedad_min +0.3136 * direc_viento +2.9732 * radiacion_global +-2.5598 * insolacion +-8.3406

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9391
Mean absolute error	1.5634
Root mean squared error	1.9843
Relative absolute error	32.3426 %
Root relative squared error	34.3447 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 1.98 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9284
Mean absolute error	1.714
Root mean squared error	2.1457
Relative absolute error	35.458 %
Root relative squared error	37.1389 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

M5 pruned model tree:
tem_med <= 0.522: LM1 (1814/37.334%)
tem_med > 0.522 : LM2 (1454/28.419%)

LM num: 1

tem_min_24h = 0.392 * mes - 0.0014 * ano + 27.5443 * tem_med - 0.01 * tem_min - 2.7661 * humedad_med + 2.3783 * humedad_max + 4.6832 * humedad_min + 0.6752 * direc_viento + 3.3722 * radiacion_global - 3.2129 * insolacion - 7.5248

LM num: 2

tem_min_24h = 2.5001 * mes - 0.0018 * ano + 33.5489 * tem_med - 4.3706 * tem_min + 5.6222 * humedad_med - 2.4008 * humedad_max + 2.5623 * humedad_min + 0.0032 * direc_viento + 3.2676 * radiacion_global - 2.4669 * insolacion - 9.548

Number of Rules: 2
Time taken to build model: 3.36 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9407
Mean absolute error	1.542
Root mean squared error	1.9596
Relative absolute error	31.8997 %
Root relative squared error	33.9171 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
M5 pruned model tree:
    tem_med <= 0.522: LM1 (1820/36.885%)
    tem_med >  0.522 : LM2 (1448/29.318%)

    LM num: 1

tem_min_24h = 0.4184 * mes + 27.6055 * tem_med - 0.0112 * tem_min -
1.5549 * humedad_med + 1.6624 * humedad_max + 4.226 * humedad_min +
0.5323 * direc_viento + 3.8834 * radiacion_global - 3.3388 * insolacion -
7.6916

        LM num: 2

tem_min_24h = 2.6418 * mes + 34.2792 * tem_med - 5.5642 * tem_min +
6.1277 * humedad_med - 3.1834 * humedad_max + 2.6455 * humedad_min
+ 0.0022 * direc_viento + 3.6066 * radiacion_global - 2.5942 * insolacion -
9.2208

    Number of Rules: 2
    Time taken to build model: 16.42 seconds

    === Cross-validation ===
    === Summary ===

    Correlation coefficient                  0.9408
    Mean absolute error                      1.5407
    Root mean squared error                  1.9569
    Relative absolute error                 31.8727 %
    Root relative squared error             33.8713 %
    Total Number of Instances             3268
    Ignored Class Unknown Instances         20
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers**-- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	14.15 s	0.931	1.69	2.1308
500	0.2	10	0.01	13.91 s	0.932	1.6766	2.1143
500	0.2	15	0.01	22.83 s	0.9322	1.6767	2.1168
500	0.01	15	0.01	19.31 s	0.9384	1.5737	1.9964
3000	0.01	15	0.01	118.04 s	0.9423	1.5307	1.9331

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 5.000

Los valores obtenidos son:

Time taken to build model: 470.14 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9426
Mean absolute error	1.5219
Root mean squared error	1.9291
Relative absolute error	31.4836 %
Root relative squared error	33.3899 %
Total Number of Instances	3268
Ignored Class Unknown Instances	20

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la velocidad media del viento a 24h:

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-0.0427 * dia +-0.0678 * mes +-0.0433 * ano +0.0355 * tem_med +1.6008 * tem_max +-1.1725 * tem_min + 0.1359 * humedad_med +-0.1796 * humedad_max +0.4801 * humedad_min +-0.5563 * precip_acumulada +2.7351 * vel_media_viento +-0.1035 * direc_viento +0.6035 * vel_racha_max +-0.1593 * direc_racha_max +0.8415 * radiacion_global +-0.808 * insolacion +-0.6671 * presion_ats_media +1.7326

Time taken to build model: 2.78 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.4906
Mean absolute error	0.8176
Root mean squared error	1.0996
Relative absolute error	84.4323 %
Root relative squared error	90.1139 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-0.1018 * ano +1.2225 * tem_med +1.2569 * tem_max +-2.4447 * tem_min +0.753 * humedad_med +-0.4886 * humedad_max +0.1998 * humedad_min +-0.5607 * precip_acumulada +3.7733 * vel_media_viento +-0.1263 * direc_viento +1.5169 * vel_racha_max +0.3992 * radiacion_global +-0.6167 * insolacion +-0.5248 * presion_ats_media +1.6483

Time taken to build model: 0.03 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5095
Mean absolute error	0.8046
Root mean squared error	1.0496
Relative absolute error	83.0884 %
Root relative squared error	86.0161 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 1.38 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.4916
Mean absolute error	0.8124
Root mean squared error	1.0624
Relative absolute error	83.893 %
Root relative squared error	87.0653 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:
vel_racha_max <= 0.296: LM1 (1796/69.917%)
vel_racha_max > 0.296 : LM2 (1459/99.584%)

LM num: 1

vel_media_viento_24h = -0.1306 * mes - 0.1546 * ano + 0.7678 * tem_med + 2.3221 * tem_max - 2.6346 * tem_min - 0.8334 * humedad_med + 0.5829 * humedad_max + 0.8703 * humedad_min - 0.0046 * precip_acumulada + 3.1706 * vel_media_viento - 0.001 * direc_viento + 1.6808 * vel_racha_max + 0.0033 * radiacion_global - 0.6819 * insolacion - 1.5181 * presion_ats_media + 2.2758

LM num: 2

vel_media_viento_24h = -0.001 * ano + 0.0124 * tem_med + 1.8748 * tem_max - 1.8367 * tem_min + 1.752 * humedad_med - 0.9205 * humedad_max + 0.002 * humedad_min - 0.9894 * precip_acumulada + 3.7585 * vel_media_viento - 0.1678 * direc_viento + 1.2181 * vel_racha_max + 0.0041 * radiacion_global - 0.0063 * insolacion - 0.0053 * presion_ats_media + 1.1465

Number of Rules: 2
Time taken to build model: 3.6 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient	0.5346
Mean absolute error	0.7833
Root mean squared error	1.0308
Relative absolute error	80.9258 %
Root relative squared error	84.5119 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

```
M5 pruned model tree:
    vel_racha_max <= 0.296 : LM1 (1796/69.917%)
    vel_racha_max >  0.296 : LM2 (1459/99.584%)

LM num: 1

vel_media_viento_24h = -0.1306 * mes - 0.1546 * ano + 0.7678 * tem_med +
2.3221 * tem_max - 2.6346 * tem_min - 0.8334 * humedad_med + 0.5829 *
humedad_max + 0.8703 * humedad_min - 0.0046 * precip_acumulada +
3.1706 * vel_media_viento - 0.001 * direc_viento + 1.6808 * vel_racha_max +
0.0033 * radiacion_global - 0.6819 * insolacion - 1.5181 * presion_ats_media
+ 2.2758

LM num: 2

vel_media_viento_24h = -0.001 * ano + 0.0124 * tem_med + 1.8748 *
tem_max - 1.8367 * tem_min + 1.752 * humedad_med - 0.9205 *
humedad_max + 0.002 * humedad_min - 0.9894 * precip_acumulada +
3.7585 * vel_media_viento - 0.1678 * direc_viento + 1.2181 * vel_racha_max
+ 0.0041 * radiacion_global - 0.0063 * insolacion - 0.0053 *
presion_ats_media + 1.1465

    Number of Rules: 2
    Time taken to build model: 2.25 seconds

    === Cross-validation ===
    === Summary ===

    Correlation coefficient                  0.5132
    Mean absolute error                      0.7987
    Root mean squared error                  1.0473
    Relative absolute error                 82.4753 %
    Root relative squared error             85.8314 %
    Total Number of Instances                3255
    Ignored Class Unknown Instances            33
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	23.11 s	0.4164	0.8934	1.1822
500	0.2	10	0.01	23.64 s	0.4311	0.874	1.1537
500	0.2	15	0.01	32.81 s	0.4225	0.8808	1.1741
500	0.01	15	0.01	35.13 s	0.5153	0.7973	1.0467
2000	0.01	15	0.01	145.21	0.511	0.8011	1.0524

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 2.000

Los valores obtenidos son:

Time taken to build model: 302.71 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5215
Mean absolute error	0.7918
Root mean squared error	1.0422
Relative absolute error	81.7666 %
Root relative squared error	85.4098 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos de seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-0.0626 * ano +2.3355 * vel_media_viento +-0.1852 * direc_viento +0.8097 * vel_racha_max +-0.7226 * presion_ats_media +2.1249

Time taken to build model: 2.14 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.506
Mean absolute error	0.8184
Root mean squared error	1.0948
Relative absolute error	84.5098 %
Root relative squared error	89.7192 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-0.1044 * ano +3.2752 * vel_media_viento +-0.155 * direc_viento +1.5463 * vel_racha_max +-0.5232 * presion_ats_media +1.7775

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5087
Mean absolute error	0.8051
Root mean squared error	1.0501
Relative absolute error	83.1332 %
Root relative squared error	86.0551 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 0.48 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.4916
Mean absolute error	0.8124
Root mean squared error	1.0624
Relative absolute error	83.893 %
Root relative squared error	87.0653 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:

vel_racha_max <= 0.296: LM1 (1796/70.826%)
vel_racha_max > 0.296 : | vel_media_viento <= 0.445 : | | vel_media_viento <= 0.346 : LM2 (776/97.612%) | | vel_media_viento > 0.346 : LM3 (386/98.572%) | vel_media_viento > 0.445 : LM4 (297/103.19%)

LM num: 1

vel_media_viento_24h = -0.167 * ano + 1.992 * vel_media_viento - 0.0013 * direc_viento + 2.1716 * vel_racha_max - 1.4805 * presion_ats_media + 2.5179

LM num: 2

vel_media_viento_24h = -0.0011 * ano + 2.4066 * vel_media_viento - 0.0094 * direc_viento + 0.0583 * vel_racha_max - 0.0053 * presion_ats_media + 2.12

LM num: 3

vel_media_viento_24h = -0.0011 * ano + 0.1786 * vel_media_viento - 0.828 * direc_viento + 3.5855 * vel_racha_max - 0.0053 * presion_ats_media + 1.949

LM num: 4

vel_media_viento_24h = -0.0011 * ano + 4.2136 * vel_media_viento - 0.0155 * direc_viento + 0.0683 * vel_racha_max + 2.2645 * presion_ats_media + 0.0564

Time taken to build model: 1.65 seconds
=== Cross-validation ===

Correlation coefficient	0.5086
Mean absolute error	0.8021
Root mean squared error	1.0503
Relative absolute error	82.8272 %
Root relative squared error	86.0758 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

M5 pruned model tree:

vel_racha_max <= 0.296: LM1 (1796/70.826%)
vel_racha_max > 0.296 : | vel_media_viento <= 0.445 : | | vel_media_viento <= 0.346 : LM2 (776/97.612%)| | vel_media_viento > 0.346 : LM3 (386/98.572%)| vel_media_viento > 0.445 : LM4 (297/103.19%)

LM num: 1

vel_media_viento_24h = -0.167 * ano + 1.992 * vel_media_viento - 0.0013 * direc_viento + 2.1716 * vel_racha_max - 1.4805 * presion_ats_media + 2.517

LM num: 2

vel_media_viento_24h = -0.0011 * ano + 2.4066 * vel_media_viento - 0.0094 * direc_viento + 0.0583 * vel_racha_max - 0.0053 * presion_ats_media + 2.12

LM num: 3

vel_media_viento_24h = -0.0011 * ano + 0.1786 * vel_media_viento - 0.828 * direc_viento + 3.5855 * vel_racha_max - 0.0053 * presion_ats_media + 1.949

LM num: 4

vel_media_viento_24h = -0.0011 * ano + 4.2136 * vel_media_viento - 0.0155 * direc_viento + 0.0683 * vel_racha_max + 2.2645 * presion_ats_media + 0.0564

Number of Rules: 4
Time taken to build model: 1.35 seconds

=== Cross-validation ===

Correlation coefficient	0.5086
Mean absolute error	0.8021
Root mean squared error	1.0503
Relative absolute error	82.8272 %
Root relative squared error	86.0758 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers --** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	14.03 s	0.4347	0.8684	1.1303
500	0.2	10	0.01	13.63 s	0.4537	0.851	1.1081
500	0.2	15	0.01	19.48 s	0.425	0.8508	1.108
500	0.01	15	0.01	18.1 s	0.4946	0.8136	1.0615
2000	0.01	15	0.01	65.56 s	0.5155	0.7991	1.046

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3.000

Los valores obtenidos son:

Time taken to build model: 268.17 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5161
Mean absolute error	0.7988
Root mean squared error	1.0452
Relative absolute error	82.4859 %
Root relative squared error	85.6542 %
Total Number of Instances	3255
Ignored Class Unknown Instances	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la precipitación acumulada a 24h:

Con todos los atributos de entrada:

- **NBTree:**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Time taken to build model: 4.33 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2293	70.0367 %
Incorrectly Classified Instances	981	29.9633 %
Kappa statistic	0.381	
Mean absolute error	0.2554	
Root mean squared error	0.378	
Relative absolute error	75.6983 %	
Root relative squared error	92.0545 %	
Total Number of Instances	3274	
Ignored Class Unknown Instances	14	

=== Confusion Matrix ===

a	b	c	<-- classified as
1579	368	1	a = 0
504	714	3	b = 1
32	73	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **DecisionTable (-X 1 -S "weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1"):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
Time taken to build model: 1.22 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2316               70.7392 %
Incorrectly Classified Instances       958               29.2608 %
Kappa statistic                          0.3882
Mean absolute error                      0.2677
Root mean squared error                  0.366
Relative absolute error                 79.3419 %
Root relative squared error             89.1298 %
Total Number of Instances             3274
Ignored Class Unknown Instances         14

=== Confusion Matrix ===
   a    b    c   <-- classified as
1628  320    0 |    a = 0
 533  688    0 |    b = 1
  33   72    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Logistic (-R 1.0E-8 -M -1):**

  Los valores obtenidos son:

Time taken to build model: 0.46 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2345	71.6249 %
Incorrectly Classified Instances	929	28.3751 %
Kappa statistic	0.4008	
Mean absolute error	0.2647	
Root mean squared error	0.3646	
Relative absolute error	78.4562 %	
Root relative squared error	88.7945 %	
Total Number of Instances	3274	
Ignored Class Unknown Instances	14	

=== Confusion Matrix ===

a	b	c	<-- classified as
1674	274	0	a = 0
550	670	1	b = 1
43	61	1	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **BayesNet (-D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2240	68.4178 %
Incorrectly Classified Instances	1034	31.5822 %
Kappa statistic	0.3677	
Mean absolute error	0.2331	
Root mean squared error	0.4147	
Relative absolute error	69.0858 %	
Root relative squared error	100.98 %	
Total Number of Instances	3274	
Ignored Class Unknown Instances	14	

=== Confusion Matrix ===

a	b	c	<-- classified as
1433	515	0	a = 0
414	807	0	b = 1
34	71	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **RBFNetwork (-B 1 -S 2 -R 1.0E-8 -M -1 -W 0.1):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

```
Time taken to build model: 0.15 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2244               68.2482 %
Incorrectly Classified Instances      1044               31.7518 %
Kappa statistic                          0.3345
Mean absolute error                      0.2877
Root mean squared error                  0.3797
Relative absolute error                 85.4171 %
Root relative squared error             92.5559 %
Total Number of Instances             3288

=== Confusion Matrix ===

   a    b    c   <-- classified as
 1596  366    0 |    a = 0
  573  648    0 |    b = 1
   42   63    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error
500	0.2	10	0.2	30.5 s	2277 (69.548 %)	997 (30.452 %)	0.2552
500	0.2	10	0.01	29.35 s	2278 (69.578%)	99 (30.42%)	0.2566
500	0.2	15	0.01	42.18 s	2212(67.562 %)	1062(32.437%)	0.256
500	0.01	15	0.01	40.8 s	2349(71.747%)	925(28.2529 %)	0.2645
1000	0.01	15	0.01	90.23 s	2328(71.105 %)	946(28.8943 %)	0.2614

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 775

Los valores obtenidos son:

```
Time taken to build model: 104.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2347               71.686 %
Incorrectly Classified Instances      927                28.314 %
Kappa statistic                       0.4106
Mean absolute error                   0.262
Root mean squared error               0.3651
Relative absolute error               77.6726 %
Root relative squared error           88.9131 %
Total Number of Instances             3274
Ignored Class Unknown Instances       14

=== Confusion Matrix ===
   a    b    c   <-- classified as
1626  322    0 |    a = 0
 500  721    0 |    b = 1
  37   68    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos seleccionados de entrada:

- **NBTree:**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
Time taken to build model: 0.61 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2357               71.6849 %
Incorrectly Classified Instances       931               28.3151 %
Kappa statistic                          0.4112
Mean absolute error                      0.2453
Root mean squared error                  0.3691
Relative absolute error                 72.8505 %
Root relative squared error             89.9569 %
Total Number of Instances             3288

=== Confusion Matrix ===

   a    b    c   <-- classified as
1631  331    0 |    a = 0
 495  726    0 |    b = 1
  30   75    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **DecisionTable(-X 1 -S "weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1"):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
Time taken to build model: 0.18 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2315               70.4075 %
Incorrectly Classified Instances      973                29.5925 %
Kappa statistic                       0.3786
Mean absolute error                   0.2698
Root mean squared error               0.3671
Relative absolute error               80.102 %
Root relative squared error           89.4815 %
Total Number of Instances             3288

=== Confusion Matrix ===
   a    b    c   <-- classified as
1645  316    1 |    a = 0
 550  670    1 |    b = 1
  32   73    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Logistic (-R 1.0E-8 -M -1):**

  Los valores obtenidos son:

Time taken to build model: 0.22 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	2 287	69.556 %
Incorrectly Classified Instances	1001	30.444 %
Kappa statistic	0.3363	
Mean absolute error	0.2807	
Root mean squared error	0.3736	
Relative absolute error	83.344 %	
Root relative squared error	91.0601 %	
Total Number of Instances	3288	

=== Confusion Matrix ===

```
   a    b    c   <-- classified as
1750  212    0 |    a = 0
 683  537    1 |    b = 1
  48   57    0 |    c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **BayesNet ((-D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2357	71.6849 %
Incorrectly Classified Instances	931	28.3151 %
Kappa statistic	0.4112	
Mean absolute error	0.2453	
Root mean squared error	0.3691	
Relative absolute error	72.8237 %	
Root relative squared error	89.9665 %	
Total Number of Instances	3288	

=== Confusion Matrix ===

a	b	c	<-- classified as
1631	331	0	a = 0
495	726	0	b = 1
30	75	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **RBFNetwork (-B 1 -S 2 -R 1.0E-8 -M -1 -W 0.1):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

Time taken to build model: 0.13 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2277	69.2518 %
Incorrectly Classified Instances	1011	30.7482 %
Kappa statistic	0.3229	
Mean absolute error	0.2843	
Root mean squared error	0.3772	
Relative absolute error	84.412 %	
Root relative squared error	91.9311 %	
Total Number of Instances	3288	

=== Confusion Matrix ===

a	b	c	<-- classified as
1774	188	0	a = 0
718	503	0	b = 1
54	51	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers --** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error
500	0.2	10	0.2	10.9 s	2297 (70.1588 %)	977 (29.8412 %)	0.2729
500	0.2	10	0.01	11.17 s	2300 (70.2505 %)	974 (29.7495 %)	0.2736
500	0.2	15	0.01	16.21 s	2300 (70.2505 %)	974 29.7495 %	0.2733
500	0.01	15	0.01	15.19 s	2288 (69.8839 %)	986 (30.1161 %)	0.2774
1000	0.01	15	0.01	30.59 s	2302 (70.3115 %)	972 (29.6885 %)	0.2746

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15(2 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 1300

Los valores obtenidos son:

Time taken to build model: 39.91 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2315	70.7086 %
Incorrectly Classified Instances	959	29.2914 %
Kappa statistic	0.3758	
Mean absolute error	0.2741	
Root mean squared error	0.3721	
Relative absolute error	81.2456 %	
Root relative squared error	90.6263 %	
Total Number of Instances	3274	
Ignored Class Unknown Instances	14	

=== Confusion Matrix ===

a	b	c	<-- classified as
1691	257	0	a = 0
597	624	0	b = 1
39	66	0	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Evaluación e interpretación

A la hora de evaluar los modelos que he construido en la fase de minería de datos me voy a centrar en analizar los resultados de la evaluación de estos modelos con la validación cruzada con 10 pliegues. Este método de evaluación es muy preciso ya que entrena con todos los pliegues menos uno y luego evalúa el modelo creado con el pliegue que no se ha utilizado para entrenar. Este proceso se repite tantas veces como pliegues halla, escogiendo siempre un conjunto de pliegues distintos para entrenar y otro para evaluar.

Predicción de la temperatura máxima a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	3.84 s	0.9185	2.6133	3.312	37.104%	39.533%	3268	20
LinearRegression	0.04 s	0.9159	2.6443	3.3628	37.543%	40.139%	3268	20
Isotonic regression	4.79 s	0.914	2.65	3.38	37.901 %	40.577%	3268	20
LeastMedSq	1.97 s	0.916	2.6413	3.3601	37.501%	40.107%	3268	20
MultilayerPerceptron	760.8s	0.9257	2.5019	3.1714	35.522%	37.855%	3268	20
Bagging (M5P -M 1.0) 10 iteraciones	22.97 s	0.9195	2.5986	3.293	36.895%	39.307%	3268	20

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.6 s	0.9203	2.5893	3.278	36.7681 %	39.132 %	3268	20
LinearRegression	0.03 s	0.9165	2.636	3.3512	37.4316 %	40.006%	3268	20
Isotonic regression	2.91 s	0.9	2.9269	3.6514	41.5631 %	43.590%	3268	20
LeastMedSq	1.49 s	0.9165	2.6356	3.3505	37.4267 %	39.997%	3268	20
MultilayerPerceptron	548.69 s	0.9274	2.4778	3.1349	35.1808 %	37.419%	3268	20
Bagging (M5P -M 1.0) 10 iteraciones	20.8 s	0.9232	2.5419	3.2195	36.0954 %	38.433%	3268	20

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura máxima al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 2.47 grados y un error relativo de un 35.18 %. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 (bagging) y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye en la mayoría (todos) de los modelos un poco (M5P, LinearRegression, Bagging, MultilayerPerceptron) y que los resultados mejoran (M5P, LinearRegression, MultilayerPerceptron y Bagging) con unos modelos y empeoran (Isotonic regression,) con otros pero no varían mucho frente a los obtenidos con todos los atributos.

Predicción de la temperatura media a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	3.87 s	0.9509	1.6108	2.0642	28.5692%	30.9586%	3268	20
LinearRegression	0.1 s	0.9494	1.6248	2.0941	28.8181%	31.4077%	3268	20
Isotonic regression	3.96 s	0.9424	1.7557	2.2305	31.1404%	33.4532%	3268	20
LeastMedSq	1.69 s	0.9493	1.6249	2.0946	28.8204%	31.4153%	3268	20
MultilayerPerceptron	621.33s	0.9554	1.5341	1.9704	27.2102%	29.552 %	3268	20
Bagging (M5P -M 1.0) 10 iteraciones	22.07 s	0.9514	1.6042	2.0531	28.4524%	30.7921%	3268	20

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relaive squared error	Total Number of Instances	Unknown Instances
M5P -M 1.0	3.45 s	0.9507	1.6126	2.0676	28.6018%	31.0106%	3268	20
LinearRegression	0.04 s	0.9494	1.6242	2.0942	28.8082%	31.4093%	3268	20
Isotonic regression	3.96 s	0.9424	1.7557	2.2305	31.1404%	33.4532%	3268	20
LeastMedSq	1.42s	0.9494	1.623	2.0946	28.786%	31.3906%	3268	20
MultilayerPerceptron	537.s	0.9549	1.5448	1.9786	27.399%	29.67%	3268	20
Bagging (M5P -M 1.0) 10 iteraciones	17.33 s	0.9513	1.6042	2.0553	28.4524%	30.825 %	3268	20

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura media al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 1.5341 grados y un error relativo de un 27.2%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 (bagging) y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye con la mayoría de los modelos un poco (M5P, LinearRegression, Bagging) y que los resultados mejoran (LinearRegression) con unos modelos y empeoran con otros pero prácticamente no varían frente a los obtenidos con todos los atributos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Predicción de la temperatura mínima a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	3.7 s	0.9405	1.5412	1.9618	31.882%	33.955%	3268	20
LinearRegression	0.1 s	0.9391	1.5635	1.9848	32.345%	34.353%	3268	20
Isotonic regression	4.95 s	0.9284	1.714	2.1457	35.458 %	37.138%	3268	20
LeastMedSq	**3.93 s**	0.9392	1.5628	1.9834	32.330%	34.329%	3268	20
MultilayerPerceptron	561.95 s	0.9458	1.4702	1.8753	30.413%	32.458%	3268	20
Bagging (M5P -M 1.0) 10 iteraciones	36.32 s	0.9406	1.5435	1.9611	31.931 %	33.944 %	3268	20

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	3.36 s	0.9407	1.542	1.9596	31.8997%	33.9171%	3268	20
LinearRegression	0.02 s	0.9391	1.5634	1.9843	32.3426%	34.3447%	3268	20
Isotonic regression	1.98 s	0.9284	1.714	2.1457	35.458 %	37.1389%	3268	20
LeastMedSq	1.71 s	0.9392	1.5631	1.9827	32.3357%	34.3175%	3268	20
MultilayerPerceptron	470.14 s	0.9426	1.5219	1.9291	31.4836%	33.3899%	3268	20
Bagging (M5P -M 1.0) 10 iteraciones	16.42 s	0.9408	1.5407	1.9569	31.8727%	33.8713%	3268	20

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura mínima al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 1.4702 grados y un error relativo de un 30.413%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye con la mayoría de los modelos un poco (M5P, LinearRegression, Bagging) y que los resultados mejoran (LinearRegression, MultilayerPerceptron) con unos modelos y empeoran (Isotonic regression,) con otros pero prácticamente no varían frente a los obtenidos con todos los atributos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa de los mejores resultados para la predicción de las temperaturas:

Temperatura	Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error
Temperatura máxima	**MultilayerPerceptron**	548.69 s	0.9274	2.4778	3.1349	35.1808 %	37.419%
Temperatura media	**MultilayerPerceptron**	621.33s	0.9554	1.5341	1.9704	27.2102%	29.552 %
Temperatura mínima	**MultilayerPerceptron**	561.95 s	0.9458	1.4702	1.8753	30.413%	32.458%

Como se observa las mejores predicciones las he obtenido mediante el método perceptrón multicapa. A demás se aprecia claramente que la predicción de la temperatura media es la más precisa (error relativo de un 27%) seguida de la predicción de la temperatura mínima (30%) y ya más atrás la temperatura máxima (35%). El hecho de que la temperatura media es la mejor predecible es en gran parte al hecho de que la temperatura media es menos variable que las temperaturas máximas y mínimas.

Predicción de la velocidad media del viento a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	3.6 s	0.5346	0.7833	1.0308	80.9258 %	84.5119 %	3255	33
LinearRegression	0.03 s	0.5095	0.804	1.0496	83.0884 %	86.0161 %	3255	33
Isotonic regression	1.38 s	0.4916	0.814	1.064	83.893 %	87.0653 %	3255	33
LeastMedSq	2.78 s	0.4906	0.8176	1.0996	84.4323 %	90.1139 %	3255	33
MultilayerPerceptron	302.71 s	0.5215	0.7918	1.0422	81.7666 %	85.4098 %	3255	33
Bagging (M5P -M 1.0) 10 iteraciones	2.25 s	0.5132	0.7987	1.0473	82.4753 %	85.8314 %	3255	33

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	1.65 s	0.5086	0.8021	1.0503	82.8272 %	86.0758 %	3255	33
LinearRegression	0.02 s	0.5087	0.801	1.0501	83.1332 %	86.051 %	3255	33
Isotonic regression	0.48 s	0.4916	0.8124	1.0624	83.893 %	87.0653 %	3255	33
LeastMedSq	2.14 s	0.506	0.8184	1.0948	84.50%	89.71%	3255	33
MultilayerPerceptron	268.1s	0.5161	0.7988	1.0452	82.48%	85.65%	3255	33
Bagging (M5P -M 1.0) 10 iteraciones	1.35 s	0.5086	0.8021	1.0503	82.8272 %	86.0758 %	3255	33

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la velocidad media del viento al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 0.7918 m/s y un error relativo de un 81.76%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye bastante y que los resultados mejoran (LinearRegression,) con unos modelos y empeoran (Isotonic regression, MP5, MultilayerPerceptron) con otros pero prácticamente no varían frente a los obtenidos con todos los atributos.

Predicción de la precipitación acumulada a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
NBTree	159.27 s	2302 (70.0122%)	1000 (30.4136%)	0.2638	0.3789	77.586%	91.9107 %	3288	0
DecisionTable	1.19 s	2333 (70.955 %)	955 (29.045 %)	0.2731	0.3693	80.322%	89.5787 %	3288	0
Logistic	0.41 s	2344 (71.289%)	944 (28.7105%)	0.268	0.367	78.823%	89.0268 %	3288	0
BayesNet	0.05 s	2237 (68.035%)	1051 (31.9%)	0.2349	0.4159	69.088%	100.8908 %	3288	0
RBFNetwork	0.15 s	2244 (68.2482 %)	1044 (31.7518%)	0.2877	0.3797	85.417%	92.5559 %	3288	0
MultilayerPerceptron	104.0s	2347 (71.686 %)	927 (28.314 %)	0.262	0.3651	77.672%	88.9131 %	3274	14

Con los atributos seleccionados:

Método	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
NBTree	0.52 s	2343 (71.2591 %)	945 (28.7409%)	0.2482	0.371	72.998%	90.0015 %	3288	0
DecisionTable	0.17 s	2333 (70.955 %)	955 (29.045 %)	0.2711	0.3684	79.755%	89.3684 %	3288	0
Logistic	0.19 s	2275 (69.191 %)	1013 (30.809 %)	0.2839	0.3758	83.50%	91.1741 %	3288	0
BayesNet	0.06 s	2343 (71.2591 %)	945 (28.7409%)	0.2481	0.371	72.969%	90.0099 %	3288	0
RBFNetwork	0.13 s	2277 (69.2518 %)	1011 (30.7482%)	0.2843	0.3772	84.412 %	91.9311 %	3288	0
MultilayerPerceptron	155.91 s	2319 (70.8308 %)	955 (29.1692%)	0.2717	0.3689	80.541%	89.8353 %	3274	14

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura mínima al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se clasifican correctamente un 71.686 de las instancias. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante una regresión Logística con un 71.28% de las instancias clasificadas correctamente.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En los resultados de todos las clasificaciones se observa que la tasa de acierto de las instancias en la clase 1 (no llueve) es de 83.4%, la tasa de acierto de las instancias de la clase 2 (llueve débilmente) es de un 60% y la tasa de acierto de las instancias de la clase 3 (llueve fuerte) es de un 0%.

Dado que la mayoría (el 59.4%) de las instancias pertenecían a la clase 1(no hay precipitación acumulada) y luego a la clase 2 (hay poca precipitación acumulada) al final la tasa de acierto general es de un 71.86%. De estos resultados se deduce que clasifica bien las instancias de la clase 1, clasifica regular las de la clase 2 y clasifica fatal las de la clase 3.

Otro aspecto a destacar es que al realizar la clasificación sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye bastante en casi todos los métodos (Excepto bayesNet) y que los resultados empeoran frente a los obtenidos con todos los atributos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Estación de Gorramend

A continuación voy a mostrar el trabajo realizado en cada una de las fases del proceso del descubrimiento del conocimiento KDD, con el objetivo de poder predecir las temperaturas, velocidad media del viento y precipitación acumulada a 24h.

Integración y recopilación de los datos:

La integración de los datos se ha realizado de la misma forma que para la estación de Montes de Cierzo pero con los datos de la estación meteorológica de Gorramendi

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección, Limpieza y transformación de los datos:

En esta sub fase se tiene en cuenta diferentes aspectos como:

Limpieza de datos:

Tratamiento de valores perdidos (missing values)

- Aplicando el filtro ReplaceMissingValues el cual sustituye todos los valores no encontrados (missing) de un atributo por la moda y la media de ese atributo.

Tratamiento de valores anómalos (outlier values):

- Mediante el filtro IntercuatileRange detectamos los valores outlier (anómalos) así como los valores extremos. Pero debido a que al eliminar estos valores los métodos obtienen peores resultados hemos decidido no aplicar este filtro.

Trasformación de datos:

Discretizacion y numeración:

- Como he comentado anteriormente he discretizado el atributo numérico precipitación acumulada a 24 h en tres valores nominales (1, 2,3).

Normalización de rango (escalado):

- Al ser todos mis atributos numéricos y cada uno de ellos con un rango de valores muy diferentes he decidido normalizarlos para comprender mejor los atributos. Para ello he utilizado el filtro llamado Normalize el cual ha normalizado los valores entre 0 y 1.

Reducción de la dimensión:

- He eliminado dos atributos llamados **velocidad media viento km/h** y dirección **de la racha máxima km/h** ya que tenía otros dos atributos iguales pero medidos en m/s. Con los atributos y muestras que dispongo no es necesaria en mi caso ya que Weka puede tratarlos bien.

Aumento de la dimensión:

- Probé a introducir nuevos atributos derivados de los ya existentes como la diferencia entre la temperatura máxima y la mínima pero sin embargo los resultados no mejoraban por lo que no introduje nuevos atributos derivados. A demás al introducir nuevos atributos aumentaba el tiempo de cómputo y los resultados no mejoraban.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de datos:

Selección de las características (atributos) más importantes:

Weka incorpora una gran cantidad de métodos para estudiar la relevancia de atributos y realizar una selección automática de los mismos. Para utilizar estos métodos una vez cargados los datos es necesario elegir uno de estos métodos selección en la pestaña del explorador llamada select atributes. Hay dos tipos de evaluadores principales: los SubsetEval (evaluadores de subconjuntos o selectores) y AtributteEval (Prorreteadores de atributos). Los primeros tienen que elegir una estrategia o método de búsqueda de subconjuntos (Search Method) para encontrar el mejor subconjunto de atributos que nos proporcione los mejores resultados para un modelo de clasificación. Como resultado tendremos aquellos atributos con los cuales se ha realizado una mejor clasificación. El segundo tipo de evaluadores solo pueden combinarse con un Ranker ya que no seleccionan los atributos sino que los ordenan por su relevancia a la hora de clasificar el atributo objetivo.

A continuación muestro los resultados obtenidos del proceso de selección de atributos para los datos de la estación de Gorramendi:

En primer lugar voy a enumerar todos los atributos que dispongo para realizar las predicciones/clasificaciones:

Atributo	Numero	Atributo	Numero
día	**1**	**Humedad mínima**	**9**
mes	**2**	**Precipitación acumulada**	**10**
año	**3**	**Velocidad media del viento**	**11**
Temperatura media	**4**	**Dirección del viento**	**12**
Temperatura máxima	**5**	**Velocidad de la racha máxima**	**13**
Temperatura mínima	**6**	**Dirección de la racha máxima**	**14**
Humedad media	**7**	**Radiación global**	**15**
Humedad máxima	**8**	**insolación**	**16**

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura máxima:

Aplicando evaluadores de subconjuntos:

Atributo	**Evaluador de atributos**	**Modelo**	**Estrategia de búsqueda**	**Atributos seleccionados**
Temperatura máxima a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	4,6,7,8,9,13,14,15
Temperatura máxima a 24h	WrapperSubsetEval	M5Rules	BestFirst	4,6,7,8,9,13,14,15
Temperatura máxima a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,3,4,6,7,8,9,11,13, 14,15
Temperatura máxima a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	4,6,7,8,9,13,14,15

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son bastante parecidos salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 4 (temperatura media), 6 (temperatura mínima), 7 (Humedad media), 8 (humedad máxima), 9 (humedad mínima), 13 (velocidad racha máxima), 14 (dirección racha máxima) ,15 (Radiación global)

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo RelieF que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

0.008677   2 mes
 0.008135   6 tem_min
 0.006631   4 tem_med
 0.005994   5 tem_max
 0.005727   3 ano
 0.003583   9 humedad_min
 0.003425  15 radiacion_global
 0.003267  12 direc_viento
 0.002282   7 humedad_med
 0.00207   14 direc_racha_max
 0.00136    8 humedad_max
 0.001351  16 insolacion
 0.000779   1 dia
 0.000367  10 precip_acumulada
-0.001647  13 vel_racha_max
-0.002739  11 vel_media_viento

Selected attributes: 2,6,4,5,3,9,15,12,7,14,8,16,1,10,13,11 : 16
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura máxima). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el mes, las temperaturas, el año, humedad mínima y radiación.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura mínima:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Temperatura mínima a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	1,2,4,5,6,7,8,9,12,13,14,15,16
Temperatura mínima a 24h	WrapperSubsetEval	M5Rules	BestFirst	2,4,5,6,7,8,13,14,15
Temperatura mínima a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,4,5,6,7,9,10,13,14,15
Temperatura mínima a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,4,5,6,7,8,9,13,14,15

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son prácticamente bastante parecidos salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos 2,4, 8, 13, 15,16.

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo RelieF que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

0.0128161   6 tem_min
 0.0104094   4 tem_med
 0.0085306   2 mes
 0.0079421   5 tem_max
 0.0065825   3 ano
 0.0058417   8 humedad_max
 0.0043938   7 humedad_med
 0.003369    9 humedad_min
 0.0029149  12 direc_viento
 0.0015117  15 radiacion_global
 0.0013152   1 día
 0.0001561  16 insolación
 0.0000644  13 vel_racha_max
-0.0000105  11 vel_media_viento
-0.0005514  10 precip_acumulada
-0.0006362  14 direc_racha_max

Selected attributes: 6,4,2,5,3,8,7,9,12,15,1,16,13,11,10,14 : 16
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura mínima). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el las temperaturas mínima y media, el mes y temperatura máxima.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Temperatura media:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Temperatura media a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	1,2,4,5,6,7,8,11,12, 13,14,15,16
Temperatura media a 24h	WrapperSubsetEval	M5Rules	BestFirst	1,2,3,4,5,6,7,8,9,11, 13,14,15
Temperatura media a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	2,4,5,6,7,8,9,11,13, 14,15
Temperatura media a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	2,3,4,5,6,7,8,9,13,1 4,15

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son bastante parecidos salvo por pequeñas diferencias. Además todos estos métodos de selección comparten como atributos seleccionados los atributos (2, 12, 14, 15, 16, 17, 20).

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo RelieF que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:

 0.010786   6 tem_min
 0.00862    2 mes
 0.008566   4 tem_med
 0.006933   5 tem_max
 0.005736   3 ano
 0.003926   9 humedad_min
 0.003873   8 humedad_max
 0.003703   7 humedad_med
 0.003439  12 direc_viento
 0.002359  15 radiacion_global
 0.000912  14 direc_racha_max
 0.000689  16 insolacion
 0.000525   1 dia
-0.000182  10 precip_acumulada
-0.000942  13 vel_racha_max
-0.001515  11 vel_media_viento

Selected attributes: 6,2,4,5,3,9,8,7,12,15,14,16,1,10,13,11 : 16
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (temperatura media). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el la temperatura mínima, el mes, temperatura media, temperatura máxima y el año.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Velocidad media del viento a 24h:

Aplicando evaluadores de subconjuntos:

Atributo	Evaluador de atributos	Modelo	Estrategia de búsqueda	Atributos seleccionados
Velocidad media del viento a 24h	WrapperSubsetEval	M5Rules	GeneticSearch	6,7,10,11,12,13,15, 16
Velocidad media del viento a 24h	WrapperSubsetEval	M5Rules	BestFirst	4,8,10,11,12,13,15, 16
Velocidad media del viento a 24h	WrapperSubsetEval	LinearRegresion	GeneticSearch	4,9,10,11,12,13,15, 16
Velocidad media del viento a 24h	WrapperSubsetEval	LinearRegresion	BestFirst	4,7,10,11,12,13,15, 16

ras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos son parecidos. Además todos estos métodos de selección comparten como atributos seleccionados los atributos (4, 5, 10, 12,15).

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

Ranked attributes:

0.011294 13 vel_racha_max
0.011253 11 vel_media_viento
0.009048 3 ano
0.006598 1 dia
0.005134 12 direc_viento
0.004833 10 precip_acumulada
0.003529 2 mes
0.003458 16 insolacion
0.00298 6 tem_min
0.002333 4 tem_med
0.001869 5 tem_max
0.00185 8 humedad_max
0.001649 9 humedad_min
0.001424 7 humedad_med
0.001416 15 radiacion_global
0.000865 14 direc_racha_max

Selected attributes: 13,11,3,1,12,10,2,16,6,4,5,8,9,7,15,14 : 16

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (velocidad media del viento). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el la velocidad de la racha máxima y velocidad media del viento.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos: Predicción de la Precipitación acumulada a 24h:

Aplicando evaluadores de subconjuntos:

Atributo	**Evaluador de atributos**	**Modelo**	**Estrategia de búsqueda**	**Atributos seleccionados**
Precipitación acumulada a 24h	WrapperSubsetEval	J48	GeneticSearch	6,10,11
Precipitación acumulada a 24h	WrapperSubsetEval	J48	BestFirst	6,10,11
Precipitación acumulada a 24h	WrapperSubsetEval	Logistic	GeneticSearch	2,4,5,6,7,8,10,11,12 ,14,15,16
Precipitación acumulada a 24h	WrapperSubsetEval	Logistic	BestFirst	2,4,5,6,7,8,9,10,11, 12,13,14,16

Tras aplicar los selectores de atributos podemos observar que los atributos seleccionados por los distintos métodos no coinciden ya que solo tienen un atributo en común el numero 9. Se observa que con la selección de atributos se disminuye drásticamente el número de atributos seleccionados.

Aplicando prorreteadores de Atributos:

A continuación se va a mostrar una lista ordenada con la importancia de cada atributo a la hora de clasificar. Para ello se va utilizar el algoritmo **RelieF** que determina la importancia de cada atributo a la hora de clasificar el atributo objetivo.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

RelieFAttributeEval:

```
Ranked attributes:
 0.011294  13 vel_racha_max
 0.011253  11 vel_media_viento
 0.009048   3 ano
 0.006598   1 dia
 0.005134  12 direc_viento
 0.004833  10 precip_acumulada
 0.003529   2 mes
 0.003458  16 insolacion
 0.00298    6 tem_min
 0.002333   4 tem_med
 0.001869   5 tem_max
 0.00185    8 humedad_max
 0.001649   9 humedad_min
 0.001424   7 humedad_med
 0.001416  15 radiacion_global
 0.000865  14 direc_racha_max

Selected attributes: 13,11,3,1,12,10,2,16,6,4,5,8,9,7,15,14 : 16
```

Aplicando este método obtenemos el conjunto de todos los atributos ordenados en función de su importancia (Relevancia) para predecir o clasificar el atributo objetivo (precipitación acumulada). Como se observa en los resultados de la tabla los atributos más influyentes en la clasificación son el la velocidad de la racha máxima y velocidad media del viento.

Tras la aplicación de diferentes métodos para seleccionar estos atributos he decidido quedar con los atributos seleccionados mediante el evaluador de atributos WrapperSubsetEval (evaluador de subconjuntos envolvente) basado en el clasificador M5Rules y con un método de búsqueda genético (GeneticSearch). Esta decisión es debido a los algoritmos WrapperSubsetEval son capaces de escoger mejor aquellos los atributos claves para la clasificación (el punto negativo es que tardan mucho tiempo).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En esta fase voy a mostrar los resultados obtenidos de aplicar los diferentes métodos de minería de datos a la tarea de predicción de los atributos. Como método de evaluación he utilizado una validación cruzada con 10 pliegues. En este método los datos se dividen aleatoriamente en n (pliegues) grupos. Un grupo se reserva para el conjunto de prueba y los n-1 restantes (juntando sus datos) se construye un modelo y se usa para predecir el resultado de los datos del grupo reservado. Este proceso se repite n veces dejando cada vez un grupo diferente para la prueba. Esto quiere decir que se calculan n ratios de error independientes. Finalmente se construye un modelo con todos los datos y se obtienen sus ratios de error y precisión promediando los n ratios de error disponibles.

Además en esta fase se aplicara las técnicas de minería de datos a todos los atributos de entrada y a los atributos seleccionados en la fase anterior y así realizar una comparativa de los resultados obtenidos.

A la hora de realizar la predicción de las temperaturas, la velocidad y la precipitación acumulada hemos utilizado estos métodos ya que son los que mejores resultados nos han ofrecido:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la temperatura máxima a 24h:

Con todos los atributos de entrada:

Resultados obtenidos con los diferentes métodos:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_max_24h = 0.2394 * dia + 0.1827 * mes + -0.3039 * ano + 14.8659 * tem_med + -2.4683 * tem_max + 20.9278 * tem_min +8.8304 * humedad_med +-5.9518 * humedad_max +-3.2253 * humedad_min + -1.4548 * precip_acumulada +-0.1892 * vel_media_viento +-0.0171 * direc_viento +-0.9672 * vel_racha_max +-0.555 * direc_racha_max +6.8975 * radiacion_global +-0.958 * insolacion + -6.2946

Time taken to build model: 2.54 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8845
Mean absolute error	2.564
Root mean squared error	3.242
Relative absolute error	45.6549 %
Root relative squared error	46.6781 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_max_24h =-0.3681 * ano +16.5874 * tem_med +-2.1375 * tem_max +19.792 * tem_min + 8.7199 * humedad_med + -5.7249 * humedad_max + -3.0538 * humedad_min + -1.4611 * precip_acumulada + -1.2266 * vel_racha_max + -0.5976 * direc_racha_max + 6.5601 * radiacion_global + -0.7086 * insolacion +-6.6142

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8844
Mean absolute error	2.5633
Root mean squared error	3.2412
Relative absolute error	45.642 %
Root relative squared error	46.6666 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 3.86 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.859
Mean absolute error	2.7922
Root mean squared error	3.5558
Relative absolute error	49.7192 %
Root relative squared error	51.1976 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
(using smoothed linear models)
LM1 (3198/46.334%)
LM num: 1

tem_max_24h = -0.3681 * ano + 16.5874 * tem_med - 2.1375 * tem_max + 19.792 * tem_min + 8.7199 * humedad_med - 5.7249 * humedad_max - 3.0538 * humedad_min - 1.4611 * precip_acumulada - 1.2266 * vel_racha_max - 0.5976 * direc_racha_max + 6.5601 * radiacion_global - 0.7086 * insolacion - 6.6142

Number of Rules : 1
Time taken to build model: 2.69 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8841
Mean absolute error	2.5658
Root mean squared error	3.2448
Relative absolute error	45.6875 %
Root relative squared error	46.7194 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
(using smoothed linear models)
LM1 (3198/45.362%)
LM num: 1

tem_max_24h = -0.4016 * ano + 12.4073 * tem_med + 22.1635 * tem_min + 7.7782 * humedad_med - 4.7076 * humedad_max - 2.6337 * humedad_min - 2.3633 * precip_acumulada - 0.7332 * direc_racha_max + 6.7147 * radiacion_global - 0.5277 * insolacion - 7.5206

Number of Rules: 1
Time taken to build model: 22.41 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8849
Mean absolute error	2.5576
Root mean squared error	3.2348
Relative absolute error	45.5409 %
Root relative squared error	46.5757 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers --** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	19.49 s	0.8737	2.667	3.3939
500	0.2	10	0.01	19.18 s	0.8772	2.6316	3.3581
500	0.2	15	0.01	28.31 s	0.8765	2.6347	3.361
500	0.01	15	0.01	31.28 s	0.8857	2.5458	3.2259
3000	0.01	15	0.01	147.69 s	0.8885	2.5114	3.1883

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15(1 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3.250

Time taken to build model: 178.92 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8885
Mean absolute error	2.5112
Root mean squared error	3.1893
Relative absolute error	44.7152 %
Root relative squared error	45.9201 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_max_24h =11.3232 * tem_med + 22.5582 * tem_min +8.8579 * humedad_med + -5.776 * humedad_max + -3.0653 * humedad_min +-1.2783 * vel_racha_max + -0.5777 * direc_racha_max +5.8812 * radiacion_global + -6.7134

Time taken to build model: 2.14 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8848
Mean absolute error	2.5604
Root mean squared error	3.2387
Relative absolute error	45.59 %
Root relative squared error	46.6309 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_max_24h = 13.197 * tem_med + 21.3331 * tem_min + 8.5811 * humedad_med + -5.505 * humedad_max + -2.9017 * humedad_min +- 1.3494 * vel_racha_max + -0.593 * direc_racha_max + 5.8859 * radiacion_global + -7.0978

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8849
Mean absolute error	2.5569
Root mean squared error	3.2343
Relative absolute error	45.5287 %
Root relative squared error	46.5679 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 1.83 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.859
Mean absolute error	2.7922
Root mean squared error	3.5558
Relative absolute error	49.7192 %
Root relative squared error	51.1976 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:
(using smoothed linear models)
LM1 (3198/46.401%)
LM num: 1

tem_max_24h = 13.197 * tem_med + 21.3331 * tem_min + 8.5811 * humedad_med - 5.505 * humedad_max - 2.9017 * humedad_min - 1.3494 * vel_racha_max - 0.593 * direc_racha_max + 5.8859 * radiacion_global - 7.0978

Number of Rules: 1
Time taken to build model: 2.31 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8849
Mean absolute error	2.5569
Root mean squared error	3.2343
Relative absolute error	45.5287 %
Root relative squared error	46.5679 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
(using smoothed linear models)
LM1 (3198/45.431%)

LM num: 1

tem_max_24h = 12.1651 * tem_med + 22.5085 * tem_min + 7.4817 * humedad_med - 4.4581 * humedad_max - 2.5443 * humedad_min - 0.722 * vel_racha_max - 0.6931 * direc_racha_max + 6.1765 * radiacion_global - 7.7277

Number of Rules : 1
Time taken to build model: 17.71 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8858
Mean absolute error	2.5487
Root mean squared error	3.2221
Relative absolute error	45.3827 %
Root relative squared error	46.3922 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – número de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	11.12 s	0.881	2.5943	3.2905
500	0.2	10	0.01	11.25 s	0.8819	2.5796	3.2798
500	0.2	15	0.01	16.55 s	0.8831	2.5706	3.2641
500	0.01	15	0.01	16.3 s	0.8852	2.547	3.2318
3000	0.01	15	0.01	92.28 s	0.8874	2.5285	3.2024

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15 (1 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3250

Los valores obtenidos son:

Time taken to build model: 98.44 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8874
Mean absolute error	2.5271
Root mean squared error	3.2017
Relative absolute error	44.998 %
Root relative squared error	46.0986 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.1102 * dia + 0.4297 * mes + -0.2745 * ano + 16.2963 * tem_med + -4.3267 * tem_max + 20.5381 * tem_min + 7.8131 * humedad_med + -4.3441 * humedad_max + -2.7361 * humedad_min + -0.3465 * precip_acumulada + 0.0295 * vel_media_viento + -0.0453 * direc_viento + -1.2222 * vel_racha_max + -0.5693 * direc_racha_max + 4.4333 * radiacion_global + 0.0059 * insolacion + -9.2139

Time taken to build model: 1.67 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9092
Mean absolute error	2.0243
Root mean squared error	2.5874
Relative absolute error	40.0617 %
Root relative squared error	41.6366 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.418 * mes +-0.2763 * ano +17.3642 * tem_med + -4.8791 * tem_max + 20.4058 * tem_min +7.5715 * humedad_med +-4.194 * humedad_max + -2.6473 * humedad_min +-1.208 * vel_racha_max + -0.6245 * direc_racha_max + 4.4531 * radiacion_global +-9.3615

Time taken to build model: 0.03 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9093
Mean absolute error	2.0212
Root mean squared error	2.5851
Relative absolute error	40.0002 %
Root relative squared error	41.5986 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 3.15 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8909
Mean absolute error	2.1957
Root mean squared error	2.8226
Relative absolute error	43.4526 %
Root relative squared error	45.4215 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
(using smoothed linear models)
LM1 (3198/41.367%)

LM num: 1

tem_med_24h = 0.418 * mes - 0.2763 * ano + 17.3642 * tem_med - 4.8791 * tem_max + 20.4058 * tem_min + 7.5715 * humedad_med - 4.194 * humedad_max - 2.6473 * humedad_min - 1.208 * vel_racha_max - 0.6245 * direc_racha_max + 4.4531 * radiacion_global - 9.3615

Number of Rules : 1

Time taken to build model: 1.9 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.909
Mean absolute error	2.0283
Root mean squared error	2.5895
Relative absolute error	40.1402 %
Root relative squared error	41.67 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:
(using smoothed linear models)
LM1 (3198/40.444%)

LM num: 1

tem_med_24h = 0.3983 * mes - 0.2883 * ano + 14.3797 * tem_med - 3.3874 * tem_max + 22.1876 * tem_min + 6.3082 * humedad_med - 3.2648 * humedad_max - 2.0014 * humedad_min - 0.7128 * vel_racha_max - 0.6969 * direc_racha_max + 4.6399 * radiacion_global - 9.8918

Number of Rules : 1

Time taken to build model: 18.45 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9105
Mean absolute error	2.0119
Root mean squared error	2.5687
Relative absolute error	39.8159 %
Root relative squared error	41.3357 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (**hiddenLayers**).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	17.16 s	0.9006	2.1118	2.7154
500	0.2	10	0.01	16.89 s	0.9029	2.0941	2.6924
500	0.2	15	0.01	24.98 s	0.9019	2.0939	2.7059
500	0.01	15	0.01	24.5 s	0.9096	2.013	2.5822
3000	0.01	15	0.01	162.64 s	0.9112	1.9965	2.5615

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15 (1 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 2.000

Los valores obtenidos son:

Time taken to build model: 110.54 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9116
Mean absolute error	1.9941
Root mean squared error	2.5555
Relative absolute error	39.4627 %
Root relative squared error	41.1231 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_med_24h =0.1164 * dia +0.4608 * mes +9.3124 * tem_med +0.3276 * tem_max +22.9857 * tem_min +3.6681 * humedad_med +-2.4559 * humedad_max +-0.525 * vel_media_viento +-0.0575 * direc_viento +-1.0908 * vel_racha_max +-0.5272 * direc_racha_max +4.3245 * radiacion_global +0.1737 * insolacion +-9.3871

Time taken to build model: 1.42 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9083
Mean absolute error	2.0329
Root mean squared error	2.5998
Relative absolute error	40.2304 %
Root relative squared error	41.835 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_med_24h = 0.4625 * mes + 10.2572 * tem_med + 22.7499 * tem_min +3.5599 * humedad_med + -2.3579 * humedad_max +-1.5233 * vel_racha_max +-0.5667 * direc_racha_max + 4.5004 * radiacion_global + -9.5578

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9084
Mean absolute error	2.0296
Root mean squared error	2.5975
Relative absolute error	40.1659 %
Root relative squared error	41.7991 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 2.47 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.8909
Mean absolute error	2.1957
Root mean squared error	2.8226
Relative absolute error	43.4526 %
Root relative squared error	45.4215 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:
(using smoothed linear models)

tem_min <= 0.493 : LM1 (1553/39.32%)
tem_min > 0.493 : LM2 (1645/42.702%)

LM num: 1

tem_med_24h = 0.3513 * mes + 0.0981 * tem_med + 6.4016 * tem_max + 25.8902 * tem_min + 0.0341 * humedad_med - 0.0226 * humedad_max - 1.0063 * vel_media_viento - 0.0146 * vel_racha_max - 0.0054 * direc_racha_max + 4.2209 * radiacion_global + 8.7347

LM num: 2

tem_med_24h = 1.0885 * mes + 22.7467 * tem_med - 6.9927 * tem_max + 15.1015 * tem_min + 6.2528 * humedad_med - 4.9765 * humedad_max - 1.8624 * vel_racha_max - 0.8478 * direc_racha_max + 4.8173 * radiacion_global - 8.4371

Number of Rules : 2
Time taken to build model: 1.65 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.909
Mean absolute error	2.0245
Root mean squared error	2.5903
Relative absolute error	40.0651 %
Root relative squared error	41.6833 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

(using smoothed linear models)
LM1 (3198/40.606%)

LM num: 1

tem_med_24h = 0.4489 * mes + 9.563 * tem_med + 23.6982 * tem_min + 3.2949 * humedad_med - 1.922 * humedad_max - 0.9202 * vel_racha_max - 0.6616 * direc_racha_max + 4.6796 * radiacion_global - 10.0781

Number of Rules : 1
Time taken to build model: 18.3 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9112
Mean absolute error	2.0052
Root mean squared error	2.5594
Relative absolute error	39.683 %
Root relative squared error	41.1859 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	16.22 s	0.9031	2.0875	2.6913
500	0.2	10	0.01	16.42 s	0.9058	2.0506	2.65
500	0.2	15	0.01	25.83 s	0.9067	2.0531	2.6582
500	0.01	15	0.01	24.6 s	0.9088	2.0255	2.5926
3000	0.01	15	0.01	157.13 s	0.9115	1.9954	2.5564

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15 (1 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 2500

Los valores obtenidos son:

Time taken to build model: 147.88 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9117
Mean absolute error	1.9942
Root mean squared error	2.5541
Relative absolute error	39.4647 %
Root relative squared error	41.0995 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la temperatura mínima a 24h:

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h = -0.0494 * dia + 0.7264 * mes + -0.2329 * ano +15.5319 * tem_med + -4.6949 * tem_max +20.2339 * tem_min + 6.1893 * humedad_med + -2.1851 * humedad_max + -1.9793 * humedad_min + 0.2885 * precip_acumulada + 0.0857 * vel_media_viento +0.0026 * direc_viento + -1.3543 * vel_racha_max + -0.5534 * direc_racha_max + 2.5106 * radiacion_global + 0.433 * insolacion + -11.8482

Time taken to build model: 2.64 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9136
Mean absolute error	1.7448
Root mean squared error	2.2991
Relative absolute error	37.7705 %
Root relative squared error	40.6735 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h = 0.7102 * mes + -0.2102 * ano + 15.7535 * tem_med + -5.0864 * tem_max + 21.0248 * tem_min + 6.1853 * humedad_med + -2.145 * humedad_max + -1.9529 * humedad_min + -1.0483 * vel_racha_max + -0.5647 * direc_racha_max + 2.3689 * radiacion_global + 0.5914 * insolacion + -12.3043

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9136
Mean absolute error	1.7374
Root mean squared error	2.2975
Relative absolute error	37.6105 %
Root relative squared error	40.6457 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 4.06 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9026
Mean absolute error	1.7964
Root mean squared error	2.4333
Relative absolute error	38.8875 %
Root relative squared error	43.0468 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:

tem_min <= 0.488 : | tem_med <= 0.325 : LM1 (445/37.53%)
| tem_med > 0.325 : LM2 (1071/38.336%)
tem_min > 0.488 : LM3 (1682/40.308%)

LM num: 1

tem_min_24h = 0.023 * mes - 0.0021 * ano - 4.6667 * tem_med + 17.2769 * tem_max + 23.4234 * tem_min + 0.1015 * humedad_med - 0.021 * humedad_max - 0.0444 * humedad_min + 0.9789 * vel_media_viento - 0.0103 * vel_racha_max - 0.0167 * direc_racha_max + 0.0931 * radiacion_global + 1.9641 * insolacion - 11.6712

LM num: 2

tem_min_24h = 0.6374 * mes - 0.2949 * ano + 7.2089 * tem_med - 1.6449 * tem_max + 21.905 * tem_min + 0.6715 * humedad_med - 0.021 * humedad_max - 0.4966 * humedad_min - 1.2306 * vel_media_viento - 0.0103 * vel_racha_max - 0.0103 * direc_racha_max + 2.9894 * radiacion_global + 0.012 * insolacion - 8.9708

LM num: 3

tem_min_24h = 1.7161 * mes - 0.0019 * ano + 26.6606 * tem_med - 11.5119 * tem_max + 12.5182 * tem_min + 8.3965 * humedad_med - 4.3489 * humedad_max - 1.9088 * humedad_min - 1.46 * vel_racha_max - 0.6606 * direc_racha_max + 3.6274 * radiacion_global + 0.0052 * insolacion - 10.6195

Time taken to build model: 2.86 seconds

=== Cross-validation ===

Correlation coefficient	0.9167
Mean absolute error	1.7141
Root mean squared error	2.2579
Relative absolute error	37.1049 %
Root relative squared error	39.9451 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
M5 pruned model tree:

tem_min <= 0.496 :
|   tem_min <= 0.346 : LM1 (536/39.756%)
|   tem_min >  0.346 : LM2 (1042/36.859%)
tem_min >  0.496 :
|   tem_min <= 0.622 :
|   |   tem_med <= 0.55 : LM3 (415/31.98%)
|   |   tem_med >  0.55 : LM4 (540/37.529%)
|   tem_min >  0.622 :
|   |   tem_min <= 0.704 : LM5 (429/37.719%)
|   |   tem_min >  0.704 :
|   |   |   tem_min <= 0.736 : LM6 (98/34.106%)
|   |   |   tem_min >  0.736 :
|   |   |   |   insolacion <= 0.769 : LM7 (67/43.689%)
|   |   |   |   insolacion >  0.769 :
|   |   |   |   |   mes <= 0.591 : LM8 (49/58.377%)
|   |   |   |   |   mes >  0.591 : LM9 (22/50.784%)

Time taken to build model: 20.81 seconds

=== Cross-validation ===

Correlation coefficient                  0.9181
Mean absolute error                      1.6941
Root mean squared error                  2.2403
Relative absolute error                 36.6722 %
Root relative squared error             39.634  %
Total Number of Instances                3198
Ignored Class Unknown Instances            90
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	19.58 s	0.9033	1.841	2.4447
500	0.2	10	0.01	19.13 s	0.907	1.8139	2.3975
500	0.2	15	0.01	28.06 s	0.9043	1.8524	2.433
500	0.01	15	0.01	28.37 s	0.9155	1.7187	2.2742
3000	0.01	15	0.01	169.66 s	0.9178	1.7	2.2468

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15 (1 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 3.500

Los valores obtenidos son:

Time taken to build model: 207.29 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.918
Mean absolute error	1.6981
Root mean squared error	2.2438
Relative absolute error	36.7584 %
Root relative squared error	39.695 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos seleccionados

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

tem_min_24h =-0.0498 * dia + 0.7406 * mes + 15.6038 * tem_med +-4.7895 * tem_max + 20.2876 * tem_min + 6.1892 * humedad_med + -2.1553 * humedad_max + -1.9923 * humedad_min + 0.0222 * direc_viento + -1.2117 * vel_racha_max + -0.5348 * direc_racha_max + 2.5389 * radiacion_global + 0.4035 * insolacion + -12.0344

Time taken to build model: 2.52 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9137
Mean absolute error	1.7433
Root mean squared error	2.2984
Relative absolute error	37.7376 %
Root relative squared error	40.6601 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

tem_min_24h =0.7203 * mes +15.6767 * tem_med + -5.0962 * tem_max + 21.1324 * tem_min + 6.1599 * humedad_med +-2.1001 * humedad_max + -1.9708 * humedad_min + -1.018 * vel_racha_max + -0.5464 * direc_racha_max + 2.3959 * radiacion_global +0.5741 * insolacion + -12.4583

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9136
Mean absolute error	1.7369
Root mean squared error	2.2977
Relative absolute error	37.5987 %
Root relative squared error	40.6488 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 3.14 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9026
Mean absolute error	1.7964
Root mean squared error	2.4333
Relative absolute error	38.8875 %
Root relative squared error	43.0468 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:

tem_min <= 0.488 : | tem_med <= 0.325 : LM1 (445/37.629%) | tem_med > 0.325 : LM2 (1071/38.403%) tem_min > 0.488 : LM3 (1682/40.308%)

LM num: 1

tem_min_24h = 0.0231 * mes + 0.1995 * tem_med + 15.5201 * tem_max + 20.8895 * tem_min + 0.1013 * humedad_med - 0.0206 * humedad_max - 0.0446 * humedad_min - 0.01 * vel_racha_max - 0.0166 * direc_racha_max + 0.0934 * radiacion_global + 1.7115 * insolacion - 11.5029

LM num: 2

tem_min_24h = 0.6337 * mes + 5.3325 * tem_med + 0.0206 * tem_max + 22.1877 * tem_min + 0.0777 * humedad_med - 0.0206 * humedad_max - 0.03 * humedad_min - 1.1762 * vel_racha_max - 0.0101 * direc_racha_max + 2.9689 * radiacion_global + 0.0118 * insolacion - 8.9254

LM num: 3

tem_min_24h = 1.7162 * mes + 26.66 * tem_med - 11.512 * tem_max + 12.5192 * tem_min + 8.3962 * humedad_med - 4.3485 * humedad_max - 1.9089 * humedad_min - 1.4597 * vel_racha_max - 0.6604 * direc_racha_max + 3.6276 * radiacion_global + 0.0051 * insolacion - 10.6208

Number of Rules : 3
Time taken to build model: 2.48 seconds

=== Cross-validation ===

Correlation coefficient	0.9175
Mean absolute error	1.7041
Root mean squared error	2.2477
Relative absolute error	36.8894 %
Root relative squared error	39.7643 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

M5 pruned model tree:
(using smoothed linear models)

Number of Rules : 14

Time taken to build model: 19.03 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9182
Mean absolute error	1.6896
Root mean squared error	2.239
Relative absolute error	36.5753 %
Root relative squared error	39.6109 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	17.44 s	0.9076	1.8182	2.3921
500	0.2	10	0.01	16.33 s	0.9093	1.7992	2.3664
500	0.2	15	0.01	24.13 s	0.9096	1.7893	2.3638
500	0.01	15	0.01	23.73 s	0.9154	1.7226	2.2763
3000	0.01	15	0.01	143.68 s	0.9181	1.6945	2.2428

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15 (1 capa oculta de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 4.000

Los valores obtenidos son:

Time taken to build model: 197.14 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.9179
Mean absolute error	1.6931
Root mean squared error	2.2459
Relative absolute error	36.6508 %
Root relative squared error	39.7326 %
Total Number of Instances	3198
Ignored Class Unknown Instances	90

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con todos los atributos de entrada:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h = 0.0576 * dia +0.2473 * mes + -0.1252 * ano +-1.3697 * tem_med + 1.3898 * tem_max +-1.3111 * tem_min +0.1937 * humedad_med + -0.9964 * humedad_max + -0.0433 * humedad_min +-1.3871 * precip_acumulada +5.3927 * vel_media_viento +0.7109 * direc_viento + 2.5158 * vel_racha_max +-0.0932 * direc_racha_max + -2.1203 * radiacion_global + 1.0938 * insolacion + 6.9372

Time taken to build model: 2.6 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5769
Mean absolute error	2.3807
Root mean squared error	3.1843
Relative absolute error	81.4864 %
Root relative squared error	85.9116 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h = -1.1929 * tem_min + 0.9899 * humedad_med +-1.1529 * humedad_max + -0.4698 * humedad_min + -2.1917 * precip_acumulada + 4.8931 * vel_media_viento + 1.0606 * direc_viento +7.3316 * vel_racha_max + -0.3314 * direc_racha_max +-4.5616 * radiacion_global +2.9571 * insolacion + 6.4106

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5808
Mean absolute error	2.3216
Root mean squared error	3.0166
Relative absolute error	79.4622 %
Root relative squared error	81.3858 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 2.49 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5307
Mean absolute error	2.4305
Root mean squared error	3.1418
Relative absolute error	83.19 %
Root relative squared error	84.7648 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0):**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

M5 pruned model tree:
(using smoothed linear models)

LM1 (3150/80.915%)
LM num: 1

vel_media_viento_24h = -1.1929 * tem_min + 0.9899 * humedad_med - 1.1529 * humedad_max - 0.4698 * humedad_min - 2.1917 * precip_acumulada + 4.8931 * vel_media_viento + 1.0606 * direc_viento + 7.3316 * vel_racha_max - 0.3314 * direc_racha_max - 4.5616 * radiacion_global + 2.9571 * insolacion + 6.4106

Number of Rules: 1
Time taken to build model: 2.79 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5819
Mean absolute error	2.3234
Root mean squared error	3.0137
Relative absolute error	79.5259 %
Root relative squared error	81.3082 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

M5 pruned model tree:

vel_racha_max <= 0.248: LM1 (1648/62.445%)
vel_racha_max > 0.248 : | vel_racha_max <= 0.398 : LM2 (924/86.484%)
| vel_racha_max > 0.398 : LM3 (578/105.001%)

LM num: 1

vel_media_viento_24h = -0.0025 * ano - 10.4421 * tem_med + 4.7792 * tem_max + 4.2833 * tem_min + 0.0092 * humedad_med - 0.0121 * humedad_max - 0.0228 * precip_acumulada + 2.9042 * vel_media_viento + 1.0917 * direc_viento + 9.6344 * vel_racha_max - 0.0044 * direc_racha_max - 2.5361 * radiacion_global + 1.5649 * insolacion + 5.5234

LM num: 2

vel_media_viento_24h = 0.609 * mes - 0.0028 * ano + 7.1618 * tem_med + 0.0638 * tem_max - 8.6404 * tem_min + 0.0101 * humedad_med - 0.0133 * humedad_max - 2.675 * precip_acumulada + 5.3909 * vel_media_viento + 0.8965 * direc_viento + 0.1529 * vel_racha_max - 0.0198 * direc_racha_max - 7.8663 * radiacion_global + 5.5264 * insolacion + 7.7104

LM num: 3

vel_media_viento_24h = -0.0028 * ano - 12.479 * tem_med + 9.467 * tem_max - 0.1427 * tem_min - 7.0599 * humedad_med - 0.0133 * humedad_max + 3.8512 * humedad_min - 0.0823 * precip_acumulada + 4.4931 * vel_media_viento + 2.8856 * direc_viento + 0.2032 * vel_racha_max - 2.2421 * direc_racha_max - 12.4696 * radiacion_global + 9.1255 * insolacion + 14.8799

Number of Rules: 3, Time taken to build model: 21.66 seconds

=== Cross-validation ===

Correlation coefficient	0.5831
Mean absolute error	2.3182
Root mean squared error	3.0104
Relative absolute error	79.3454 %
Root relative squared error	81.2189 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild –** si esta a falso permite realizar la red neuronal manualmente introduciendo el numero de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	19.25 s	0.4705	2.6211	3.4644
500	0.2	10	0.01	2.5802	0.4887	2.5802	3.3901
500	0.2	15	0.01	27.75 s	0.4712	2.5901	3.4396
500	0.01	15	0.01	26.94 s	0.5776	2.3208	3.0305
2000	0.01	15	0.01	121.45 s	0.5745	2.324	3.0422

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15 (2 capas ocultas de 15 neuronas)
learningRate = 0.01
momentum = 0.01
trainingTime = 1000

Los valores obtenidos son:

Time taken to build model: 108.81 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5795
Mean absolute error	2.3146
Root mean squared error	3.0236
Relative absolute error	79.2242 %
Root relative squared error	81.574 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos de seleccionados:

- **LeastMedSq (-S 1 -G 0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-1.0461 * tem_min + -0.4135 * humedad_med +-1.3735 * precip_acumulada +5.4531 * vel_media_viento +0.6754 * direc_viento +2.4451 * vel_racha_max +-2.3474 * radiacion_global +1.2481 * insolacion +6.3977

Time taken to build model: 1.85 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5785
Mean absolute error	2.3787
Root mean squared error	3.1826
Relative absolute error	81.4185 %
Root relative squared error	85.8663 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **LinearRegression (-S0 -R 1.0E-8):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Linear Regression Model

vel_media_viento_24h =-1.0629 * tem_min +-2.3164 * precip_acumulada +5.0028 * vel_media_viento +0.9558 * direc_viento +7.1581 * vel_racha_max +-4.7354 * radiacion_global + 3.1713 * insolacion +5.6536

Time taken to build model: 0.01 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5828
Mean absolute error	2.3162
Root mean squared error	3.0112
Relative absolute error	79.2797 %
Root relative squared error	81.2402 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Isotonic regression:**

  Los valores obtenidos son:

Time taken to build model: 1.39 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5307
Mean absolute error	2.4305
Root mean squared error	3.1418
Relative absolute error	83.19 %
Root relative squared error	84.7648 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **M5P (-M 1.0) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

M5 pruned model tree:
(using smoothed linear models)
LM1 (3150/80.977%)

LM num: 1

vel_media_viento_24h = -1.0629 * tem_min - 2.3164 * precip_acumulada + 5.0028 * vel_media_viento + 0.9558 * direc_viento + 7.1581 * vel_racha_max - 4.7354 * radiacion_global + 3.1713 * insolacion + 5.6536

Number of Rules : 1

Time taken to build model: 2.9 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5828
Mean absolute error	2.3162
Root mean squared error	3.0112
Relative absolute error	79.2797 %
Root relative squared error	81.2402 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Bagging (M5P (-P 100 -S 1 -I 10 -W weka.classifiers.trees.M5P -- -M 1.0):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
M5 pruned model tree:
    (using smoothed linear models)
    vel_racha_max <= 0.248 : LM1 (1648/62.558%)
    vel_racha_max >  0.248 : LM2 (1502/95.974%)

LM num: 1

vel_media_viento_24h = -1.4867 * tem_min - 0.0228 * precip_acumulada +
2.7501 * vel_media_viento + 1.0854 * direc_viento + 9.7004 * vel_racha_max
    - 2.631 * radiacion_global + 1.5002 * insolacion + 5.505

    LM num: 2

vel_media_viento_24h = -1.1616 * tem_min - 2.4001 * precip_acumulada +
5.8466 * vel_media_viento + 1.4095 * direc_viento + 5.7037 * vel_racha_max
- 9.1035 * radiacion_global + 7.1564 * insolacion  + 5.8735

    Number of Rules : 2
    Time taken to build model: 14.64 seconds

    === Cross-validation ===
    === Summary ===
    Correlation coefficient                  0.5857
    Mean absolute error                      2.3094
    Root mean squared error                  3.0035
    Relative absolute error                 79.0461 %
    Root relative squared error             81.0331 %
    Total Number of Instances               3150
    Ignored Class Unknown Instances          138
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correlation coefficient	Mean absolute error	Mean squared error
500	0.2	10	0.2	10.15 s	0.4985	2.5444	3.352
500	0.2	10	0.01	10.13 s	0.5101	2.4997	3.2895
500	0.2	15	0.01	15.13 s	0.5069	2.5178	3.3092
500	0.01	15	0.01	14.42 s	0.5788	2.3161	3.027
2000	0.01	15	0.01	65.02 s	0.583	2.312	3.0154

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15,15
learningRate = 0.01
momentum = 0.01
trainingTime = 1.500

Los valores obtenidos son:

Time taken to build model: 141.84 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.5831
Mean absolute error	2.3109
Root mean squared error	3.0126
Relative absolute error	79.0958 %
Root relative squared error	81.2781 %
Total Number of Instances	3150
Ignored Class Unknown Instances	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Minería de datos a la Predicción de la precipitación acumulada a 24h:

Con todos los atributos de entrada:

- **NBTree:**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Time taken to build model: 1.4 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1483	57.975 %
Incorrectly Classified Instances	1075	42.025 %
Kappa statistic	0.266	
Mean absolute error	0.3478	
Root mean squared error	0.4257	
Relative absolute error	87.2724 %	
Root relative squared error	95.3644 %	
Total Number of Instances	2558	
Ignored Class Unknown Instances	730	

=== Confusion Matrix ===

a	b	c	<-- classified as
652	428	10	a = 0
348	769	45	b = 1
47	197	62	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **DecisionTable (-X 1 -S "weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1"):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

Time taken to build model: 0.72 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1480	57.8577 %
Incorrectly Classified Instances	1078	42.1423 %
Kappa statistic	0.2584	
Mean absolute error	0.3566	
Root mean squared error	0.4238	
Relative absolute error	89.4637 %	
Root relative squared error	94.9525 %	
Total Number of Instances	2558	
Ignored Class Unknown Instances	730	

=== Confusion Matrix ===

a	b	c	<-- classified as
693	390	7	a = 0
394	743	25	b = 1
59	203	44	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Logistic (-R 1.0E-8 -M -1):**
  Los valores obtenidos son:

```
Time taken to build model: 0.72 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1451               56.724  %
Incorrectly Classified Instances      1107               43.276  %
Kappa statistic                       0.2348
Mean absolute error                   0.3541
Root mean squared error               0.4224
Relative absolute error              88.8492 %
Root relative squared error          94.627 %
Total Number of Instances             2558
Ignored Class Unknown Instances        730

=== Confusion Matrix ===
   a   b   c   <-- classified as
 668 415   7 |   a = 0
 376 759  27 |   b = 1
  64 218  24 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **BayesNet (-D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5) :**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1343	52.502 %
Incorrectly Classified Instances	1215	47.498 %
Kappa statistic	0.2051	
Mean absolute error	0.344	
Root mean squared error	0.476	
Relative absolute error	86.3185 %	
Root relative squared error	106.6246 %	
Total Number of Instances	2558	
Ignored Class Unknown Instances	730	

=== Confusion Matrix ===

```
  a   b   c   <-- classified as
623 411  56 |   a = 0
380 626 156 |   b = 1
 64 148  94 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **RBFNetwork (-B 2 -S 1 -R 1.0E-8 -M -1 -W 0.1):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

Time taken to build model: 0.32 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1419	55.473 %
Incorrectly Classified Instances	1139	44.527 %
Kappa statistic	0.2038	
Mean absolute error	0.3661	
Root mean squared error	0.4295	
Relative absolute error	91.8476 %	
Root relative squared error	96.2269 %	
Total Number of Instances	2558	
Ignored Class Unknown Instances	730	

=== Confusion Matrix ===

a	b	c	<-- classified as
613	474	3	a = 0
354	800	8	b = 1
60	240	6	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers --** Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error
500	0.2	10	0.2	17.78 s	1423 (55.6294 %)	1135 (44.370%)	0.3518
500	0.2	10	0.01	17.67 s	1429 (55.864 %)	1129 (44.136 %)	0.351
500	0.2	15	0.01	26.18 s	1402 (54.8084 %)	1156 (45.191%)	0.3496
500	0.01	15	0.01	25.93 s	1452 (56.7631 %)	1106 (43.236%)	0.355
700	0.01	15	0.01	35.5 s	1465 (57.2713 %)	1093 (42.7287%)	0.3542

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15
learningRate = 0.01
momentum = 0.01
trainingTime = 800

Los valores obtenidos son:

```
Time taken to build model: 41.22 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1467               57.3495 %
Incorrectly Classified Instances      1091               42.6505 %
Kappa statistic                          0.2431
Mean absolute error                      0.3539
Root mean squared error                  0.4257
Relative absolute error                 88.7805 %
Root relative squared error             95.359  %
Total Number of Instances             2558
Ignored Class Unknown Instances        730

=== Confusion Matrix ===
   a   b   c   <-- classified as
 630 453   7 |   a = 0
 325 815  22 |   b = 1
  54 230  22 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Con los atributos seleccionados de entrada:

- **NBTree:**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1478               57.7795 %
Incorrectly Classified Instances      1080               42.2205 %
Kappa statistic                          0.2657
Mean absolute error                      0.3584
Root mean squared error                  0.4262
Relative absolute error                 89.9162 %
Root relative squared error             95.4773 %
Total Number of Instances             2558
Ignored Class Unknown Instances        730

=== Confusion Matrix ===

   a   b   c   <-- classified as
 757 322  11 |   a = 0
 454 662  46 |   b = 1
  67 180  59 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **DecisionTable(-X 1 -S "weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1"):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son:

```
Time taken to build model: 0.1 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1473               57.5841 %
Incorrectly Classified Instances      1085               42.4159 %
Kappa statistic                          0.2618
Mean absolute error                      0.3594
Root mean squared error                  0.4251
Relative absolute error                 90.1715 %
Root relative squared error             95.2356 %
Total Number of Instances             2558
Ignored Class Unknown Instances        730

=== Confusion Matrix ===
   a   b   c   <-- classified as
 749 330  11 |   a = 0
 451 666  45 |   b = 1
  66 182  58 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **Logistic (-R 1.0E-8 -M -1):**

  Los valores obtenidos son:

Time taken to build model: 0.16 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1388	54.2611 %
Incorrectly Classified Instances	1170	45.7389 %
Kappa statistic	0.1934	
Mean absolute error	0.378	
Root mean squared error	0.4343	
Relative absolute error	94.8348 %	
Root relative squared error	97.304 %	
Total Number of Instances	2558	
Ignored Class Unknown Instances	730	

=== Confusion Matrix ===

a	b	c	<-- classified as
818	268	4	a = 0
602	549	11	b = 1
91	194	21	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **BayesNet ((-D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5) :**

  Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

  Los valores obtenidos son

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1478	57.7795 %
Incorrectly Classified Instances	1080	42.2205 %
Kappa statistic	0.2657	
Mean absolute error	0.3583	
Root mean squared error	0.4262	
Relative absolute error	89.8863 %	
Root relative squared error	95.4847 %	
Total Number of Instances	2558	
Ignored Class Unknown Instances	730	

=== Confusion Matrix ===

a	b	c	<-- classified as
757	322	11	a = 0
454	662	46	b = 1
67	180	59	c = 2

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **RBFNetwork (-B 3 -S 1 -R 1.0E-8 -M -1 -W 0.1):**

Entre paréntesis aparecen los parámetros elegidos para el modelo en cuestión.

Los valores obtenidos son

```
Time taken to build model: 0.17 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1328               51.9156 %
Incorrectly Classified Instances      1230               48.0844 %
Kappa statistic                          0.1474
Mean absolute error                      0.3771
Root mean squared error                  0.4355
Relative absolute error                 94.6223 %
Root relative squared error             97.5737 %
Total Number of Instances             2558
Ignored Class Unknown Instances        730

=== Confusion Matrix ===

   a   b   c   <-- classified as
 639 446   5 |   a = 0
 483 667  12 |   b = 1
  72 212  22 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- **MultilayerPerceptron:**

A la hora de evaluar este modelo he realizado diferentes pruebas cambiando estos parámetros y con ello he sacado las siguientes conclusiones:

- **autoBuild** – si esta a falso permite realizar la red neuronal manualmente introduciendo el número de capas ocultas y neuronas que quieras. Si esta a verdadero crea la red neuronal de forma automática de a cuerdo al número de neuronas que le has introducido (hiddenLayers).

- **hiddenLayers** -- Define el numero de neuronas en cada capa oculta. Normalmente a más neuronas el coste en tiempo y procesamiento se aumenta ya que tiene que calcular los pesos de más neuronas. En primer lugar empiezo con pocas neuronas e una sola capa para luego aumentar su número (el máximo número de capas ocultas es 2 ya que está visto que con este número es suficiente) y encontrar el numero de neuronas y capas ocultas que mejor resultado me den.

- **learningRate** – Tasa de aprendizaje. Este parámetro define cuanto de rápido queremos avanzar en la dirección opuesta al gradiente del error (dirección que minimice el error).Por tanto con una tasa de aprendizaje alta se irá rápidamente hacia el mínimo del error y los pesos de la red se actualizaran más rápidamente. Sin embargo al tener una tasa alta se puede perder detalle y no llegar a converger a un mínimo mientras que con una tasa de aprendizaje pequeña se avanza en la función de error de más en poco a poco.

- **momentum** – El momento permite a la red ignorar características pequeñas o de mínimo significado en la superficie del error. Sin el momento una red puede estancarse en un mínimo local, con momento la red puede deslizarse e evitar caer en un mínimo local. Por tanto es un parámetro a tener en cuenta.

- **trainingTime** – numero de épocas que dura el entrenamiento (aprendizaje de la red neuronal). Normalmente a mas épocas (iteraciones) la predicción mejora (se minimiza el error) pero también se puede llegar a un momento que de tanta iteraciones se produzca un sobre entrenamiento dando lugar a empeoramiento de los resultados. Por tanto hay que buscar el número de épocas que minimice el error de la validación y no del entrenamiento.

Los valores obtenidos son:

trainingTime	learningRate	hiddenLayers	Momentum	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error
500	0.2	10	0.2	7.86 s	1380 (53.9484 %)	1178 (46.0516 %)	0.3687
500	0.2	10	0.01	7.91 s	1369 (53.5184 %)	1189 (46.4816 %)	0.369
500	0.2	15	0.01	11.28 s	1382 (54.0266 %)	1176 (45.9734 %)	0.3683
500	0.01	15	0.01	10.74 s	1394 (54.4957 %)	1164 (45.5043 %)	0.3775
1000	0.01	15	0.01	22.4 s	1418 (55.4339 %)	1140 (44.5661 %)	0.3724

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último tras la realización de distintas pruebas modificando los parámetros hiddenLayers, learningRate, momentum y trainingTime encontré los valores donde la red neuronal daba los mejores resultados en su evaluación:

hiddenLayers = 15
learningRate = 0.01
momentum = 0.01
trainingTime = 1500

Los valores obtenidos son:

```
Time taken to build model: 33.7 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1427               55.7858 %
Incorrectly Classified Instances      1131               44.2142 %
Kappa statistic                          0.2203
Mean absolute error                      0.3706
Root mean squared error                  0.4328
Relative absolute error                 92.9805 %
Root relative squared error             96.9606 %
Total Number of Instances             2558
Ignored Class Unknown Instances        730

=== Confusion Matrix ===
   a   b   c   <-- classified as
 837 249   4 |   a = 0
 581 569  12 |   b = 1
  82 203  21 |   c = 2
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Evaluación e interpretación

A la hora de evaluar los modelos que he construido en la fase de minería de datos me voy a centrar en analizar los resultados de la evaluación de estos modelos con la validación cruzada con 10 pliegues. Este método de evaluación es muy preciso ya que entrena con todos los pliegues menos uno y luego evalúa el modelo creado con el pliegue que no se ha utilizado para entrenar. Este proceso se repite tantas veces como pliegues halla, escogiendo siempre un conjunto de pliegues distintos para entrenar y otro para evaluar.

Predicción de la temperatura máxima a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.69 s	0.8841	2.5658	3.2448	45.687%	46.719 %	3198	90
LinearRegression	0.04 s	0.8844	2.5633	3.2412	45.642 %	46.666 %	3198	90
Isotonic regression	3.86 s	0.859	2.7922	3.5558	49.719%	51.197%	3198	90
LeastMedSq	2.54 s	0.8845	2.564	3.242	45.654%	46.678%	3198	90
MultilayerPerceptron	178.9s	0.8885	2.5112	3.1893	44.715%	45.920%	3198	90
Bagging (M5P - M 1.0) 10 iteraciones	22.41 s	0.8849	2.5576	3.2348	45.540%	46.575%	3198	90

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.31 s	0.8849	2.5569	3.2343	45.528%	46.567%	3198	90
LinearRegression	0.02 s	0.8849	2.5569	3.2343	45.528%	46.567%	3198	90
Isotonic regression	1.83 s	0.859	2.7922	3.5558	49.719%	51.197%	3198	90
LeastMedSq	2.14 s	0.8848	2.5604	3.2387	45.59 %	46.630%	3198	90
MultilayerPerceptron	98.44 s	0.8874	2.5271	3.2017	44.998 %	46.098%	3198	90
Bagging (M5P - M 1.0) 10 iteraciones	17.71 s	0.8858	2.5487	3.2221	45.382%	46.392%	3198	90

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura máxima al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 2.5112 grados y un error relativo de un 44.715% .Pese a ello los resultados los considero satisfactorios aunque no del todo precisos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 (bagging) y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye un poco y que los resultados mejoran (M5P, LinearRegression, LeastMedSq y Bagging) con unos modelos y empeoran (MultilayerPerceptron) con otros pero no varían mucho frente a los obtenidos con todos los atributos.

Predicción de la temperatura media a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	1.9 s	0.909	2.0283	2.5895	40.140%	41.67 %	3198	90
LinearRegression	0.03 s	0.9093	2.0212	2.5851	40.000%	41.598%	3198	90
Isotonic regression	3.15 s	0.8909	2.1957	2.8226	43.452%	45.421%	3198	90
LeastMedSq	1.67 s	0.9092	2.0243	2.5874	40.061%	41.636%	3198	90
MultilayerPerceptron	110.54 s	0.9116	1.9941	2.5555	39.462%	41.123%	3198	90
Bagging (M5P - M 1.0) 10 iteraciones	18.45 s	0.9105	2.0119	2.5687	39.815%	41.335%	3198	90

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	1.65 s	0.909	2.0245	2.5903	40.065%	41.683%	3198	90
LinearRegression	0.02 s	0.9084	2.0296	2.5975	40.165%	41.799%	3198	90
Isotonic regression	2.47 s	0.8909	2.1957	2.8226	43.452%	45.421%	3198	90
LeastMedSq	1.42 s	0.9083	2.0329	2.5998	40.230%	41.835 %	3198	90
MultilayerPerceptron	147.88 s	0.9117	1.9942	2.5541	39.464%	41.099%	3198	90
Bagging (M5P - M 1.0) 10 iteraciones	18.3 s	0.9112	2.0052	2.5594	39.683 %	41.185%	3198	90

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura media al día siguiente se observa que el mejor resultado se obtiene mediante el Perceptrón multicapa con el cual se obtiene un error absoluto de 1.9941 grados y un error relativo de un 39.462%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el árbol de decisión MP5 (bagging) y por detrás se encuentran las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye un poco y que los resultados mejoran (M5P) con unos modelos y empeoran (MultilayerPerceptron , LinearRegression, LeastMedSq y Bagging) con otros pero no varían mucho frente a los obtenidos con todos los atributos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Predicción de la temperatura mínima a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	UnknownInstances
M5P	2.86 s	0.9167	1.7141	2.2579	37.104%	39.945%	3198	90
LinearRegression	0.04 s	0.9136	1.7374	2.2975	37.610%	40.645%	3198	90
Isotonic regression	4.06 s	0.9026	1.7964	2.4333	38.887%	43.046%	3198	90
LeastMedSq	2.64 s	0.9136	1.7448	2.2991	37.770%	40.673%	3198	90
MultilayerPerceptron	207.29 s	0.918	1.6981	2.2438	36.758%	39.695 %	3198	90
Bagging (M5P - M 1.0) 10 iteraciones	20.81 s	0.9181	1.6941	2.2403	36.672%	39.634%	3198	90

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.48 s	0.9175	1.7041	2.2477	36.889%	39.764%	3198	90
LinearRegression	0.02 s	0.9136	1.7369	2.2977	37.598%	40.648%	3198	90
Isotonic regression	3.14 s	0.9026	1.7964	2.4333	38.887%	43.046%	3198	90
LeastMedSq	2.52 s	0.9137	1.7433	2.2984	37.737%	40.660%	3198	90
MultilayerPerceptron	197.14 s	0.9179	1.6931	2.2459	36.650%	39.732%	3198	90
Bagging (M5P - M 1.0) 10 iteraciones	19.03 s	0.9182	1.6896	2.239	36.575%	39.610%	3198	90

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura mínima al día siguiente se observa que el mejor resultado se obtiene mediante el modelo compuesto Bagging (sobre el árbol de decisión MP5) con el cual se obtiene un error absoluto de 1.6896 grados y un error relativo de un 36.57%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el perceptrón multicapa y por detrás se encuentran el árbol de decisión MP5 y las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye con la mayoría y que los resultados mejoran (M5P, Bagging, LinearRegression, LeastMedSq , MultilayerPerceptron) con unos modelos y se igualan (Isotonic regression) con otros pero prácticamente no varían frente a los obtenidos con todos los atributos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa de los mejores resultados para la predicción de las temperaturas:

Temperatura	Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error
Temperatura máxima	**MultilayerPerceptron**	178.9s	0.8885	2.5112	3.1893	44.7152 %	45.9201 %
Temperatura media	**MultilayerPerceptron**	110.54 s	0.9116	1.9941	2.5555	39.4627 %	41.1231 %
Temperatura mínima	**Bagging (M5P -M 1.0) 10 iteraciones**	19.03 s	0.9182	1.6896	2.239	36.5753 %	39.6109 %

Como se observa las mejores predicciones las he obtenido mediante el método perceptrón multicapa y el método compuesto Bagging sobre un árbol de decisión MP5. A demás se aprecia claramente que la predicción de la temperatura mínima es la más precisa (error relativo de un 36,57%) seguida de la predicción de la temperatura media (39.46%) y ya más atrás la temperatura máxima (44.71%).

Predicción de la velocidad media del viento a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	2.79 s	0.5819	2.3234	3.0137	79.525%	81.308%	3150	138
LinearRegression	0.04 s	0.5808	2.3216	3.0166	79.462%	81.385%	3150	138
Isotonic regression	2.49 s	0.5307	2.4305	3.1418	83.19 %	84.764%	3150	138
LeastMedSq	2.6 s	0.5769	2.3807	3.1843	81.486%	85.911%	3150	138
MultilayerPerceptron	108.81 s	0.5795	2.3146	3.0236	79.224%	81.574%	3150	138
Bagging (M5P -M 1.0) 10 iteraciones	21.66 s	0.5831	2.3182	3.0104	79.345%	81.218%	3150	138

Con los atributos seleccionados:

Método	Time to build	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
M5P	1.53 s	0.5828	2.3162	3.0112	79.279%	81.240%	3150	138
LinearRegression	0.01 s	0.5828	2.3162	3.0112	79.279%	81.240%	3150	138
Isotonic regression	1.39 s	0.5307	2.4305	3.1418	83.19 %	84.764%	3150	138
LeastMedSq	1.85 s	0.5785	2.3787	3.1826	81.418%	85.866%	3150	138
MultilayerPerceptron	141.84 s	0.5831	2.3109	3.0126	79.095%	81.278%	3150	138
Bagging (M5P -M 1.0) 10 iteraciones	14.64 s	0.5857	2.3094	3.0035	79.046%	81.033%	3150	138

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la velocidad media del viento al día siguiente se observa que el mejor resultado se obtiene mediante el modelo compuesto Bagging (sobre el árbol de decisión MP5) con el cual se obtiene un error absoluto de 2.3094 m/s y un error relativo de un 79.046%. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante el perceptrón multicapa árbol y por detrás se encuentran el árbol de decisión MP5 y las regresiones que a pesar de ser más rápidas en cómputo dan peores resultados. Otro aspecto a destacar es que al realizar las predicciones sobre los atributos seleccionados y no sobre el total de atributos se puede ver que el tiempo de construcción de los modelos disminuye considerablemente y que los resultados mejoran (LinearRegression, M5P, Bagging, LeastMedSq, MultilayerPerceptron) o se igualan (Isotonic Regresion).

Predicción de la precipitación acumulada a 24h:

Con todos los atributos de entrada:

Método	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
NBTree	1.4 s	1483 (57.975 %)	1075 (42.025 %)	0.3478	0.4257	87.272%	95.3644 %	2558	730
Decision Table	0.72 s	1480 (57.8577 %)	1078 (42.1423%)	0.3566	0.4238	89.463%	94.9525 %	2558	730
Logistic	0.72 s	1451 (56.724 %)	1107 (43.276 %)	0.3541	0.4224	88.849%	94.627 %	2558	730
BayesNet	0.05 s	1343 (52.502 %)	1215 (47.498 %)	0.344	0.476	86.318%	106.6246 %	2558	730
RBFNetwork	0.32 s	1419 (55.473 %)	1139 (44.527%)	0.3661	0.4295	91.847%	96.2269 %	2558	730
MultilayerPerceptron	41.22 s	1467 (57.3495%)	1091 (42.650%)	0.3539	0.4257	88.780%	95.359 %	2558	730

Con los atributos seleccionados:

Método	Time to build	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error	Total Number of Instances	Unknown Instances
NBTree	0.05 s	1478 (57.7795 %)	1080 (42.220%)	0.3584	0.4262	89.916%	95.4773 %	2558	730
Decision Table	0.1 s	1473 (57.5841%)	1085 (42.415%)	0.3594	0.4251	90.171%	95.2356 %	2558	730
Logistic	0.16 s	1478 (57.7795%)	1080 (42.220%)	0.3583	0.4262	89.886%	95.4847 %	2558	730
BayesNet	0.02 s	1328 (51.9156 %)	1230 (48.0844%)	0.3771	0.4355	94.622%	97.5737 %	2558	730
RBFNetwork	0.17 s	1328 (51.9156 %)	1230 (48.0844%)	0.3771	0.4355	94.622%	97.5737 %	2558	730
MultilayerPerceptron	33.7 s	1427 (55.7858 %)	1131 (44.2142%)	0.3706	0.4328	92.980%	92.9805 %	2558	730

Como se observa tras aplicar diferentes métodos de minería de datos para predecir la temperatura mínima al día siguiente se observa que el mejor resultado se obtiene mediante el árbol de decisión **NBTree** con el cual se clasifican correctamente un 57.97 % de las instancias. Pese a ello los resultados los considero satisfactorios aunque no del todo precisos. El segundo mejor resultado se obtiene mediante una regresión Logística con un 57.85% de las instancias clasificadas correctamente. En los resultados de todos las clasificaciones se observa que la tasa de acierto de las instancias en la clase 1 (no llueve) es de entorno al 59%, la tasa de acierto de las instancias de la clase 2 (llueve débilmente) es de un entorno 66% y la tasa de acierto de las instancias de la clase 3 (llueve fuerte) es de un 20%.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Dado que la mayoría (1162 instancias, el 45%) de las instancias pertenecían a la clase 2 (hay poca precipitación acumulada) y luego (1090 instancias) a la clase 1 (no hay poca precipitación acumulada) al final la tasa de acierto general es de un 57.97%. De estos resultados se deduce que clasifica más o menos bien las instancias de la clase 2 y las de la clase 1 y clasifica fatal las de la clase 3 (20% de acierto).

Otro aspecto a destacar es que al realizar la clasificación sobre los atributos seleccionados (3 atributos) y no sobre el total (23 atributos) de atributos se puede ver que el tiempo de construcción de los modelos disminuye bastante en y que los resultados empeoran frente a los obtenidos con todos los atributos excepto con la regresión Logística.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa de los resultados de las Estaciones:

Antes de comenzar esta comparativa, hay que tomar como punto de partida el hecho que para la predicción de las temperaturas, velocidad media del viento y precipitación acumulada a 24h se han tomado como datos de entrenamiento, los datos meteorológicos de las estaciones de los años 2000 a 2008 . Dado que cada una de las estaciones disponía de distintos parámetros (atributos) meteorológicos, estos han podido influir a la hora de la predicción.

Del conjunto de atributos meteorológicos de las estaciones podemos encontrar 16 atributos en común los cuales son: **día, mes, año, temperatura media, temperatura máxima (a 2m), temperatura mínima (a 2m), humedad media (relativa), humedad máxima (relativa), humedad mínima (relativa), precipitación acumulada, velocidad media del viento, dirección del viento, velocidad de la racha máxima del viento, dirección de la racha máxima del viento, radiación global, insolación total**.

En el caso de la estación de **Gorramendi** contamos solo con loas 16 atributos comunes a las 3 estaciones. En la estación de **Larrabide** a parte de los atributos comunes contamos con otro atributo la **presión atmosférica media**. Por último en la estación de **Montes de Cierzo** a parte de los atributos comunes contamos con la **temperatura media aire a 150m**, **temperatura media suelo a 50m**, t**emperatura media suelo a 200m**, **velocidad media del viento a 2m**, **velocidad de la racha máxima a 2m**, **radiación solar reflectada media** , **dirección de la racha máxima a 2m**.

A continuación voy a proceder a realizar una comparativa entre las 3 estaciones a dos niveles. En el primer nivel comparare los resultados de la selección de atributos más importantes en las tres estaciones y en el segundo nivel comparare los resultados de las técnicas de minería de datos para las tres estaciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa a nivel de selección de atributos:

A la hora de comparar los resultados de la selección de atributos para cada una de las estaciones nos vamos a centrar en los resultados obtenidos de aplicar el algoritmo **RelieFAttributeEval**. Este algoritmo nos devuelve una lista ordenada con los atributos más influyentes a la hora de clasificar el atributo objetivo.

Selección de atributos para la predicción de la temperatura máxima a 24h.

Para la realización de comparativa en la siguiente tabla voy a mostrar los 10 primeros atributos más influyentes en la predicción de la temperatura máxima a 24h para cada una de las estaciones.

Estación de Larrabide		**Estación de Gorramendi**		**Estación de Montes de Cierzo**	
Atributo	**Importancia**	**Atributo**	**Importancia**	**Atributo**	**Importancia**
Mes	0.010691	Mes	0.008677	Mes	0.010265
Temperatura máxima	0.004089	Temperatura mínima	0.008135	Insolación	0.006925
Temperatura media	0.00405	Temperatura media	0.006631	Temperatura media	0.006787
Temperatura mínima	0.00268	Temperatura máxima	0.005994	año	0.006645
Año	0.002303	Año	0.005727	Temperatura máxima	0.004444
Día	0.001468	Humedad mínima	0.003583	Temperatura media a 150m	0.004413
Radiación global	0.000884	Radiación global	0.003425	Temperatura media suelo a 50m	0.00417
Precipitación acumulada	0.000246	Dirección del viento	0.003267	Día	0.003269
Dirección de la racha máxima	-0.000155	Humedad media	0.002282	Velocidad med viento a 2m	0.002002
Velocidad de la racha máxima	-0.000499	Dirección de la racha máxima	0.00207	Velocidad media viento	0.001784

Como se observa las temperaturas (media, máxima y mínima) son junto al mes, los parámetros más influyentes en la predicción de la temperatura máxima a 24 h en las tres estaciones. También se observa que son bastante importantes también el año, la radiación global y la insolación. Además en la estación de Montes de Cierzo el atributo exclusivo (de esta estación) **temperatura media del suelo a 50m** es de los más influyentes a la hora de predecir la temperatura máxima a 24h.

Con ello se pueden concluir que los atributos más importantes a la hora de predecir la temperatura máxima son parecidos en las 3 estaciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos para la predicción de la temperatura mínima a 24h.

Para la realización de comparativa en la siguiente tabla voy a mostrar los 10 primeros atributos más influyentes en la predicción de la temperatura mínima a 24h para cada una de las estaciones.

Estación de Larrabide		**Estación de Gorramendi**		**Estación de Montes de Cierzo**	
Atributo	**Importancia**	**Atributo**	**Importancia**	**Atributo**	**Importancia**
Temperatura media	0.0092634	Temperatura media	0.0128161	Insolación	0.01173
Temperatura mínima	0.0083826	Temperatura mínima	0.0104094	Temperatura máxima	0.010124
Mes	0.0081206	Mes	0.0085306	Mes	0.009083
Temperatura máxima	0.0057843	Temperatura máxima	0.0079421	Temperatura media	0.007583
Año	0.004287	año	0.0065825	Temperatura med suelo 50m	0.006002
Presión atmosférica	0.0029983	Humedad máxima	0.0058417	Temperatura media a 150m	0.004311
Día	0.0029284	Humedad media	0.0043938	Año	0.003216
Precipitación acumulada	0.001081	Humedad mínima	0.003369	Día	0.00117
Velocidad racha máxima	0.0010397	Dirección del viento	0.0029149	Humedad máxima	0.000864
Velocidad media viento	0.0009309	Radiación global	0.0015117	Temperatura mínima	0.000247

Como se observa las temperaturas (media, máxima y mínima) son junto al mes los parámetros más influyentes en la predicción de la temperatura mínima a 24 h en las tres estaciones. También se observa que son bastante importantes también el año, la radiación global, la insolación y las Humedades (máxima, media, mínima).

En la estación de Larrabide se observa que el atributo **presión atmosférica** que es exclusivo de esa estación también es de los más influyentes y puede beneficiarle a la hora de realizar las predicciones. También en la estación de Montes de cierzo los dos atributos exclusivos **temperatura media del suelo a 50m** y **temperatura media a 150m** son de los más influyentes a la hora de predecir la temperatura mínima a 24h.

Con ello se pueden concluir que los atributos más importantes a la hora de predecir la temperatura mínima a 24h son parecidos en las 3 estaciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos para la predicción de la temperatura media a 24h.

Para la realización de comparativa en la siguiente tabla voy a mostrar los 10 primeros atributos más influyentes en la predicción de la temperatura media a 24h para cada una de las estaciones.

Estación de Larrabide		**Estación de Gorramendi**		**Estación de Montes de Cierzo**	
Atributo	**Importancia**	**Atributo**	**Importancia**	**Atributo**	**Importancia**
Mes	0.0118613	Temperatura mínima	0.010786	Mes	0.012388
Temperatura media	0.0086337	Mes	0.00862	Insolación	0.01179
Temperatura mínima	0.0067844	Temperatura Media	0.008566	Temperatura media	0.00893
Temperatura máxima	0.0061183	Temperatura máxima	0.006933	Temperatura máxima	0.00866
Año	0.0044153	Año	0.005736	Año	0.00644
Día	0.0030759	Humedad mínima	0.003926	Temperatura med suelo 50m	0.006422
Precipitación acumulada	0.0015906	Humedad máxima	0.003873	Temperatura media 150m	0.005992
Presión Atmosférica	0.0012667	Humedad media	0.003703	Día	0.002287
Velocidad racha máxima	0.0005632	Dirección del viento	0.003439	Dirección racha máxima	0.00037
Radiación global	0.0000879	Radiación global	0.002359	Humedad mínima	0.000299

Como se observa las temperaturas (media, máxima y mínima) son junto al mes los parámetros más influyentes en la predicción de la temperatura media a 24 h en las tres estaciones. También se observa que son bastante importantes también el año, la radiación global, la insolación y las humedades (máxima, media y mínima).

En la estación de Larrabide se observa que el atributo **presión atmosférica** que es exclusivo de esa estación también es de los más influyentes y puede beneficiarle a la hora de realizar las predicciones. También en la estación de Montes de cierzo los dos atributos exclusivos **temperatura media del suelo a 50m** y **temperatura media a 150m** son de los más influyentes a la hora de predecir la temperatura media a 24h.

Con ello se pueden concluir que los atributos más importantes a la hora de predecir la temperatura media a 24h son parecidos en las 3 estaciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos para la predicción de la Velocidad media del viento a 24h.

Para la realización de comparativa en la siguiente tabla voy a mostrar los 10 primeros atributos más influyentes en la predicción de la Velocidad media del viento a 24h para cada una de las estaciones.

Estación de Larrabide		**Estación de Gorramendi**		**Estación de Montes de Cierzo**	
Atributo	**Importancia**	**Atributo**	**Importancia**	**Atributo**	**Importancia**
Velocidad med del viento	0.013285	Velocidad racha máxima	0.011294	Precipitación acumulada	0.005276
Presión atmosférica	0.009581	Velocidad media viento	0.011253	Dirección del ciento	0.004112
Velocidad de la racha máxima	0.007852	Año	0.009048	Día	0.004104
Precipitación acumulada	0.005591	Día	0.006598	Temperatura med suel 200m	0.003502
Día	0.005103	Dirección del viento	0.005134	Temperatura med suelo 50m	0.003395
Año	0.004233	Precipitación acumulada	0.004833	mes	0.003171
Humedad mínima	0.003044	Mes	0.003529	Año	0.002659
Temperatura mínima	0.002506	Insolación	0.003458	Humedad máxima	0.002627
Humedad máxima	0.002307	Temperatura media	0.00298	Temperatura Máxima	0.002616
Mes	0.002165	Temperatura máxima	0.002333	Insolación	0.00214

Como se observa la **velocidad media del viento**, **la velocidad de la racha máxima** y la **precipitación acumulada** son los parámetros más influyentes en la predicción de la **velocidad media del viento a 24h** en las tres estaciones. También se observa que son bastante importantes también el año, **la dirección del viento** y la **presión atmosférica**.

En la estación de Larrabide se observa que el atributo **presión atmosférica** que es exclusivo de esa estación también es de los más influyentes y puede beneficiarle a la hora de realizar las predicciones. También en la estación de Montes de cierzo los dos atributos exclusivos **temperatura media del suelo a 50m** y **temperatura media del suelo a 200m** son de los más influyentes a la hora de predecir la velocidad media del viento a 24h.

Con ello se pueden concluir que los atributos más importantes a la hora de predecir la velocidad media del viento a 24h son parecidos en las 3 estaciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Selección de atributos para la predicción de la precipitación acumulada a 24h.

Para la realización de comparativa en la siguiente tabla voy a mostrar los 10 primeros atributos más influyentes en la predicción de la precipitación acumulada 24h para cada una de las estaciones.

Estación de Larrabide		**Estación de Gorramendi**		**Estación de Montes de Cierzo**	
Atributo	**Importancia**	**Atributo**	**Importancia**	**Atributo**	**Importancia**
Año	0.02508	Velocidad racha máxima	0.011294	Año	
Día	0.01896	Velocidad media viento	0.011253	Dirección de la racha máxima	
Mes	0.01679	Año	0.009048	Día	
Presión atmosférica	0.01462	Día	0.006598	Mes	
Dirección del viento	0.01234	Dirección del viento	0.005134	Velocidad med del viento	
Insolación	0.01063	Precipitación acumulada	0.004833	Velocidad racha máxima	
Radiación Global	0.00828	Mes	0.003529	Radiación global	
Temperatura media	0.00742	Insolación	0.003458	Velocidad media del viento 2m	
Temperatura mínima	0.00729	Temperatura mínima	0.00298	Humedad media	
Humedad máxima	0.00667	Humedad mínima	0.002333	Precipitación Acumulada	

Como se observa la **velocidad racha máxima**, **el año, el mes** y la **velocidad media del viento** son los parámetros más influyentes en la predicción de la **precipitación acumulada a 24h** en las tres estaciones. También se observa que son bastante importantes también la **radiación**, **insolación**, **precipitación acumulada**.

En la estación de Larrabide se observa que el atributo **presión atmosférica** que es exclusivo de esa estación también es de los más influyentes (el tercero) y puede beneficiarle a la hora de realizar las predicciones. También en la estación de Montes de cierzo el atributo exclusivo velocidad media **del viento a 2m** es de los más influyentes a la hora de predecir la precipitación acumulada a 24h.

Con ello se pueden concluir que los atributos más importantes a la hora de predecir la precipitación acumulada a 24h son parecidos en las 3 estaciones.

En definitiva la selección de atributos en las distintas estaciones como hemos observados guardan bastante parecido aun que no son iguales.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa a nivel de resultados de los algoritmos de minería de datos:

A la hora de comparar los resultados (de a evaluación) de los algoritmos de minería de datos para cada una de las estaciones vamos a mostrar una serie de tablas comparativas.

Comparativa de algoritmos de minería de datos para la predicción de la temperatura máxima a 24h

A continuación voy a mostrar una tabla comparativa de los resultados de la evaluación de distintos algoritmos de minería de datos para la predicción de la temperatura máxima a 24h para cada una de las 3 estaciones. Para esta comparativa he elegido 3 métodos de minería de datos muy representativos los cuales son el árbol de regresión MP5, la regresión lineal y el perceptrón Multicapa. Hay que señalar que estos resultados se han obtenido al aplicar el correspondiente algoritmo de minería de datos sobre el conjunto de todos los datos (como hemos visto anteriormente se obtienen resultados muy parecidos que seleccionando) y con un método de evaluación cruzada con 10 pliegues.

Estación	Método	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error
Larrabide	**M5P**	0.9185	2.6133	3.312	37.104%	39.533%
Montes de cierzo	**M5P**	0.9303	2.3487	3.0346	33.668%	36.686%
Gorramendi	**M5P**	0.8841	2.5658	3.2448	45.687%	46.7194 %
Larrabide	**LinearRegression**	0.9159	2.6443	3.3628	37.543%	40.139%
Montes de cierzo	**LinearRegression**	0.9301	2.3474	3.0376	33.649%	36.723%
Gorramendi	**LinearRegression**	0.8844	2.5633	3.2412	45.642 %	46.6666 %
Larrabide	**MultilayerPerceptron**	0.9257	2.5019	3.1714	35.522%	37.855%
Montes de cierzo	**MultilayerPerceptron**	0.9357	2.2887	2.9284	32.8%	35.4%
Gorramendi	**MultilayerPerceptron**	0.8885	2.5112	3.1893	44.715%	45.920%

Como se observa al aplicar los algoritmos MP5 y LinearRegresion, la estación de montes de Cierzo obtiene los mejores resultados con error absoluto medio de 2.34 grados (error relativo de un 33%) seguida de la estación de Gorramendi con un error absoluto medio de 2.56 grados (error relativo de un 45%) y de la estación de Larrabide con un error absoluto de un 2.61 grados (error relativo de un 37%). Al aplicar el perceptrón multicapa sucede lo mismo los mejores resultados se obtienen en la estación de Montes de Cierzo seguida de la estación de Gorramendi y de la estación de Larrabide en este orden.

En cuanto a la comparación de los distintos algoritmos de minería de datos, se observa que el MultilayerPerceptrón obtiene los mejores resultados en cada una de las estaciones seguida del árbol de regresión MP5 y por último las regresiones lineales.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa de algoritmos de minería de datos para la predicción de la temperatura media a 24h

A continuación voy a mostrar una tabla comparativa de los resultados de la evaluación de distintos algoritmos de minería de datos para la predicción de la temperatura media a 24h para cada una de las 3 estaciones. Para esta comparativa he elegido 3 métodos de minería de datos muy representativos los cuales son el árbol de regresión MP5, la regresión lineal y el perceptrón Multicapa. Hay que señalar que estos resultados se han obtenido al aplicar el correspondiente algoritmo de minería de datos sobre el conjunto de todos los datos (como hemos visto anteriormente se obtienen resultados muy parecidos que seleccionando atributos) y con un método de evaluación cruzada con 10 pliegues.

Estación	Método	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error
Larrabide	**M5P**	0.9509	1.6108	2.0642	28.5692%	30.9586%
Montes de cierzo	**M5P**	0.94	1.626	2.0792	31.595%	34.104%
Gorramendi	**M5P**	0.909	2.0283	2.5895	40.140%	41.67 %
Larrabide	**LinearRegression**	0.9494	1.6248	2.0941	28.8181%	31.4077%
Montes de cierzo	**LinearRegression**	0.9395	1.631	2.0868	31.693%	34.228%
Gorramendi	**LinearRegression**	0.9093	2.0212	2.5851	40.000%	41.598%
Larrabide	**MultilayerPerceptron**	0.9554	1.5341	1.9704	27.2102%	29.552 %
Montes de cierzo	**MultilayerPerceptron**	0.9463	1.5452	1.9712	30.027 %	32.332%
Gorramendi	**MultilayerPerceptron**	0.9116	1.9941	2.5555	39.462%	41.123%

Como se observa al aplicar los algoritmos MP5,LinearRegresion y Multilayer Perceptron, la estación de Larrabide obtiene los mejores resultados con error absoluto medio entre 1.6 y 1.5 grados (error relativo entre un 28% y 27%) muy seguida de la estación de Montes de Cierzo con un error absoluto medio entre 1.6 y 1.5 grados (error relativo entre un 31% y 30%) y de la estación de Gorramendi la cual da peores resultados que las dos anteriores con un error absoluto entre 2 y 1.9 grados (error relativo de un 41%). Como se puede ver los mejores resultados se obtienen en la estación de Larrabide seguida de la estación de Montes de Cierzo y de la estación de Gorramedi en este orden.

En cuanto a la comparación de los distintos algoritmos de minería de datos, se observa que el MultilayerPerceptrón obtiene los mejores resultados en cada una de las estaciones seguida del árbol de regresión MP5 y por último las regresiones lineales.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A continuación voy a mostrar una tabla comparativa de los resultados de la evaluación de distintos algoritmos de minería de datos para la predicción de la temperatura mínima a 24h para cada una de las 3 estaciones. Para esta comparativa he elegido 3 métodos de minería de datos muy representativos los cuales son el árbol de regresión MP5, la regresión lineal y el perceptrón Multicapa. Hay que señalar que estos resultados se han obtenido al aplicar el correspondiente algoritmo de minería de datos sobre el conjunto de todos los datos (como hemos visto anteriormente se obtienen resultados muy parecidos que seleccionando) y con un método de evaluación cruzada con 10 pliegues.

Estación	Método	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error
Larrabide	**M5P**	0.9405	1.5412	1.9618	31.882%	33.955%
Montes de cierzo	**M5P**	0.94	1.626	2.0792	31.595%	34.104%
Gorramendi	**M5P**	0.9167	1.7141	2.2579	37.104%	39.945%
Larrabide	**LinearRegression**	0.9391	1.5635	1.9848	32.345%	34.353%
Montes de cierzo	**LinearRegression**	0.9395	1.631	2.0868	31.693%	34.228%
Gorramendi	**LinearRegression**	0.9136	1.7374	2.2975	37.610%	40.645%
Larrabide	**MultilayerPerceptron**	0.9458	1.4702	1.8753	30.413%	32.458%
Montes de cierzo	**MultilayerPerceptron**	0.9463	1.5452	1.9712	30.027 %	32.332%
Gorramendi	**MultilayerPerceptron**	0.918	1.6981	2.2438	36.758%	39.695 %

Como se observa al aplicar los algoritmos MP5,LinearRegresion y Multilayer Perceptron, la estación de Larrabide obtiene los mejores resultados con error absoluto medio entre 1.5 y 1.4 grados (error relativo entre un 30% y 31%) a la par (igual de precisos que la estación de la Larrabide) de la estación de Montes de Cierzo con un error absoluto medio entre 1.5 y 1.4 grados (error relativo entre un 31% y 30%) y de la estación de Gorramendi la cual da peores resultados que las dos anteriores con un error absoluto entre 1.7 y 1.6 grados (error relativo entre un 37% y 36%). Como se puede ver los mejores resultados se obtienen en la estación de Larrabide y en la estación de Montes de Cierzo.

En cuanto a la comparación de los distintos algoritmos de minería de datos, se observa que el Multilayer perceptrón obtiene los mejores resultados en cada una de las estaciones seguida del árbol de regresión MP5 y por último las regresiones lineales.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Comparativa de algoritmos de minería de datos para la predicción de la velocidad media del viento a 24h

A continuación voy a mostrar una tabla comparativa de los resultados de la evaluación de distintos algoritmos de minería de datos para la predicción velocidad media del viento a 24h para cada una de las 3 estaciones. Para esta comparativa he elegido 3 métodos de minería de datos muy representativos los cuales son el árbol de regresión MP5, la regresión lineal y el perceptrón Multicapa. Hay que señalar que estos resultados se han obtenido al aplicar el correspondiente algoritmo de minería de datos sobre el conjunto de todos los datos (como hemos visto anteriormente se obtienen resultados muy parecidos que seleccionando) y con un método de evaluación cruzada con 10 pliegues.

Estación	Método	Correlation coefficient	Mean absolute error	Mean squared error	Relative absolute error	Relative squared error
Larrabide	**M5P**	0.5346	0.7833	1.0308	80.9258 %	84.5119 %
Montes de cierzo	**M5P**	0.5367	1.2529	1.5487	84.6%	84.3%
Gorramendi	**M5P**	0.5819	2.3234	3.0137	79.525%	81.308%
Larrabide	**LinearRegression**	0.5095	0.804	1.0496	83.0884 %	86.0161 %
Montes de cierzo	**LinearRegression**	0.5316	1.258	1.5541	84.976%	84.674%
Gorramendi	**LinearRegression**	0.5808	2.3216	3.0166	79.462%	81.385%
Larrabide	**MultilayerPerceptron**	0.5215	0.7918	1.0422	81.7666 %	85.4098 %
Montes de cierzo	**MultilayerPerceptron**	0.536	1.2524	1.5515	84.599%	84.529%
Gorramendi	**MultilayerPerceptron**	0.5795	2.3146	3.0236	79.224%	81.574%

Como se observa al aplicar los algoritmos MP5,LinearRegresion y Multilayer Perceptron , la estación de Larrabide obtiene los mejores resultados con error absoluto medio entre 0.78 y 0.8 m/s (error relativo entre un 80% y 81%) seguida de la estación de Montes de Cierzo con un error absoluto medio entre 1.25 m/s (error relativo de un 84%) y de la estación de Gorramendi la cual da peores resultados que las dos anteriores con un error absoluto entre 2.32 y 2.31 grados (error relativo del 79%). Como se puede ver los mejores resultados se obtienen en la estación de Larrabide seguida de la estación de Montes de Cierzo y de la estación de Gorramedi en este orden.

En cuanto a la comparación de los distintos algoritmos de minería de datos, se observa que los tres algoritmos Multilayer perceptrón, el árbol de regresión MP5 y la regresiones lineal obtienen unos resultados más o menos igual de precisos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A continuación voy a mostrar una tabla comparativa de los resultados de la evaluación de distintos algoritmos de minería de datos para la predicción de la precipitación acumulada a 24h para cada una de las 3 estaciones. Para esta comparativa he elegido 3 métodos de minería de datos muy representativos los cuales son el árbol de clasificación NBtree, la regresión logística y el perceptrón Multicapa. Hay que señalar que estos resultados se han obtenido al aplicar el correspondiente algoritmo de minería de datos sobre el conjunto de todos los datos (como hemos visto anteriormente se obtienen resultados muy parecidos que seleccionando) y con un método de evaluación cruzada con 10 pliegues.

Estación	Método	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Mean squared error
Larrabide	**NBTree**	2302 (70.0122%)	1000 (30.4136%)	0.2638	0.3789
Montes de cierzo	**NBTree**	2318 (71.8092 %)	910 (28.1908%)	0.2447	0.3625
Gorramendi	**NBTree**	1483 (57.975 %)	1075 (42.025 %)	0.3478	0.4257
Larrabide	**Logistic**	2344 (71.289%)	944 (28.7105%)	0.268	0.367
Montes de cierzo	**Logistic**	2368 (73.3581 %)	860 (26.6419%)	0.2402	0.3483
Gorramendi	**Logistic**	1451 (56.724 %)	1107 (43.276 %)	0.3541	0.4224
Larrabide	**MultilayerPerceptron**	2347 (71.686 %)	927 (28.314 %)	0.262	0.3651
Montes de cierzo	**MultilayerPerceptron**	2380 (73.7299 %)	848 (26.2701%)	0.2403	0.351
Gorramendi	**MultilayerPerceptron**	1467 (57.3495%)	1091 (42.650%)	0.3539	0.4257

Como se observa al aplicar los algoritmos MP5,LinearRegresion y Multilayer Perceptron, la estación de Montes de Cierzo obtiene los mejores resultados clasificando correctamente entre un 71% y un 73% de las instancias (error absoluto medio de 0.24) seguida de la estación de Larrabide la cual ha clasificado correctamente entre un 71% y un 73% de las instancias (error absoluto medio de 0.24) y de la estación de Gorramendi la cual da peores resultados que las dos anteriores clasificando correctamente un 57% de las instancias (error absoluto medio de 0.34) . Como se puede ver los mejores resultados se obtienen en la estación de Montes de Cierzo seguida de la estación de Larrabide y de la estación de Gorramedi en este orden.

En cuanto a la comparación de los distintos algoritmos de minería de datos, se observa que el Multilayer perceptrón obtiene los mejores resultados en cada una de las estaciones seguida de la regresión logística y por último el árbol de decisión NBTree.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Conclusiones de la comparativa:

Tras la realización de la comparativa puedo llegar a las siguientes conclusiones:

- A la hora de predecir las temperaturas (máxima, media y mínima) a 24h se pueden concluir que la estación de Larrabide y de Montes de Cierzo dan resultados muy parecidos y casi iguales en la predicción de las 3 temperaturas. Además estas dos estaciones obtienen los mejores resultados frente a la Estación de Gorramendi que da bastantes peores resultados. En mi opinión estos peores resultados de la estación de Gorramendi son debidos a las características de este clima y al hecho de que esta estación solo posee el conjunto básico (común) de atributos meteorológicos mientras que las estaciones de Larrabide y Montes de Cierzo tienen otros atributos meteorológicos (presión atmosférica, temperatura media del suelo a 50m y temperatura media a 150m) que mejoran la predicción de las temperaturas. Además tres de estos atributos presión atmosférica (único en la estación de Larrabide), temperatura media del suelo a 50m y temperatura media a 150m (Únicos en las estación de Montes de Cierzo) como se vio en la selección de atributos son bastantes de influyentes a la hora de predecir las temperaturas.

- A la hora de predecir la velocidad media del viento a 24h se puede concluir que la estación de Larrabide es la que obtiene los mejores resultados seguida de lejos la estación de Montes de Cierzo y por último la estación de Gorramendi. Estos resultados notablemente mejores de la estación de Larrabide son debido a parte de las características del clima al hecho de la presencia del atributo presión atmosférica (Exclusivo de esta estación) y que además como hemos visto en la comparativa de selección de atributos es el segundo atributo más influyente en la predicción de la velocidad del viento a 24h.

- A la hora de predecir la precipitación acumulada a 24h se puede concluir que la estación de Montes de Cierzo es la que obtiene los mejores resultados seguida de cerca de la estación de Larrabide y por último la estación de Gorramendi. Esta diferencia de resultados es principalmente debido a las características del clima. Por ejemplo en la estación de Montes de Cierzo el 70% de las instancias pertenece a la clase 1 (no llueve) por lo que a la hora de entrenar y evaluar lo más probable es que pertenezcan a la clase 1 y de esta forma facilitando la predicción. Sin embargo en la estación de Gorramendi el 45% de la instancias pertenece a la clase 2 (hay poca precipitación acumulada), el 44% pertenecen a la de la clase 0 (no hay precipitación acumulada) y el resto a la clase 3 con lo que la distribución de la precipitación acumulada esta mas distribuida y por tanto es más difícil de generalizar o predecir que va suceder.

- A la hora de predecir las temperaturas (máxima, media, mínima) a 24h y velocidad media del viento a 24h se ha concluido que el perceptrón multicapa (MultilayerPerceptron) es el algoritmo de minería de datos que mejores resultados ofrece seguido del árbol de regresión MP5 y de las regresión lineal. En cuanto la predicción (clasificación) de la precipitación acumulada a 24h el perceptrón multicapa (MultilayerPerceptron) es también el algoritmo de minería de datos que mejor resultados nos ofrece seguido de la regresión logística y del árbol de decisión NBTree.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Aplicación:

Para completar el despliegue del sistema de minería de datos, he desarrollado una aplicación de escritorio en el lenguaje de programación java la cual predice la temperatura (máxima, mínima y media) a 24h, velocidad media del viento a 24h y la precipitación acumulada a 24h. Para poder ejecutar esta aplicación es necesario disponer de un ordenador con la plataforma Java ya Instalada y ejecutar el ejecutable (.jar) que se encuentra dentro de la carpeta aplicación que viene adjunta a este proyecto.

Para poder desarrollar esta aplicación he utilizado la librería java (archivo .jar) que ofrece Weka, esta librería proporciona todo el conjunto de clases y métodos para la creación y evaluación de los modelos de minería de datos.

A continuación en el siguiente apartado voy a mostrar más detalladamente la especificación de esta aplicación.

Especificación de la aplicación:

Como he comentado anteriormente esta aplicación es el despliegue del sistema de minería de datos, por lo que se puede identificar en ella las distintas fases de la minería de datos. Al iniciar esta aplicación en primer lugar hay que entrenar los modelos de minería de datos por lo que hay que pulsar en el botón llamado "**Crear modelos**". Cuando es pulsado este botón la aplicación carga los datos de la estación meteorológica correspondiente, se realiza el preprocesamiento (filtro replacemissingValues) y trasformación de los datos y se crean (generan y entrenan) los mejores algoritmos de minería de datos para la predicción de cada atributo meteorológico objetivo y los gurdan en el disco duro. Hay que comentar que este proceso puede tardar varios minutos ya que la creación y entrenamiento de los modelos (sobre todo el perceptrón multicapa) tardan bastante entrenar.

A continuación una vez creados los modelos ya se podrá predecir los atributos objetivos. Para ello se introducirán en las entradas de la interfaz los datos actuales de los atributos meteorológicos y a continuación pulsar el botón predecir**.** Tras pulsar este botón aparecerán los resultados de la predicción de los distintos algoritmos de minería de datos para la predicción de las temperaturas (media, máxima y mínima) a 24h, la velocidad media del viento a 24h y la precipitación acumulada a 24h.

```
 // train MULTILAYERPERCEPTRON
MultilayerPerceptron cl = new MultilayerPerceptron();
cl.setOptions(weka.core.Utils.splitOptions("-L 0.01 -M 0.01 -N 2000 -V 0 -S 0 -E 20 -H \"15\""));
 // further options...
cl.buildClassifier(Datatemmed);
// save model + header
Vector v = new Vector();
v.add(cl);
v.add(new Instances(Datatemmed, 0));
SerializationHelper.write(dir + "/datos/" + "EstacionGorramendi" + "/" + "modelos/tempmed.save", v);
```

Figura 12: Ejemplo de generación de un modelo MultilayerPerceptron.

Por último voy a mostrar una serie de pruebas realizadas con la aplicación:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Pruebas de la aplicación:

A continuación voy a mostrar una serie de pruebas que he realizado a la aplicación para comprobar la precisión y exactitud de los resultados. Estas pruebas consistirán en introducir un ejemplo de los datos de entrenamiento y comprobar la exactitud de los valores predichos con los valores de salida (valores reales) del ejemplo de entrenamiento.

Prueba 1:

Para el caso del día 25 de noviembre del 2007:

Atributo	Valor real	Valor Predicho
Temperatura media a 24h	0.8 grados	-0.19 grados
Temperatura máxima a 24h	2.3 grados	3.43 grados
Temperatura mínima a 24h	-0.4 grados	-2.95 grados
Precipitación acumulada a 24h	1: hay precipitación acumulada entre 0 y 15 l/m^2	1 : hay precipitación acumulada entre 0 y 15 l/m^2
Velocidad media del viento a 24h	12.2 m/s	8.010 m/s

Como se observa los resultados no son exactos pero siempre están entre un grado más o un grado menos.

Prueba 2:

Para el caso del día 8 de Junio del 2006:

Atributo	Valor real	Valor Predicho
Temperatura media a 24h	21.8 grados	21.128 grados
Temperatura máxima a 24h	27.9 grados	25.8 grados
Temperatura mínima a 24h	17.5 grados	16.225 grados
Precipitación acumulada a 24h	0: No hay precipitación acumulada	0: No hay precipitación acumulada
Velocidad media del viento a 24h	11.1 m/s	6.63 m/s

Como se observa en el cálculo de las temperaturas medias y mínimas es el que menos error se comete.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Análisis de Resultados.

Tras la realización de este proyecto he obtenido unos resultados beneficiosos ya que he desarrollado un sistema de minería de datos que a través de su despliegue mediante una aplicación java de escritorio es capaz de predecir las temperaturas a 24h, velocidad media del viento a 24h y la precipitación acumulada a 24h. Por tanto con este sistema voy a obtener nuevo conocimiento para utilizarlo de forma más apropiada.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Estudio Económico

No tiene sentido un análisis económico de este proyecto ya que ha sido desarrollado sin ningún propósito comercial. Además este proyecto solo tiene un carácter educativo y no comercial.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Conclusiones.

Tras la realización de este proyecto he aprendido que es el proceso de descubrimiento del conocimiento KDD y las fases que este tiene. Además siguiendo la metodología CRISP-DM para la implantación de sistemas de minería de datos, he desarrollado un sistema de minería de datos para la predicción de las temperaturas (mínima, media y máxima) a 24h, la velocidad media del vienta del viento a 24h y la precipitación acumulada a 24h.

Por tanto tras la realización del proyecto he cumplido todos los principales objetivos:

- Comprender que es el proceso de KDD con todas sus fases y aplicaciones.
- Entender las diferentes tareas de minería de datos (predictivas y descriptivas) así como los correspondientes métodos (regresiones, redes neuronales etc...) para la resolución de estas tareas.
- Aplicar estándar CRISP-DM para desarrollar un sistema de minería de datos.
- Desarrollar un sistema de minería de datos para predecir la temperatura (máxima, media, mínima), velocidad media del viento y precipitación (discreteada) a las 24 horas.
- Comparar y validar los diferentes métodos y algoritmos de minería de datos para establecer los mejores.
- Comparar los resultados obtenidos para cada estación meteorológica y sacar conclusiones de ello.
- Realizar el despliegue del sistema de minería de datos mediante una pequeña aplicación grafica en java.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Manual de usuario.

En este apartado voy a mostrar un manual de como se debe utilizar la aplicación que he desarrollado.

En primer lugar para arrancar la aplicación hay que ejecutar (doble clic) el archivo **ejecutable.jar** que se encuentra en la carpeta aplicación que se encuentra dentro de la carpeta del proyecto. Una vez ejecutada por primera vez la aplicación se mostrara un cuadro de dialogo don te indicara que selecciones la carpeta principal del proyecto llamada "proyecto". Esto se debe a que esta aplicación debe saber donde está esta carpeta para poder acceder a los datos de entrenamiento.

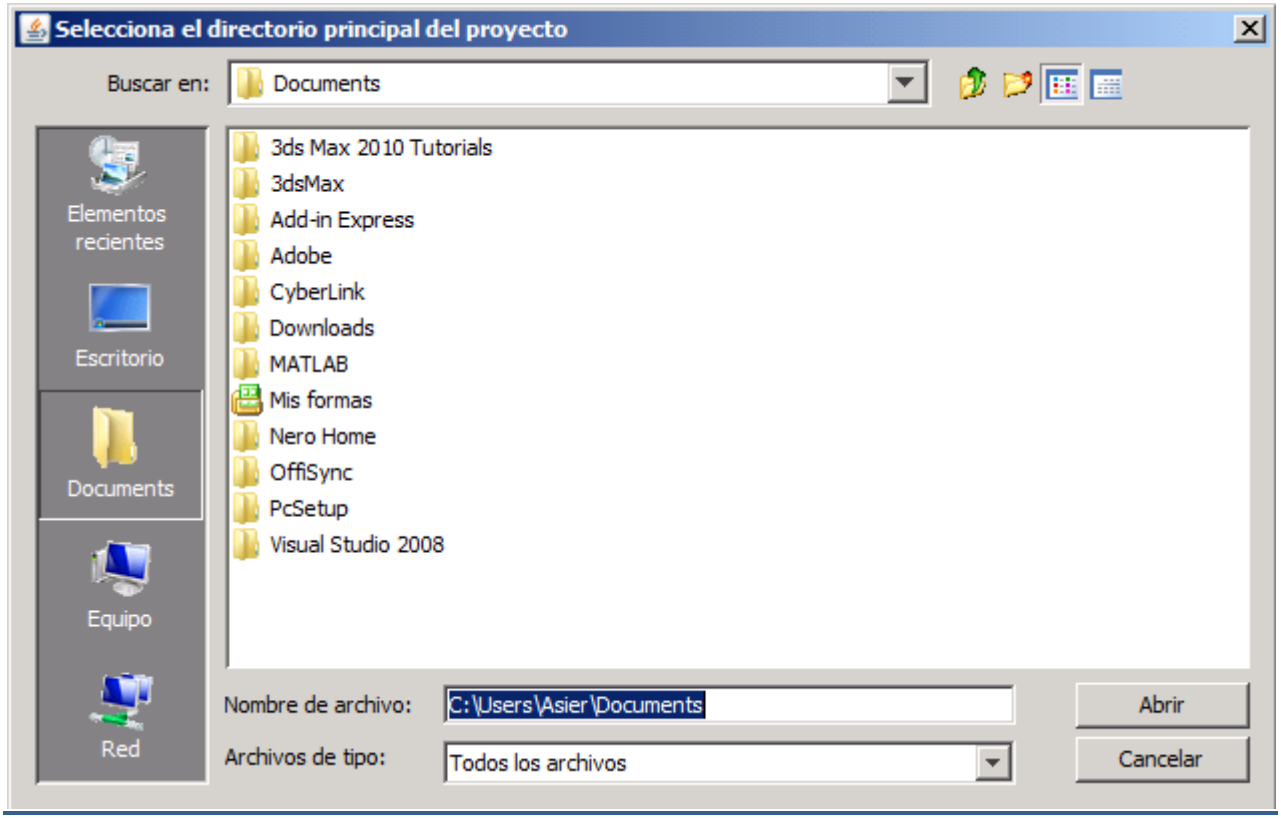


Figura 13: Selección del directorio principal del proyecto.

Tras seleccionar el directorio principal del proyecto las siguientes veces que ejecutes la aplicación este directorio será leído de disco y no saldrá esta ventana de selección del directorio principal. Una vez ya seleccionado el directorio aparecerá la interfaz principal del proyecto. La cual consta de tres pestañas (paneles) cada una de ellas con el nombre de una de las tres (Estación Larrabide, Estación Gorramendi y Estación Montes de Cierzo) estaciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

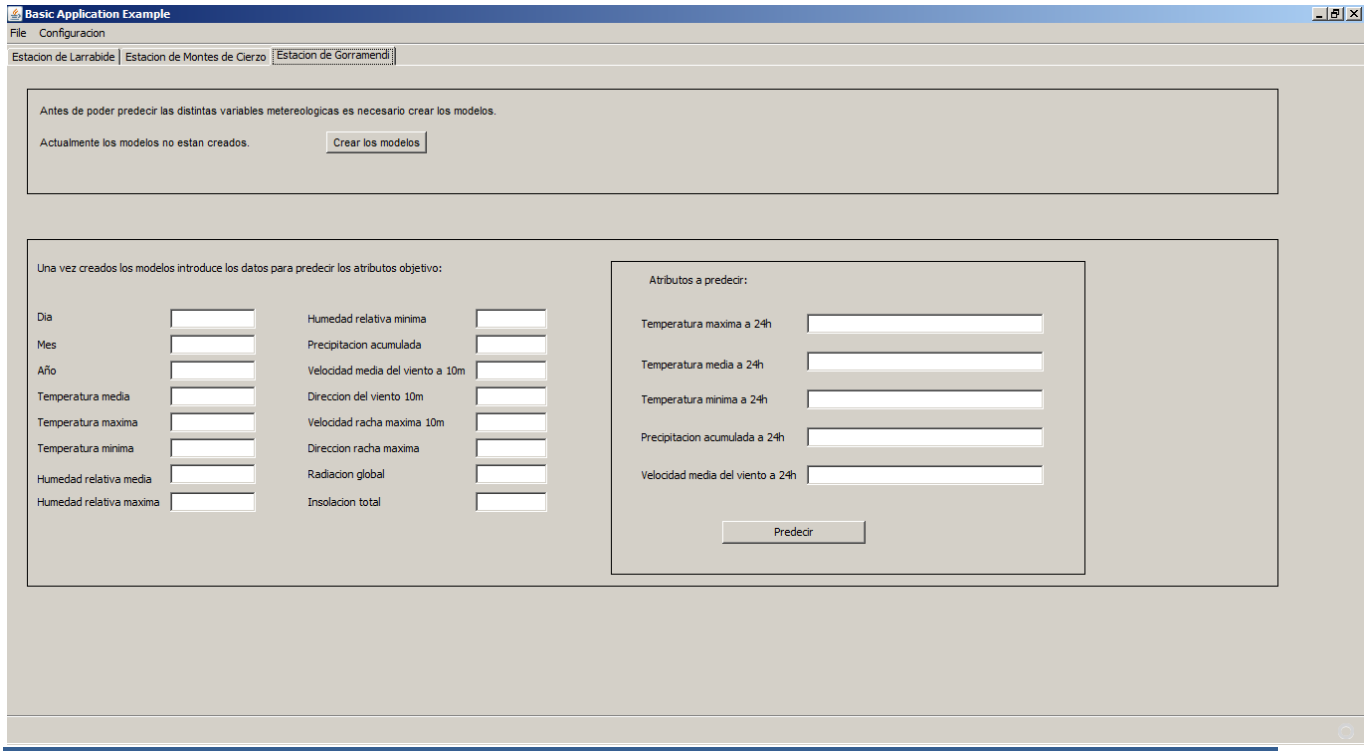


Figura 14: Interfaz principal de la aplicación:

En cada panel de las pestañas en la parte de arriba hay otro panel en el cual te indica que antes de predecir las distintas variables meteorológicas es necesario crear y entrenar los distintos modelos de minería de datos. Por tanto habrá que pulsar el botón llamado "crear los modelos" para generar los modelos de minería de datos para predecir las distintas variables meteorológicas.

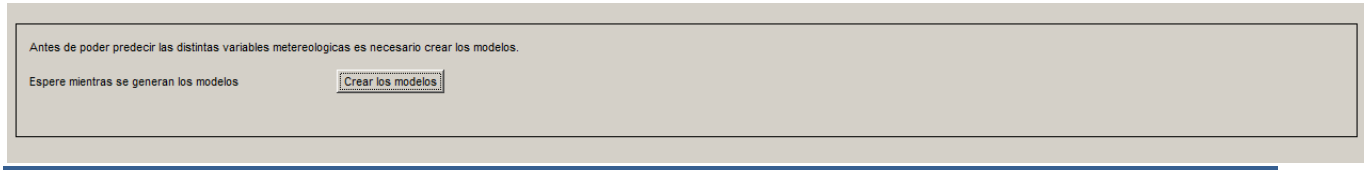


Figura 15: Indicación de que debe esperar mientras se generan los modelos.

Como se indica en el texto inferior de este panel es necesario esperar a que se terminen de generar los modelos lo cual puede tardar varios minutos una vez finalizado el proceso este mismo texto indicara que los modelos ya están creados. Una vez ya creados los modelos habrá que introducir los datos en los campos del panel inferior de la aplicación. En estos campos se introducirán los datos meteorológicos actuales (hoy) y una vez introducidos habrá que pulsar el botón predecir para mostrar los resultados obtenidos en sus correspondientes campos de texto.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

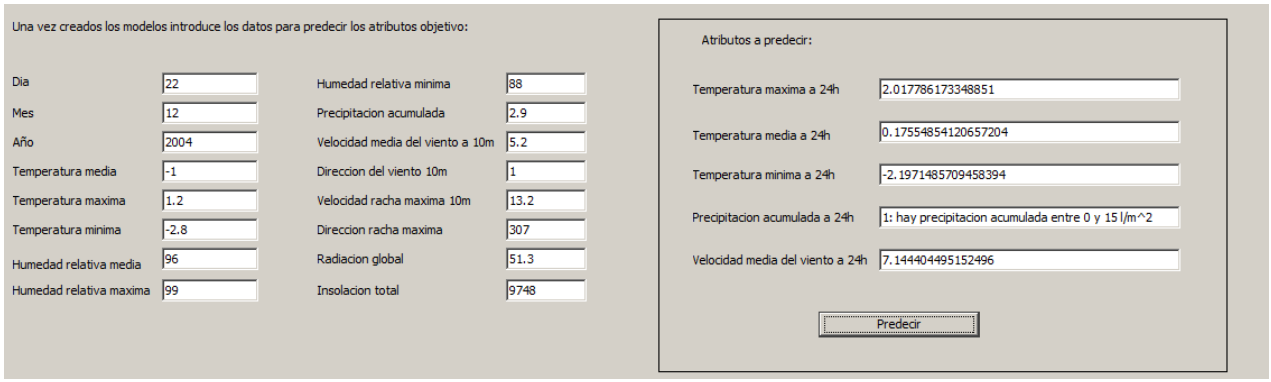


Figura 16: Un ejemplo de predicción:

En este ejemplo se observa se observa los resultados de la predicción de la temperaturas (media, máxima y mínima) a 24h, la velocidad del viento a 24h y la precipitación acumulada a 24h.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Bibliografía.

A continuación voy a mostrar las principales fuentes bibliográficas de las que he consultado para la realización del proyecto:

[1] Libro acerca de la Minería de datos, Introducción a la Minería de datos, José Hernández Orallo – María José Ramírez Quintana – Cesar Ferri Ramírez [2004]

[2] Libro acerca de la Minería de datos, Data Mining Concepts, models methods, and algorithms, Mehmed Kantardzic [2000]

[3] Página Oficial de la herramienta WEKA, http://www.cs.waikato.ac.nz/ml/weka/ [2010]

[4] Página web con información de cómo utilizar las librerías de WEKA desde código Java, http://weka.wikispaces.com/Use+Weka+in+your+Java+code#Links [2010]

[5] Página web con información del API de WEKA para utilizarlo sobre el código Java, http://weka.sourceforge.net/doc.stable/ [2010]

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira