



Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA AGRONÓMICA Y
BIOCIENCIAS**

**NEKAZARITZAKO INGENIARITZAKO ETA BIOZIENTZIETAKO GOI MAILAKO
ESKOLA TEKNIKOA**

*Uso del aprendizaje automático para la localización de enclaves arqueológicos a partir de datos
LiDAR*

presentado por

ALEXANDRE DURÓ CAZORLA

aurkeztua

MASTER UNIVERSITARIO EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA Y TELEDETECCIÓN
UNIBERTSITATE MASTERRA INFORMAZIO GEOGRAFIKOKO SISTEMETAN ETA TELEDETEKZIOAN



SEPTIEMBRE, 2021

Resumen

La identificación de yacimientos arqueológicos es una de las tareas principales en la gestión del patrimonio cultural. Navarra cuenta con un sistema de información geográfica para la gestión de su inventario arqueológico. Este inventario se alimenta de los resultados de prospecciones sistemáticas del territorio. En la última década se ha demostrado que el uso de las técnicas de teledetección por sensores remotos es muy eficaz para la identificación y caracterización de entornos arqueológicos, siendo los productos derivados de los sensores LiDAR una de las principales herramientas empleadas con este fin. En este trabajo se combinan diferentes productos derivados de LiDAR para entrenar, mediante clasificadores basados en árboles de decisión, un conjunto de ventanas de observación obtenidas de los diferentes rangos de superficie de los yacimientos inventariados en la Comunidad Foral. Los modelos obtenidos se utilizan para predecir la probabilidad que las anomalías topográficas detectadas por los sensores LiDAR indiquen o no la presencia de enclaves arqueológicos.

Palabras clave:

LiDAR, Random Forest, yacimiento arqueológico, MDT, Multi-scale Topographic Position, Local Dominance, técnicas de visualización

Abstract

The identification of archaeological sites is one of the main tasks in the management of cultural heritage. Navarra has a geographic information system for managing its archaeological inventory. This inventory is fed by the results of systematic surveys of the territory. In the last decade, it has been shown that the use of remote sensing techniques is very effective for the identification and characterization of archaeological environments, being the products derived from LiDAR sensors one of the main tools used for this purpose. In this work, different products derived from LiDAR are combined to train, by means of classifiers based on decision trees, a set of observation windows obtained from the different surface ranges of the inventoried deposits in the Autonomous Community. The models obtained are used to predict the probability that the topographic anomalies detected by the LiDAR sensors indicate or not the presence of archaeological sites.

Keywords:

LiDAR, Random Forest, archaeological site, MDT, Multi-scale Topographic Position, Local Dominance, visualization techniques

Agradecimientos

A Jesús Sesma y Jesús García Gazólaz y a la Sección de Registro, Bienes Muebles y Arqueología de Gobierno de Navarra por su apoyo y acceso al Inventario Arqueológico de Navarra y a mis compañeros de profesión María Rosario Mateo, Javier Nuin, Javier Armendáriz por sus consejos y paciencia durante esta aventura formativa.

También a mis tutores Jesús Álvarez y José Antonio Sanz por animarse a guiar a un arqueólogo por los entresijos del aprendizaje automático.

Así como a mi familia por su apoyo y comprensión durante estos años.

Índice

Resumen	i
Abstract.....	i
Agradecimientos	ii
Índice.....	iii
Índice de figuras y tablas.....	iv
Índice de tablas.....	viii
Índice de ecuaciones	ix
1.- Introducción	1
1.1.- El Inventario Arqueológico de Navarra.....	1
1.2.- El reconocimiento del territorio en arqueología.....	5
1.3.- El uso de sensores remotos en arqueología.....	5
1.4.- Hacia la detección automática de los entornos arqueológicos.....	10
1.5.- Objetivos	15
2.- Materiales y métodos.....	17
2.1.- Área de estudio.....	17
2.2.- Creación del <i>dataset</i>	18
2.2.1.- Elaboración de los productos derivados de LiDAR	18
2.2.2.- Selección de yacimientos arqueológicos del IAN.	23
2.2.3.- Preprocesamiento.....	28
2.2.4.- Montaje del <i>dataset</i>	34
2.3.- Proceso de aprendizaje	35
2.3.1.- Modelos.....	36
2.3.2.- Rendimiento	39
2.4.- Ventana de análisis	41
3.- Resultados y discusión	43
3.1.- Proceso de entrenamiento y evaluación de los modelos.....	45
3.2 Comparativa entre modelos.....	47
3.3 Predicción de nuevos datos.....	53
5.- Conclusiones	60
6.- Lista de referencias	61
ANEXOS.....	67
Anexo I Cuadernos de código Phyton (Notebooks).....	67
Anexo II. Fichas de resultados del entrenamiento de los modelos	100
Anexo III. Estadísticas descriptivas de las ventanas de observación	144

Índice de figuras y tablas

Figura 1. Mapa de densidad media de yacimientos arqueológicos por municipio registrados en el Inventario Arqueológico de Navarra. Elaboración propia.....	3
Figura 2. Mapa conceptual para relacionar las tipologías y las clases de yacimientos arqueológicos que figuran en el Inventario Arqueológico de Navarra. Elaboración propia.....	4
Figura 3. Cantidad de yacimientos arqueológicos inventariados por tipología, se observa una clara desproporcionalidad hacia los entornos arqueológico de tipo "Lugar de habitación". Elaboración propia.	4
Figura 4. Prospecciones de tipo geofísico (GPR y magnetómetro) realizadas por SOT Prospección Arqueológica dentro de un enclave arqueológico bien delimitado. Proyecto de investigación arqueológica de La Custodia (Viana) bajo la dirección científica de Javier Armendáriz Martija. Elaboración propia.....	6
Figura 5. Relación de publicaciones científicas localizadas en Web of Science bajo el epígrafe "Remote Sensing and Archaeology". La tendencia al alza en los últimos veinte años se relaciona directamente con la mejora en los sistemas de captación de los sensores, en los tiempos de computación y en el propio desarrollo de la sociedad de la información. Fuente [6].....	6
Figura 6. Variación de la reflectancia en marcas de cultivo y de suelo respectivamente. Fuente [11].....	7
Figura 7. Esquema del funcionamiento de LiDAR. Fuente [18].....	8
Figura 8. Ejemplo del uso combinado de diferentes sistemas de visualización producidos a partir del modelo digital de elevación del vuelo LiDAR de 2012 y la información espectral (RGBI) obtenida durante el vuelo. Fuente [40].....	11
Figura 9. El aprendizaje automático forma parte del proceso de descubrimiento a partir de bases de datos. Las principales técnicas del aprendizaje automático corresponden a las que se ilustran en esta figura realizada por Natalia Acevedo. Fuente [56]	12
Figura 10. Localización de la ventana de análisis dentro del ambito territorial de la Comunidad Foral. Fuente: Servicios wms de IDENA: REFERE_Lay_mapabase y IDENA: REFERE_Lay_baseorto Elaboración propia.....	17
Figura 11. Diagrama de flujo de la metodología para obtener los productos derivados de LiDAR y sus modos de visualización. Elaboración propia.....	19
Figura 12. Composición de cuatro imágenes de yacimientos que presentan características topográficas que los definen como de microescala (arriba izquierda), mesoescala (arriba derecha); macroescala (abajo izquierda) y fuera de rango (abajo derecha). Elaboración propia.	21
Figura 13. Representación gráfica de anomalías topográficas de origen arqueológico que quedan resaltadas mediante el uso de las técnicas de visualización descritas en [22]	22
Figura 14. Diagrama conceptual de la base de datos del inventario arqueológico de navarra. Elaboración propia.	25
Figura 15. Diagrama conceptual de la base de datos del inventario arqueológico de navarra. Elaboración propia.	26
Figura 16. Diagrama conceptual de la base de datos del inventario arqueológico de navarra. Elaboración propia.	26

Figura 17. Diagrama conceptual de la base de datos del inventario arqueológico de Navarra. Elaboración propia.	27
Figura 18. Diagrama de flujo del proceso de adquisición de datos del Inventario Arqueológico de Navarra.	28
Figura 19. Diagrama de flujo del preprocesamiento de los datos y montaje del dataset.	30
Figura 20. Representación de cuatro yacimientos arqueológicos del municipio de Beire (Navarra) sobre una imagen de multiescala realizada con información de dominancia local. Se observa como los yacimientos de San Julián y El Cerco son entornos topográficamente separables, mientras que los yacimientos de Cardete II y Cardete III ocupan zona llanas difícilmente separables por criterios topográficos. Elaboración propia.	31
Figura 21. Estructura de un mapa de características de una red SOM en las neuronas adyacentes a la capa de salida están conectadas entre sí por una relación de vecindad. Fuente [72].....	32
Figura 22. Ejemplo del yacimiento El Cerco (Beire) que corresponde a uno de los enclaves de la ventana G38. En rojo la delimitación del IAN, en blanco la dimensión de la ventana y la representación de los píxeles asociados a la ventana. Se observa como las ventanas regulares recogen mejor la forma de los enclaves y de la información estadística que se recoge de cada píxel.	34
Figura 23. Esquema de funcionamiento de un árbol de decisión aplicado a los datos de la ventana de observación G6. Obsérvese como a partir de los valores dados separa los datos entre clase positiva (yacimiento) o negativa (valor aleatorio). Fuente: Elaboración propia a partir de la herramienta Tree Viewer de Orange Data Mining 3.29.....	37
Figura 24. Representación gráfica del funcionamiento de los diferentes métodos basados en los árboles de decisión. Fuente [78].....	38
Figura 25. Modelo de ficha en el que se observan los datos de la ventana G6, que ha resultado ser un excelente clasificador. Elaboración propia.....	44
Figura 26. Gráfica de barras que relaciona la frecuencia que ha sido utilizada cada una de las variables condicionales y cuál ha sido el estadístico más relevante en cada caso. Elaboración propia.	45
Figura 27. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score. La escala de colores de la leyenda corresponde a los valores de F1 score. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.	47
Figura 28. Gráfica de dispersión entre la medida de evaluación Media Geométrica y los valores de dispersión de tipologías de yacimientos usados en cada ventana. En rojo seleccionados los buenos clasificadores y en azul los que aportan un resultado más pobres. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.....	48
Figura 29. Gráfica de dispersión entre la medida de evaluación F1 score y los valores de dispersión de tipologías de yacimientos usados en cada ventana. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.	49
Figura 30. Gráfica de dispersión entre la medida de evaluación ROC AUC y los valores de dispersión de tipologías de yacimientos usados en cada ventana. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.	49

Figura 31. Gráfica de dispersión entre la medida de evaluación AUC PR y los valores de dispersión de tipologías de yacimientos usados en cada ventana. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.	50
Figura 32. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score para los datos de la ventana G8 aplicados al resto de ventanas diseñadas. Resalta con valores superiores a 80 la ventana G5. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.	50
Figura 33. Gráfica de dispersión entre las medida de evaluación Media Geométrica y F1 score para los datos de la ventana G25 aplicados al resto de ventanas diseñadas. Resalta con valores superiores a 80 las ventanas G23, G29 y G35. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.....	51
Figura 34. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score para los datos de la ventana G40 aplicados al resto de ventanas diseñadas. Resalta con valores superiores a 80 las ventanas G29, G34 y G38. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.....	51
Figura 35. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score para los datos de la ventana G15 aplicados al resto de ventanas diseñadas. Resalta con valores inferiores a 80 la propia ventana G15. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.....	52
Figura 36. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre el sombreado analítico disponible en IDENA (ELEVAC_Ras_RelieveBN_MDT_50CM_VE2017). Elaboración propia.....	53
Figura 37. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre el sombreado orográfico basado en iluminación anisotrópica disponible en IDENA (ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017). Elaboración propia.	54
Figura 38. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre la imagen multiescalar elaborada a partir de la composición RGB de los valores de DEVmax. Elaboración propia.....	54
Figura 39. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre la imagen multiescalar elaborada a partir de la composición RGB de los valores de DEVmax, fundida con los valores de Local Dominance. Elaboración propia.	55
Figura 40. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G6 (rango 457 - 520 m ²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.	55
Figura 41. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G8 (rango 237 - 318 m ²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.	56
Figura 42. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G12 (rango 675 -794 m ²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.	56
Figura 43. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G13 (rango 4.724 -5.118 m ²); los valores se solapan sobre el	

servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia. 57

Figura 44. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G20 (rango 6.603 -7.550 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia. 57

Figura 45. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G25 (rango 8.537 -9.725 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia. 58

Figura 46. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G40 (rango 37.202 -41.177 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia. 58

Índice de tablas

Tabla 1. Investigaciones de los últimos 10 años realizadas para detectar automáticamente entornos arqueológicos. Fuente [49].....	15
Tabla 2 Relación de los diferentes modos de visualización creados a partir de los cuales se han extraído valores estadísticos para entrenar los modelos.....	23
Tabla 3. Relación del número de ventanas diferenciadas y el rango en metros cuadrados que representan.	32
Tabla 4. Tabla interpretativa para comprender el funcionamiento de las métricas de evaluación de un clasificador binario. Fuente [54].....	40
Tabla 5. Relación de las métricas de evaluación de los diferentes modelos entrenados. Elaboración propia.	46

Índice de ecuaciones

Ecuación 2.1.....	19
Ecuación 2.2.....	22
Ecuación 2.3.....	31
Ecuación 2.4.....	32
Ecuación 2.5.....	39
Ecuación 2.6.....	39
Ecuación 2.7.....	40
Ecuación 2.8.....	40
Ecuación 2.9.....	40
Ecuación 2.10.....	40
Ecuación 2.11.....	40
Ecuación 2.12.....	41

1.- Introducción

Gobierno de Navarra implementó en 2013 un sistema de información geográfica para la gestión del Inventario Arqueológico de Navarra (SIGIAN). Este inventario constituye una excelente base de datos para la gestión del patrimonio arqueológico de nuestra Comunidad.

Este **trabajo final de máster** se centra en buscar, seleccionar y aplicar estrategias de **minería de datos para la detección de entornos arqueológicos**. El objetivo es crear nuevo conocimiento a partir de la información registrada en el citado inventario, fruto de las prospecciones que se realizan sistemáticamente en Navarra desde la última década del pasado siglo.

1.1.- El Inventario Arqueológico de Navarra

La Ley Foral 14/2005, de 22 de noviembre, del Patrimonio Cultural de Navarra define el patrimonio arqueológico como un bien de dominio público. En consecuencia, es obligación del departamento competente formar y mantener actualizado el Inventario Arqueológico de Navarra (IAN), ya que este debe ser tenido en cuenta en la elaboración de los instrumentos de ordenación territorial, planeamientos urbanísticos, estudios de evaluación e impacto ambiental [...][1].

Desde 2013, la gestión y mantenimiento de este inventario se realiza mediante un visor cartográfico que alberga un sistema de información geográfica *on line* de registro supervisado (SIGIAN). Fue creado por TRACASA y, actualmente, está en uso la versión 2020.01¹. Esta herramienta consigue reducir el tiempo de gestión de expedientes, mejora de la calidad de los informes al integrar catalogación y cartografía, además permite editar vectorialmente los yacimientos arqueológicos e intersectar otros archivos vectoriales con el objeto de prevenir afecciones.

Una de las claves del éxito de SIGIAN se sustenta en la base de datos que subyace en él, el Inventario Arqueológico de Navarra (IAN). Desde la Sección de Registro, Bienes Muebles y Arqueología anualmente se licitan varios lotes de prospección a los que se presentan las empresas especializadas en prospecciones arqueológicas que, como mínimo, deben plantear los siguientes criterios metodológicos [2]:

1. *Investigación bibliográfica exhaustiva sobre todos los datos de tipo arqueológico conocidos en las zonas y términos municipales objeto de la contratación.*
2. *Investigación de la documentación administrativa, cartográfica, fotográfica, toponímica, catastral y geomorfológica existente.*
3. *Encuesta oral entre al menos 2 personas de cada término municipal con conocimiento y/o formación en historia de la localidad o hallazgos arqueológicos.*
4. *Revisión del terreno mediante prospección arqueológica intensiva, cuyo grado y metodología serán fijados en la propuesta.*

¹ Esta versión incluye un cambio de arquitectura para fusionar en una única aplicación y desplegar enteramente en Gobierno de Navarra.

5. *Se valorará la utilización de sistemas de prospección a partir de datos LiDAR en las zonas con cobertura vegetal extensa. La Sección de Registro, Bienes Muebles no facilitará el soporte técnico para poder realizar este trabajo.*
6. *Se recogerán los materiales hallados en superficie y se realizará la toma de datos y documentación para la catalogación del hallazgo.*
7. *Inspección de los lugares ya catalogados situados en este tipo de terrenos.*
8. *[...]*

Se trata de una metodología que pretende normalizar el conocimiento arqueológico sobre el territorio de Navarra, a través de la prospección pedestre y del análisis del conocimiento previo, ya sea de fuentes orales, escritas, gráficas, fotográficas, planimétricas, toponímicas y/o cartográficas.

Los resultados de estas prospecciones están volcados en el IAN a través del SIGIAN con su correspondiente ficha de inventario. Estas fichas son el reflejo gráfico de una base de datos relacional que asocia a un código un conjunto de atributos que permiten identificar, localizar, describir y valorar todos y cada uno de los enclaves arqueológicos de Navarra.

Actualmente el inventario cuenta con 9.481 registros, de los cuales 7.773 son considerados yacimientos arqueológicos con su correspondiente grado de protección, mientras que 1.708 se registran como hallazgos aislados.

Atendiendo a la densidad media de yacimientos inventariados (Figura 1) podemos ver como Navarra presenta grandes desigualdades en su territorio. Esto se ve motivado seguramente en el empleo preferente de la técnica de la prospección pedestre para el reconocimiento del territorio. Esta técnica presenta dos condicionantes claves, por un lado, todo aquello que tiene que ver con la visibilidad y la accesibilidad del terreno, así como los procesos geomorfológicos que hayan podido alterar el yacimiento; y, por otro lado, los condicionantes relacionados con la formación del registro arqueológico. Que como se explica a continuación son inherentes al objeto de estudio.

El IAN, al igual que otros inventarios arqueológicos, está sujeto a unos condicionantes que pueden determinar su uso más allá de la gestión administrativa del patrimonio. Por tanto, la primera pregunta que se debe plantear corresponde a si nuestra fuente de datos de partida es fiable, estando su respuesta en el procedimiento por el cual se identifica en el territorio el concepto de "yacimiento arqueológico".

Partiendo de que la materialidad de la evidencia es lo que define a la Arqueología como disciplina científica, se puede definir un yacimiento arqueológico como una agrupación espacialmente definida y funcionalmente significativa de vestigios materiales de actividades humanas desarrolladas en el pasado [3]. De tal manera, cuando en un inventario arqueológico, como es el caso del IAN, se plasma una delimitación espacial de un enclave arqueológico debe haberse producido una observación de la materialidad de la evidencia y, en consecuencia, una interpretación arqueológica de la misma.

De manera habitual, se utilizan cuatro tipologías para diferenciar yacimientos arqueológicos: lugar de habitación, lugar de producción, lugar ritual y lugar funerario. No obstante, en el IAN se recogen estas, y otras tres más: lugar de arte rupestre, obras públicas y otras tipologías (Figura 2 y 3). Esta división se fundamenta en interpretar al yacimiento arqueológico como un ámbito deposicional, donde se sedimentan, se estratifican y se alteran (o no) los residuos materiales de la vida humana. Hay que tener en cuenta que dentro de estas cuatro grandes

tipologías se podrán encontrar multitud de clases funcionales de yacimientos (tipos) que están directamente relacionadas con la complejidad de la sociedad que las haya generado [3].

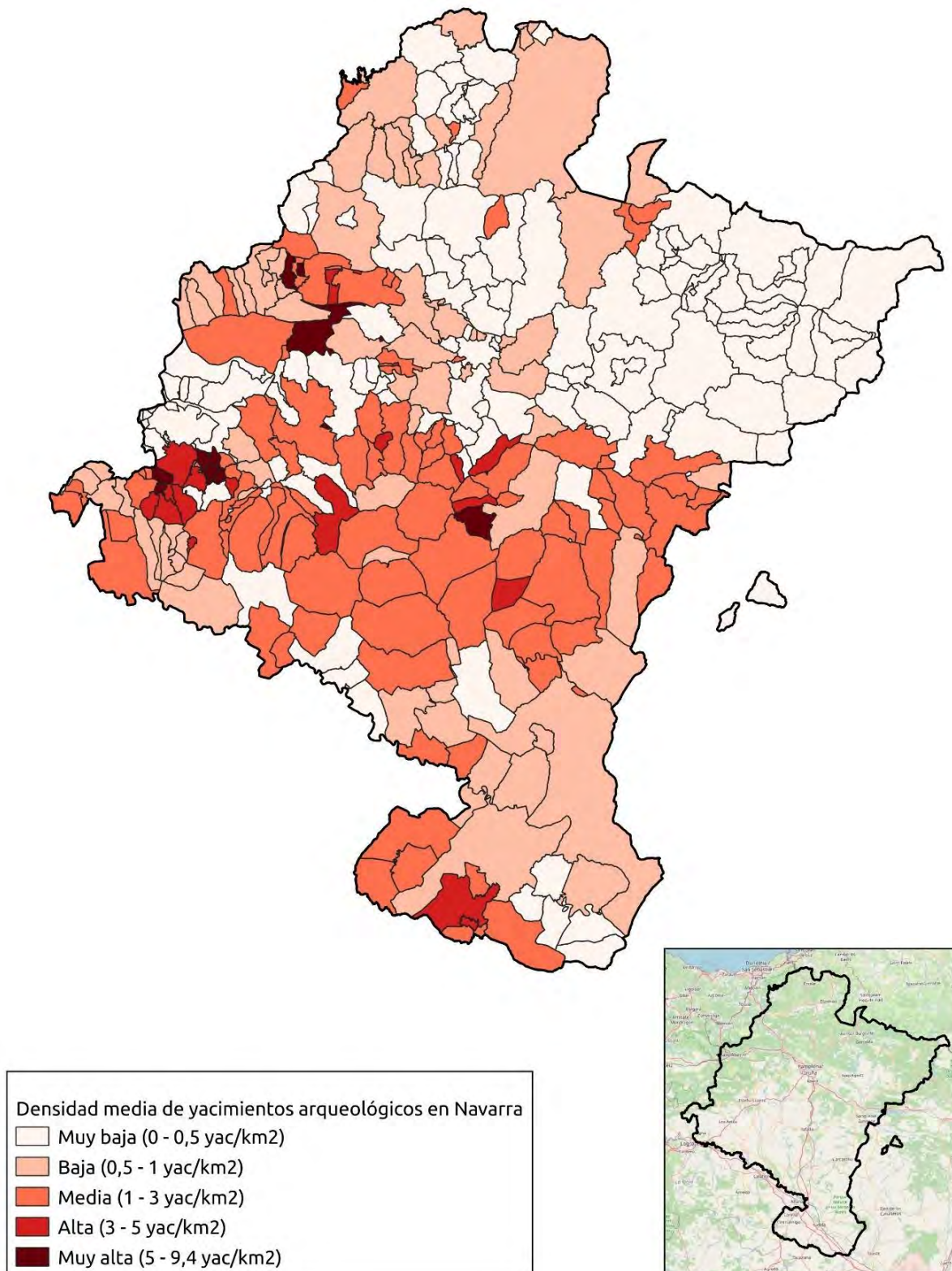


Figura 1. Mapa de densidad media de yacimientos arqueológicos por municipio registrados en el Inventario Arqueológico de Navarra. Elaboración propia.

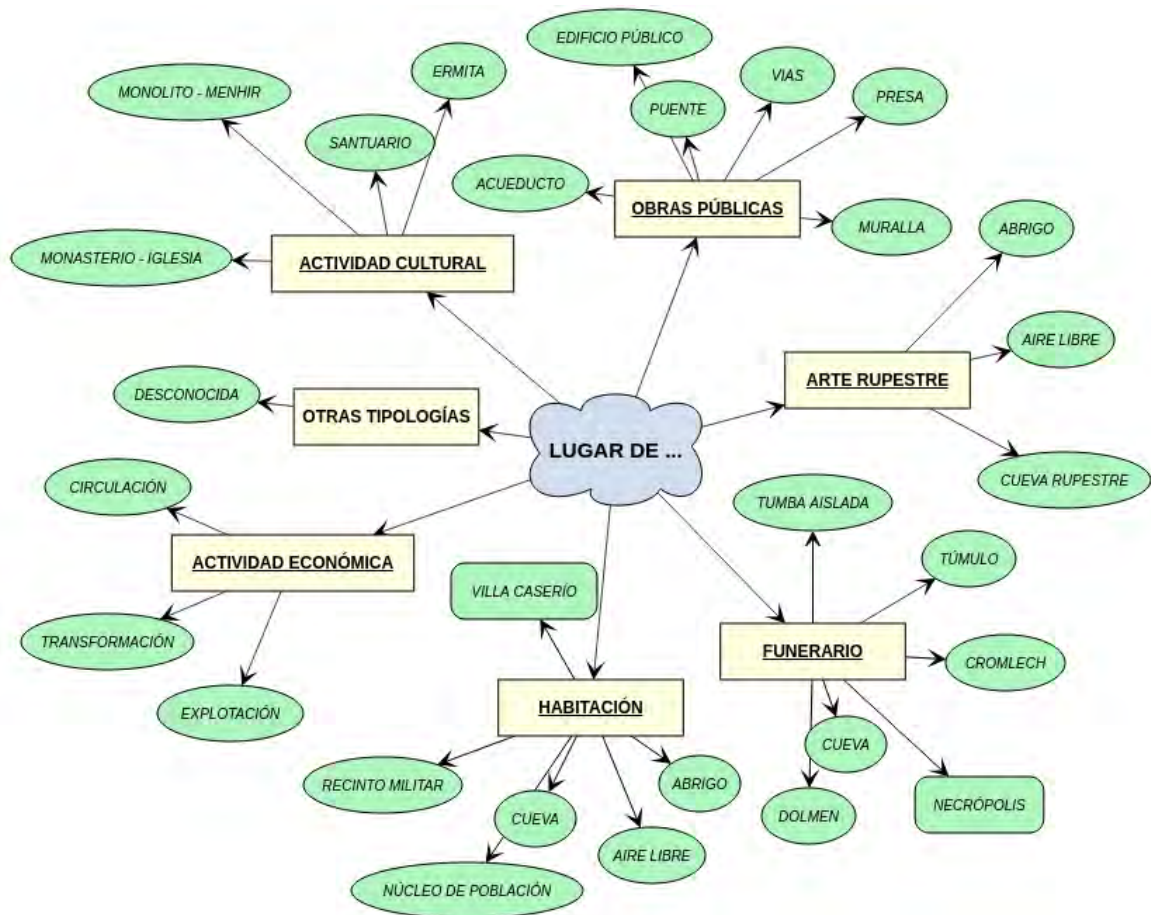


Figura 2. Mapa conceptual para relacionar las tipologías y las clases de yacimientos arqueológicos que figuran en el Inventario Arqueológico de Navarra. Elaboración propia.

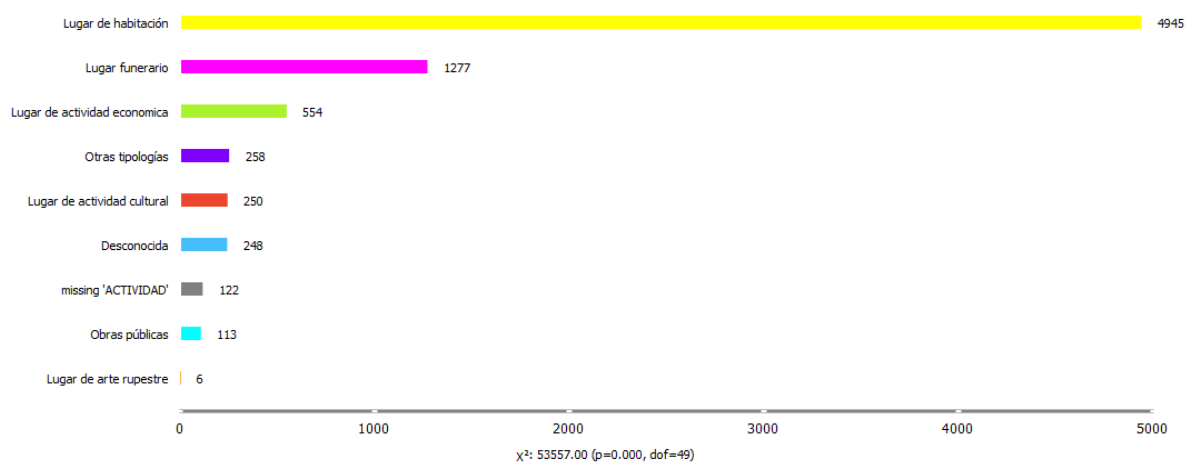


Figura 3. Cantidad de yacimientos arqueológicos inventariados por tipología, se observa una clara desproporcionalidad hacia los entornos arqueológico de tipo "Lugar de habitación". Elaboración propia.

1.2.- El reconocimiento del territorio en arqueología

Existen dos grandes grupos de técnicas con las que observar la materialidad de la evidencia: la prospección pedestre y la realizada con sensores remotos. Estas técnicas se interrelacionan y se complementan entre sí. No obstante, es la escala de observación la que determinará qué técnica o procedimiento utilizar y, siempre, será necesario una comprobación *in situ* que determine si una observación corresponde o no a un yacimiento arqueológico.

Se ha comentado anteriormente que el IAN se fundamenta principalmente en prospecciones de tipo pedestre, quedando a disposición de las empresas las propuestas metodológicas donde se recoja cómo se van a utilizar las técnicas de prospección mediante sensores remotos. Hay que mencionar que los pliegos técnicos de estas licitaciones ponen énfasis en el empleo de datos LiDAR como complemento de la prospección, especialmente en las zonas de cobertura vegetal extensa.

En este trabajo se analiza el uso de datos LiDAR o, mejor dicho, los productos derivados de dichos datos, aportando soluciones técnicas automatizadas que permitan discriminar si una anomalía observada en estos datos corresponde o no a un elemento arqueológico.

1.3.- El uso de sensores remotos en arqueología

Mediante la aplicación de sensores remotos es posible realizar un conjunto de acciones encaminadas a la prospección y a la monitorización de yacimientos y sus entornos de una manera rápida y dinámica a partir de múltiples fuentes de datos [4]. Según la escala de observación se diferencia dos tipos de sensores: los terrestres y los aéreos.

Los **sensores remotos de tipo terrestre** se usan como herramientas para prospecciones a nivel micro dentro de entornos arqueológicos bien delimitados. Los métodos y técnicas más comunes se relacionan con las propiedades físicas del suelo como son la resistividad eléctrica y el electromagnetismo, usándose herramientas (Figura 4) como el Ground Penetrating Radar (GPR) y el magnetómetro [5].

Los **sensores remotos de tipo aéreo** pueden ser activos, es decir, que emiten su propia energía y recogen información de cómo responde la superficie terrestre a esa energía; o pasivos, donde la fuente de energía es la radiación solar reflejada sobre la superficie terrestre (o la emisión de radiación propia de los objetos presentes en la superficie). En el reconocimiento arqueológico del territorio se usan como sensores activos las tecnologías LiDAR (Figura 7) y RADAR, y como sensores pasivos las imágenes espectrales, multi e hiperspectrales, y la fotografía aérea.

A nivel teórico se supone que los restos arqueológicos producen contrastes localizados en su entorno de soporte, los cuales pueden ser detectados usando el sensor adecuado. Estos contrastes pueden ser clasificados como marcas de cultivo, marcas de suelo, marcas de sombra, marcas de hielo, marcas de humedad, marcas de nieve y marcas de inundación; siendo los identificadores de suelo y de cultivo los que más han sido estudiados mediante el uso de sensores remotos. La identificación de este tipo señales han sido definidas durante el siglo XX por Crawford [7] y Wilson [8], fundamentalmente a través de la fotografía aérea. Así bien, las investigaciones de Lasaponara y Masini [9], [10] a partir de la primera década del siglo XXI pusieron el énfasis en las imágenes multiespectrales y la combinación de bandas para resaltar este tipo de marcas, poniendo las bases para el uso de una nueva herramienta en arqueología [11].

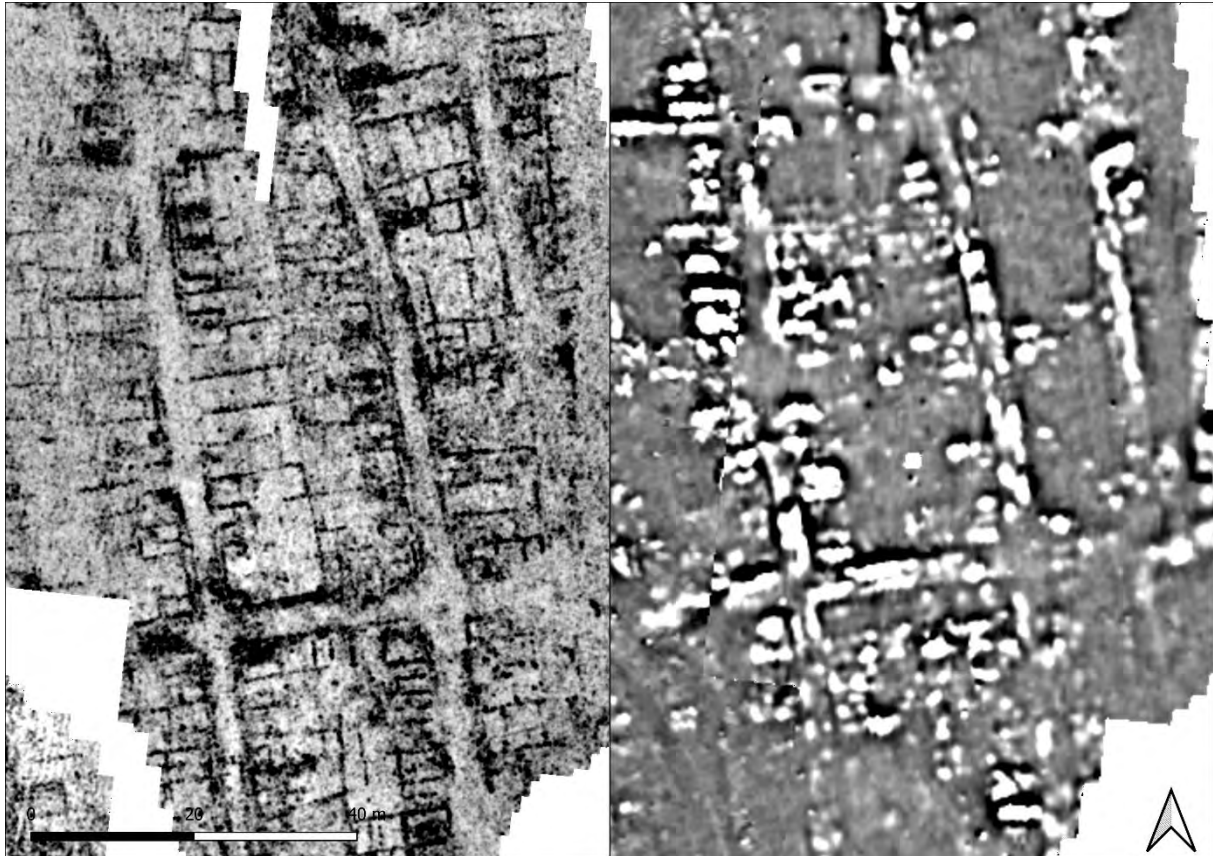


Figura 4. Prospecciones de tipo geofísico (GPR y magnetómetro) realizadas por SOT Prospección Arqueológica dentro de un enclave arqueológico bien delimitado. Proyecto de investigación arqueológica de La Custodia (Viana) bajo la dirección científica de Javier Armendáriz Martija. Elaboración propia.

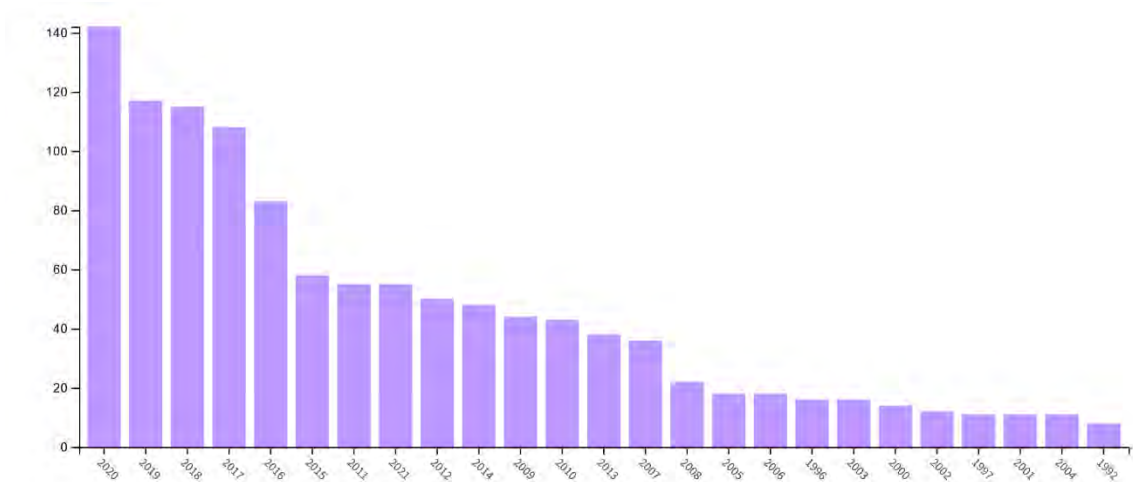


Figura 5. Relación de publicaciones científicas localizadas en Web of Science bajo el epígrafe "Remote Sensing and Archaeology". La tendencia al alza en los últimos veinte años se relaciona directamente con la mejora en los sistemas de captación de los sensores, en los tiempos de computación y en el propio desarrollo de la sociedad de la información. Fuente [6]

En cuanto a la identificación de **marcas de cultivo** (Figura 6) los procedimientos más adecuados son la fotografía aérea y las imágenes espectrales. Para detectar estas anomalías es necesario conocer las propiedades de la cubierta vegetal, el ciclo fenológico del cultivo,

las condiciones ambientales, el tipo de suelo y la topografía del terreno, de tal manera que se pueda elegir el momento adecuado para captar datos [11]. Las investigaciones realizadas en este ámbito han demostrado que es recomendable usar una alta resolución espectral y espacial para detectar anomalías en el crecimiento del cultivo, mayoritariamente, por interpretación visual [4].

En los últimos años se han desarrollado un conjunto de técnicas para mejorar la visibilidad de las imágenes basados en la fusión de datos obtenidos por combinación de bandas espectrales, filtros de paso alto y bajo, índices de vegetación y análisis de componentes principales [11]. Destacan los trabajos de Agapiou y el equipo del Laboratorio de Teledetección y Geoambiente de la Universidad Tecnológica de Chipre en el diseño de ecuaciones ortogonales basadas en la información obtenida de diferentes plataformas satelitales [12], [13] o el desarrollo de la aplicación ARCTIS, por parte de Atzberger, Wess, Doneus y Verhoven, de la Universidad de Viena, para el análisis de imágenes hiperespectrales [14]. En ambos casos se producen análisis que han permitido desarrollar metodologías que aumentan el contraste de las marcas de cultivo, caracterizándolas espectralmente, aunque estos procedimientos solo se pueden aplicar para los sensores que han sido diseñados.

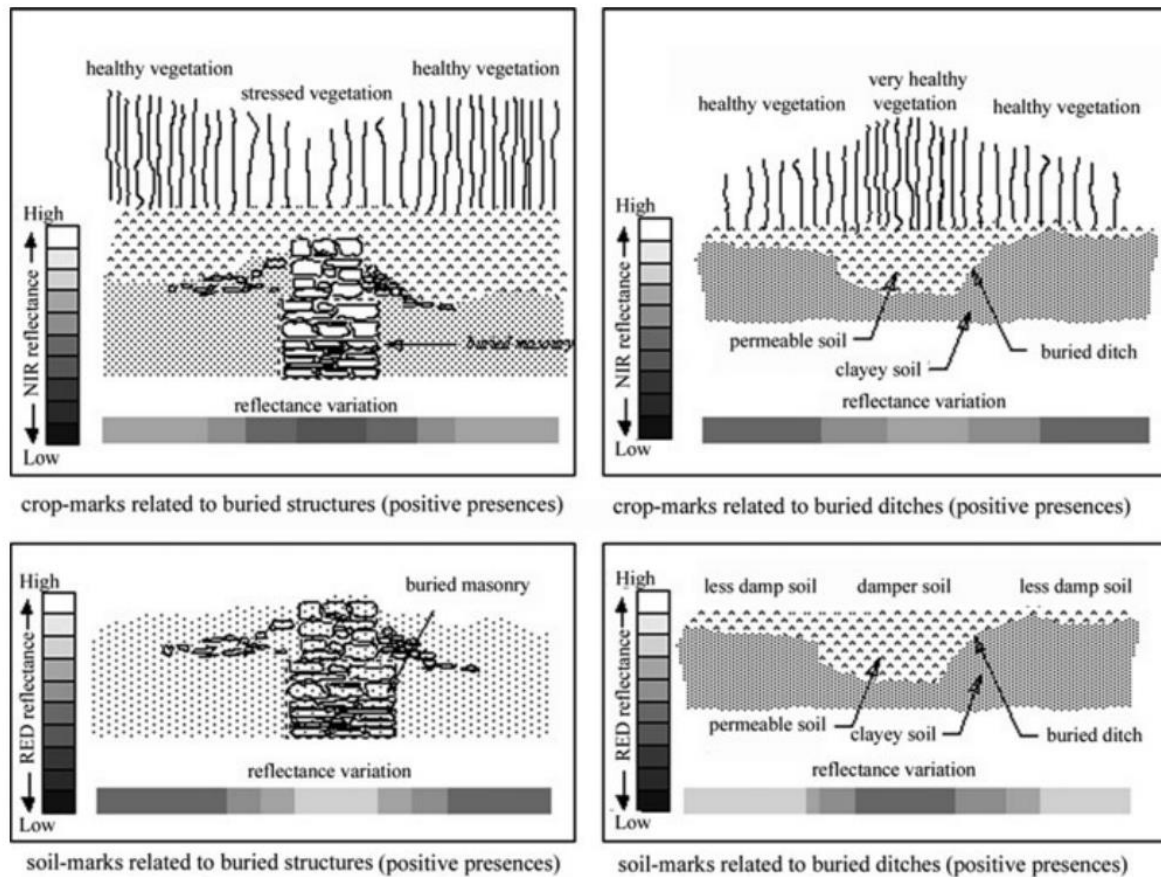


Figura 6. Variación de la reflectancia en marcas de cultivo y de suelo respectivamente. Fuente [11].

Para el reconocimiento de las **marcas de suelo** (Figura 6), los procedimientos más empleados están relacionados con la información obtenida de las diferentes tonalidades del suelo de las imágenes multiespectrales, de las variaciones en el efecto de retrodispersión de las imágenes RADAR, así como de los productos MDT obtenidos por los sensores LiDAR [4].

La información obtenida por los radares de apertura sintética ha demostrado ser válida para la detección de anomalías en entornos áridos y para la monitorización de subsidencias en enclaves arqueológicos [15]; la obtención de resultados depende completamente de la configuración del sensor en cuanto a polarización, frecuencia y escala temporal. Su combinación con los productos de imágenes multispectrales convierte al conjunto en una herramienta eficaz para el control de yacimientos arqueológicos [4].

Los sensores LiDAR requieren una mención aparte, debido a que su implementación en los flujos de trabajo en el reconocimiento arqueológico del territorio es extraordinaria, aunque su uso está aún por detrás de otras disciplinas científicas como la ingeniería forestal o el urbanismo.

1.3.1. La tecnología LiDAR en arqueología

LiDAR (*Light Detection And Ranging*) es una técnica de teledetección que emplea pulsos de luz, en forma de pulsos láser, para medir el rango (distancia) a una superficie [16] (Figura 7).

Este cálculo de distancia se basa en la medición del tiempo de retardo entre la emisión del pulso y la recepción del retorno de la señal. Además, también se mide la intensidad con la que el pulso retorna, informando del tipo de reflexión sufrido por el pulso laser, lo que permite extraer información sobre las superficies observadas [17].

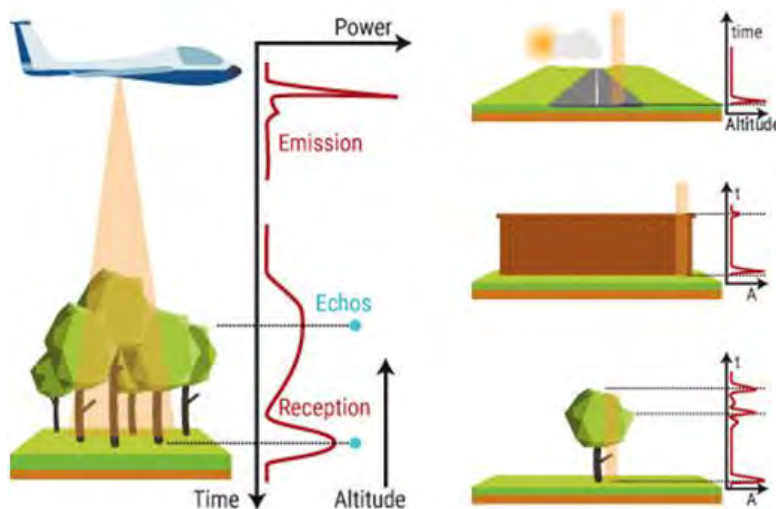


Figura 7. Esquema del funcionamiento de LiDAR. Fuente [18]

Los sensores LiDAR son sensores activos, es decir, emiten una señal y registran su eco, que operan generalmente en la región del infrarrojo cercano ($\lambda 1064 \text{ nm}$). Otras características de los sensores son la frecuencia de repetición del pulso, la frecuencia de escaneo, duración del pulso, patrón de escaneo (en zig-zag el más habitual), ángulo de escaneo, divergencia del pulso láser, la huella del pulso sobre el terreno, y la resolución radiométrica de la intensidad. La altura y velocidad del vuelo también condicionará los resultados, así como la densidad de puntos por cada metro cuadrado que se recojan durante el mismo [17].

Para aplicar esta tecnología al ámbito del patrimonio cultural se han analizado desde el punto de vista técnico por Fernández-Díaz, Carter y otros investigadores de la Universidad de Houston [19] las características de los vuelos LiDAR, analizando la altura de vuelo, al ángulo de incidencia, la densidad de puntos y la relación entre pulsos y retornos, centrándose en el

ámbito territorial mesoamericano, diseñando un manual técnico que permita a los equipos de investigación planificar vuelos con sensor LiDAR para objetivos arqueológicos [19]. Más allá de estas consideraciones, el conjunto de investigaciones publicadas se centra en el análisis de los productos derivados de LiDAR, como son los modelos digitales del terreno donde destaca la aplicación *Relief Visualization Toolbox* del equipo de investigación del Centro de investigación de la Academia de Ciencias y Artes de Eslovenia [20], [21]. Esta aplicación integra algoritmos contrastados (*Sky View Factor*, *Openness*, *Local Dominance*, *Simple Local Relief Model*, etc.) que mejoran la visualización de anomalías (Figura 8), arqueológicas o no, y que pueden usarse de manera eficiente para la extracción automática de singularidades [22], [23]. Si bien, la mayor potencialidad de LiDAR no radica exclusivamente en sus productos derivados, sino en la capacidad de filtrar y clasificar la información obtenida por el sensor [19].

1.3.2. La teledetección en España

En el ámbito del estado español se detecta el mismo patrón de aumento exponencial de publicaciones científicas relacionadas con el uso de técnicas de teledetección en el reconocimiento arqueológico del territorio. No obstante, es importante reseñar la existencia de publicaciones científicas a modo de manuales [3] o de revisiones metodológicas [24], a caballo entre la última década del siglo XX y la primera del siglo XXI, y que actúan como referentes teóricos. Se puede considerar que no será hasta la segunda década del siglo XXI cuando la aplicación práctica de estas metodologías muestre un aumento exponencial en los proyectos arqueológicos [25], fundamentalmente por las mejoras en la accesibilidad a información y la reducción de los costes computacionales en la fase de procesamiento.

Existen un buen número de publicaciones que han demostrado que el uso de la tecnología LiDAR es válida para analizar arqueológicamente el territorio [25]. El auge del uso de esta técnica se debe al desarrollo del Plan Nacional de Ortofotografía Aérea del IGN que da acceso libre a los datos recogidos por LiDAR topográfico en todo el conjunto de la península. La combinación de productos de fotografía aérea y modelos digitales del terreno creados a partir de LiDAR ha sido muy influyente en el avance de los estudios sobre la presencia militar romana en el NW de la península [26]–[30] y en la detección de estructuras megalíticas [31]–[33], o en la localización de grandes estructuras como el anfiteatro romano de Torreparedones en la provincia de Córdoba [34].

Destacan otras investigaciones, aunque relativas al uso de imágenes espectrales, para la caracterización de yacimientos y entornos arqueológicos. El equipo de investigación de la ciudad celtíbera de Segeda (Zaragoza) implementó en su flujo de trabajo, desde 2002 hasta 2012, diferente información obtenida por sensores remotos: imágenes multiespectrales, hiperspectrales, térmicas y, también, obtenidas por radar de apertura sintética. Usando procedimientos asociados a la fusión de datos y al análisis de los componentes principales (PCA) les permitieron detectar anomalías de origen arqueológico a diferentes escalas de observación [35], [36].

Otros casos de éxito se basan en información espectral adquirida desde vehículos aéreos no tripulados (UAV). Usando este medio, dos investigadores de la Universidad del País Vasco y del Museo Arqueológico de Londres, Fuldain y Hernández, han determinado el trazado de una infraestructura viaria romana, a su paso por la provincia de Álava, mediante el análisis de marcas de cultivo a partir del índice de vegetación NDVI [37], [38]. Los autores destacan que utilizando UAV's se ha reducido significativamente el tiempo y coste de las acciones de prospección, y que mediante el uso del índice NDVI han podido identificar partes conocidas y desconocidas del trazado de esta vía [37]. Sin duda, el uso de este tipo de plataformas es

una de las tendencias de mayor interés en el reconocimiento arqueológico del territorio, ya que aporta una muy alta resolución espacial de las imágenes espectrales. Su uso también ha sido probado con éxito por Orengo y Garcia-Molsosa en la región griega de Tracia como apoyo y mejora de las prospecciones pedestres, donde se ha llevado a cabo la primera prueba de registro automatizado de la dispersión de la cultura material utilizando imágenes de drones de alta resolución, fotogrametría y una combinación de aprendizaje automático y análisis geoespacial ejecutado en *Google Earth Engine* [39].

No obstante, el futuro del reconocimiento arqueológico del territorio no pasa exclusivamente por utilizar un tipo de sensor o un tipo de medio de transporte, sino por la integración de información diversa obtenida desde diferentes escalas y por distintos sensores; por la creación de bases de datos accesibles donde localizar esa información; en la complementariedad de la prospección terrestre con la prospección por teledetección y en desarrollar procedimientos de clasificación y segmentación basada en objetos.

1.4.- Hacia la detección automática de los entornos arqueológicos

De manera general, las características arqueológicas se pueden extraer de las imágenes como datos estadísticos y morfológicos mediante la interpretación manual asistida, con herramientas de un sistema de información geográfica [41]–[44], o por detección automática después de la mejora de la imagen [4].

Las investigaciones científicas, que utilizan procedimientos y análisis por detección automática de anomalías arqueológicas, son aún muy minoritarias en la península ibérica [25], [31], [32]. Así bien, éstas sí que se están dando en el contexto internacional [4], [45]–[49] aunque principalmente en países cuyos programas e infraestructuras de investigación están más consolidados [50].

La detección automática de características arqueológicas se puede considerar un problema de clasificación (Figura 9) que, en esencia, es de carácter predictivo [31], [39], [44], [51], [52]. Para resolver un problema de clasificación es necesario encontrar aquella función (modelo) que explique mejor la separabilidad de nuestros datos y, por tanto, pueda predecir la pertenencia a una clase o a otra, ya sea de un dato conocido o por conocer [53], [54].

La resolución de este tipo de problemas de clasificación es muy común en teledetección [55], estableciéndose la separabilidad de los datos al asociarlos a características individuales o conjuntos de ellas, que permiten separarlos en diferentes clases. Estas características, que describen nuestros datos de partida, pueden asociarse a la composición o geometría de los objetos observados. Típicamente, si provienen de un sensor pasivo se asociarán a nuestros datos características relativas a la reflectividad y a la composición de la cubierta terrestre (material, química, etc.), mientras que si provienen de un sensor activo corresponderán principalmente a características físicas de la cubierta y sus elementos (forma, tamaño, humedad, etc.).

Los procedimientos utilizados para la resolución de este tipo de problemas se diferencian en aproximaciones basadas en píxeles y aproximaciones basadas en objetos.

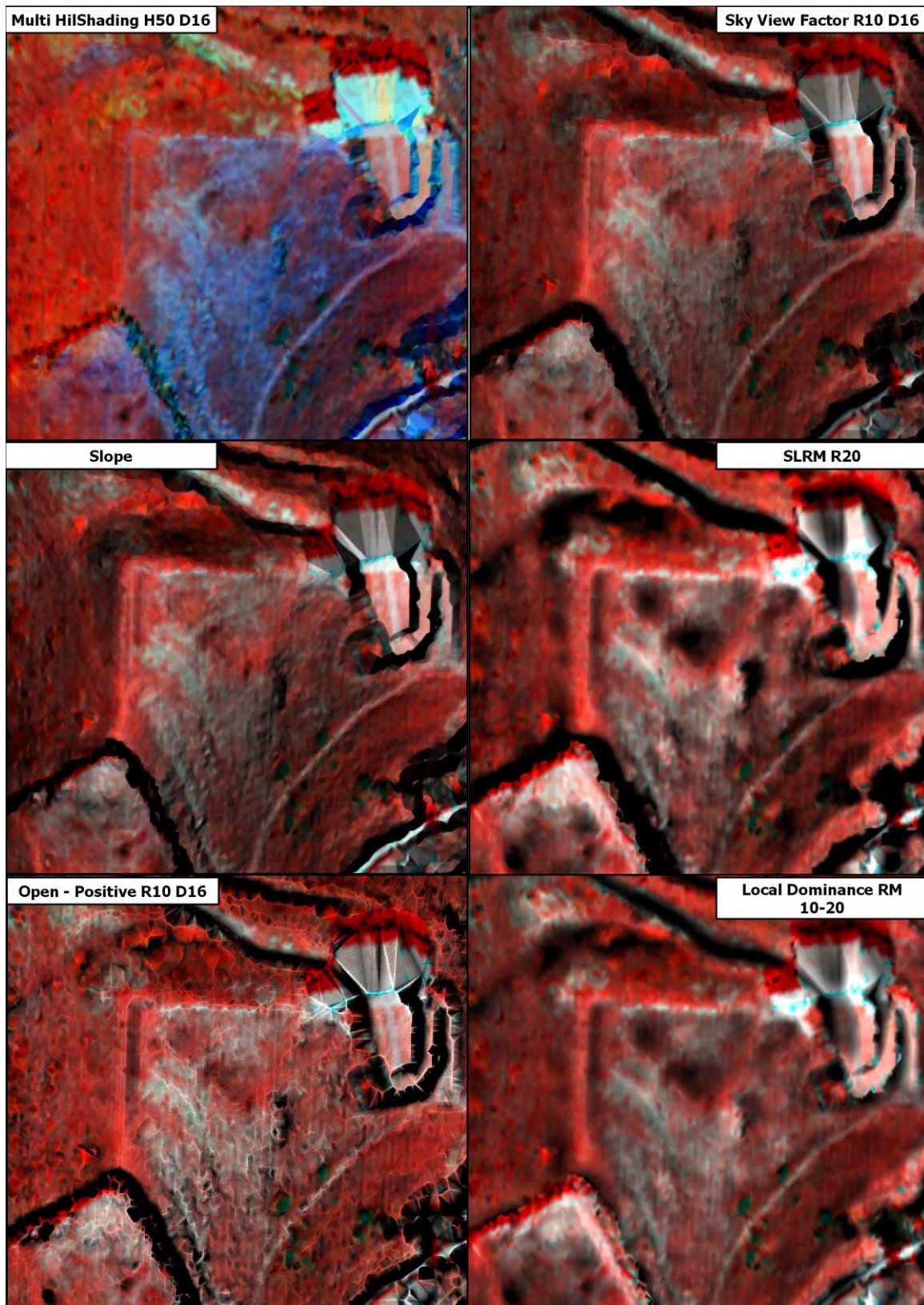


Figura 8. Ejemplo del uso combinado de diferentes sistemas de visualización producidos a partir del modelo digital de elevación del vuelo LiDAR de 2012 y la información espectral (RGBI) obtenida durante el vuelo. Fuente [40]



Figura 9. El aprendizaje automático forma parte del proceso de descubrimiento a partir de bases de datos. Las principales técnicas del aprendizaje automático corresponden a las que se ilustran en esta figura realizada por Natalia Acevedo. Fuente [56]

Las **aproximaciones basadas en píxeles** son aquellas cuya unidad de análisis es el píxel [45], [57]–[59] y se ocupan de la clasificación de píxeles individuales de una imagen en diferentes categorías que corresponden a características del paisaje [45]. Aunque se trata de un procedimiento muy utilizado en teledetección, se ha observado que para la detección de características arqueológicas son más eficientes las **aproximaciones basadas en objetos** [43], [45], [57].

La aproximación basada en objetos se conoce en la bibliografía científica como *Object-Based Image Analysis* (OBIA). Si estos objetos contienen una variable geográfica o espacial, la bibliografía se referirá a estos análisis como Geo-OBIA. De manera general, el método OBIA pretende obviar la artificialidad de los análisis basados en píxeles [59], agrupando un conjunto de valores n – dimensionales [45] en formas representativas y homogéneas de las categorías de clasificación.

El método OBIA se basa en dos grandes pasos: segmentación y clasificación. Existen multitud de procedimientos para segmentar la imagen en objetos: *edge-based*, *region-based*, *region growin/merging*, *region splitting and merging*, métodos híbridos y métodos semánticos. La publicación realizada por Hossain y Chen en 2019 del Laboratorio de Análisis Espacial e Información Geográfica de la Universidad de Queen's (Canadá) es un excelente recurso para conocer las ventajas y desventajas de estos métodos y los softwares disponibles para ejecutarlos [59]; aunque si se requiere información más específica sobre los métodos y softwares más comunes en la detección automática en arqueología habrá que acudir a Davis [50], de la Universidad estatal de Pensilvania (EE.UU.). La elección de un procedimiento de segmentación u otro dependerá del tipo de imagen o imágenes que se vayan a utilizar y, por tanto, de las variables que se quieran incorporar al proceso de clasificación; es decir, la clasificación, en última instancia, está condicionada por el tipo de

segmentación que se utilice [59]. Así también, los procesos de segmentación sólo tienen sentido con productos obtenidos de sensores con alta resolución espacial.

Los principales estudios que han aplicado OBIA a la detección automática de enclaves y características arqueológicas suelen utilizar variables que corresponden a valores espectrales, texturales, morfométricos y relacionales [45], [58]. Estos estudios se encuentran referenciados en el análisis realizado por Davis en 2019 [45]. Según este autor, hasta 2019, se habían realizado 35 publicaciones donde el método OBIA era su principal procedimiento de análisis. Davis separa las publicaciones entre las que su objetivo fue identificar enclaves y características (28), monitorear yacimientos arqueológicos (6), digitalizar características arqueológicas (3) y analizar arqueológicamente el territorio (4). Estas 35 publicaciones también han sido contextualizadas por Magnini y Bettineschi de la Universidad de Padova (Italia) [58], aunque diferenciándolas por la extensión de su área de estudio en microscópicas (4), escala ítem (2), de nivel local (11), local / regional (2) y de escala regional (17). Según estos datos, se puede afirmar que los estudios que utilizan OBIA en el ámbito de la arqueología principalmente tienen por objetivo identificar anomalías de origen arqueológico a escala regional, utilizando imágenes de alta resolución espacial.

Desde mitades de la pasada década y en paralelo al auge de los productos derivados de LiDAR [20], [22] se han utilizado un sinnúmero de parámetros geométricos (forma, dimensión, volumen, circularidad, rectangularidad, índice de posición topográfica, curvatura, pendiente, sombreado analítico u orientación) en los que fundamentar el proceso de segmentación combinándolos con diferentes algoritmos de clasificación. Atendiendo a la evaluación de la fiabilidad de las clasificaciones publicadas (Figura 10), se observa que no es hasta el año 2020 cuando se empieza a localizar expresamente en las publicaciones los valores de la matriz de confusión, este hecho quizás pueda deberse a la falta de homogenización metodológica en cuanto al uso del aprendizaje automático en arqueología. Siendo las métricas más utilizadas: *Precision*, *Recall* y *F1 score*² y que fundamentalmente se han utilizado para evaluar problemas binarios no balanceados.

Destacan sobre todo el conjunto, las investigaciones [23], [51], [60] realizadas por Guyot, Hubert-Moy, Lennon de la Universidad de Rennes (Bretaña, Francia) y de Lorho del Servicio Regional de Arqueología de Bretaña. Estas investigaciones, centradas en la identificación de túmulos, han demostrado que, si se analiza el territorio desde una perspectiva multiescalar para clasificar de manera binaria entornos y características arqueológicas homogéneas el rendimiento del clasificador roza casi la perfección, con un índice kappa del 0,984, por tanto, se considera un clasificador muy bueno.

Estos autores han planteado una metodología basada en un modelo digital del terreno de 0,25 m/px, producido a partir de un vuelo LiDAR multiespectral (532 nm y 1064 nm). Con este MDT han elaborado un producto llamado *Multiscale Topographic Position Image* [61], [62]

2 Precision: Se refiere a la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión. Se representa por la proporción entre el número de predicciones correctas (tanto positivas como negativas) y el total de predicciones.

Recall: La sensibilidad es el valor que nos indica la capacidad de nuestro estimador para discriminar los casos positivos, es la fracción de verdaderos positivos que fueron correctamente identificadas por el algoritmo.

F1 score: Nos resume la precisión y sensibilidad en una sola métrica, por ello es de gran utilidad cuando la distribución de las clases es desigual.

creado usando las desviaciones de la elevación en la microescala (1 a 10 m), la mesoescala (10 a 100 m) y la macroescala (100 a 1000 m). Los valores para estos parámetros han sido establecidos en relación con los datos empíricos del tipo de objeto a clasificar: el túmulo.

El problema se ha planteado como un problema binario (túmulo – no túmulo); se han utilizado 50 muestras de datos asociadas a un área de 1 km², cuyas localizaciones han sido distribuidas espacialmente mediante la técnica estadística *Latin hypercube* [63]–[65], la cual garantiza la variabilidad de los datos. El 88% de este *dataset* se compone de datos aleatorios y el 12% corresponden a datos confirmados como túmulos por el Servicio Regional de Arqueología de Bretaña. Se trata, por tanto, de una muestra pequeña y no balanceada. La forma elegida para cada uno de los registros, aleatorios o no, ha sido un cuadrado de 100 m², según los valores estadísticos de los túmulos. A cada una de estas localizaciones se le ha añadido la información de la imagen multiescalar para luego separar el *dataset* en dos bloques: uno para el entrenamiento (70%) y otro para la validación (30%). El modelo de clasificación elegido ha sido un *Random Forest* con 120 árboles de decisión que ha devuelto un mapa de probabilidad en el que el valor de 0,5 es el criterio para distinguir entre la clase negativa (no es túmulo) y la positiva (túmulo). Una vez entrenado el modelo se han procedido a inferir sus pesos a otras áreas del suroeste de la Bretaña francesa.

Los datos de la matriz de confusión de esta investigación devuelven unos valores que lo caracterizan como un excelente clasificador, siendo muy válido para separar la clase positiva de la negativa. El valor de la métrica *F1 score*, utilizada para este tipo de problemas binarios no balanceados, corresponde a 0,9855.

Este caso de éxito se debe a la combinación de productos de alta resolución espacial, técnicas de visualización de datos adaptadas al objeto de estudio (los túmulos son promontorios en el terreno, por tanto, son anomalías topográficas), procesos de segmentación simples y a la selección de un modelo sólido y adaptado al objeto de estudio. Aunque, estos autores han experimentado en su área de estudio con modelos de aprendizaje automático mucho más novedosos [60] como son las redes neuronales convolucionales, consideran que si el objeto de estudio es homogéneo es mucho más efectivo usar un modelo consolidado como *Random Forest*, que resolverá mucho mejor un problema binario, tanto en resultados, como en procesamiento e implementación, si lo comparamos con una red neuronal [51].

En los últimos dos años en el flujo de trabajo de la detección automática de entornos y características arqueológicas los modelos de redes neuronales y la técnica de transferencia de conocimiento están convirtiéndose en la vanguardia de la investigación. Se trata de la utilización de un tipo de modelos de aprendizaje automático que se aplican a la investigación arqueológica de manera pre-entrenada, siendo pocas las redes específicamente entrenadas para resolver problemas del dominio de la arqueología como sería el caso de la red *CarcassonNet* [48]. Los modelos utilizados están diseñados para resolver problemas de segmentación a base de crear máscaras de los objetos [66], [67], de redes basadas en la identificación de patrones de imagen [52], [60] o bien de detección de objetos [47]. Se considera que estos métodos, aunque novedosos, no son eficientes para resolver los problemas que plantea la detección automática de enclaves arqueológicos. Se detecta en el sector, una falta de homogeneidad en los criterios técnicos y teóricos de la detección automática de características arqueológicas [58], [68], así como en la delimitación física (en un entorno bidimensional como es un plano, imagen, etc.) del concepto yacimiento arqueológico, ya que como se ha reseñado anteriormente se trata de una acción completamente arbitraria (Tabla 1).

A esta falta de homogeneidad en los criterios técnicos y teóricos se le debe añadir que los análisis, realizados los últimos años, están directamente relacionados en intentar entrenar modelos de aprendizaje a partir de las tipologías (por ejemplo, lugar funerario) y tipos (dolmen, túmulo, etc.) por las que se etiquetan los yacimientos arqueológicos. Es decir, que el proceso de homogenización de los datos (valores topográficos de LiDAR) se realiza por un criterio que nada tiene que ver con los datos. Así también se intuye en la bibliografía consultada una tendencia en los objetivos de las investigaciones muy centrada en los conceptos de localización y descubrimiento.

Tabla 1. Investigaciones de los últimos 10 años realizadas para detectar automáticamente entornos arqueológicos. Fuente [49]

Authors	Objects to detect	Remote sensing data	Method	True positive rate	False positive rate
Hesse, 2013	Potential archaeological features	Airborne laser scanning (ALS), 1/m ²	Manual interpretation of digital terrain model (DTM) visualization		
Bescoby, 2006	Roman land boundaries	Historic aerial photos	Radon transform		
Sevara et al., 2016	Burial mounds in grave field	ALS, 6/m ²	DTM openness + segmentation	91%	6%
Sevara et al., 2016	Various archaeological features	ALS, 5/m ²	DTM openness, slope, roundness + segmentation	100%	35%
Zingman, Saupe, Penatti, & Lambers, 2016	Fragmented rectangular enclosures	Satellite, optical 0.5 m	Rectangle detector	100%	34%
Zingman et al., 2016	Fragmented rectangular enclosures	Satellite, optical 0.5 m	Pre-trained deep convolutional neural network (CNN)	100%	124%
Trier et al., 2018	Charcoal burning platforms	ALS, 5/m ²	Template matching	70%	72%
Trier et al., 2018	Charcoal burning platforms	ALS, 5/m ²	Pre-trained deep CNN + support vector machine classifier	86%	37%
Trier, Larsen, & Solberg, 2009	Cropmarks of levelled grave mounds	Satellite, optical 0.5 m	Template matching		
Trier & Pilo, 2012	Pitfall traps	ALS, 7/m ²	Template matching + if-tests	86%	92%
Trier, Pilo, & Johansen, 2015	Burial mounds in grave field	ALS, 7/m ²	Template matching	65%	
Trier, Zortea, & Tonning, 2015	Grave mounds in forest	ALS, 1-22/m ²	Template matching + if-tests	50%	375%
Freeland, Heung, Burley, Clark, & Knudby, 2016	Earthworks mounds	ALS, 1/m ²	DTM local relief, ratios + segmentation	71%	14%
Freeland et al., 2016	Earthworks mounds	ALS, 1/m ²	Inverted pit filling	85%	18%
Cerrillo-Cuena, 2017	Prehistoric barrows	ALS, 0.5/m ²	Curvature, topographic position index, circular Hough transform	46%	
Toumazet, Vautier, Roussel, & Dousteyssier, 2017	Grazing structures	ALS, 11/m ²	DTM local relief, segmentation, template matching	91%	34%
Guyot, Hubert-Moy, & Lorho, 2018	Burial mounds	ALS, 14/m ²	DTM local contrast at three scales, random forest classifier	98%	1%

La hipótesis de partida de este trabajo considera que la localización y descubrimiento se puede dar mediante técnicas de aprendizaje automático, pero no tanto desde el concepto de identificar, sino en el de proponer localizaciones cuya probabilidad de ser o no un yacimiento arqueológico se han aprendido a partir de una homogeneización de los valores continuos de los inventarios arqueológicos y no a partir de los valores categóricos relativos a tipos y tipologías.

1.5.- Objetivos

El propósito de esta investigación es crear nuevo conocimiento partiendo del conocimiento previo existente (IAN) a partir de la hipótesis mencionada.

Visto el contexto de las investigaciones se considera que es viable la aplicación práctica de un procedimiento tipo OBIA usando el método de análisis multiescalar desarrollado por Guyot y otros [51] para evaluar nuestra hipótesis de partida y, por tanto, se plantea un procedimiento metodológico similar, así como comparable.

A su vez, la aplicación teórica de este trabajo pondrá al patrimonio arqueológico de Navarra en consonancia con las investigaciones internacionales en el campo de la detección automática de entornos arqueológicos.

Por otro lado, la aplicación práctica de esta metodología puede dotar a la Sección de Registro, Bienes Muebles y Arqueología del Gobierno de Navarra de un *dataset* de nueva creación que albergará información de las probabilidades de identificar nuevos yacimientos arqueológicos a comprobar e integrar en el Inventario Arqueológico de Navarra; así bien, el uso de diferentes productos derivados de LiDAR aportará nueva información a los yacimientos ya catalogados en cuanto a su morfología y delimitación.

Por otro lado, a nivel práctico los productos obtenidos en el proceso de ejecución de este trabajo tienen una aplicación directa en el trabajo diario de los equipos de arqueología y, por tanto, mejorar la eficiencia con la que se trabaja para la preservación y protección del patrimonio arqueológico de Navarra.

2.- Materiales y métodos

En el siguiente apartado se describe la metodología empleada, así como el origen de los datos utilizados con el objetivo que este estudio sea reproducible y comparable con el estudio de referencia [51] cuyo procedimiento metodológico se aplica para el caso de Navarra.

2.1.- Área de estudio

El ámbito de estudio elegido es el territorio de la Comunidad Foral de Navarra que corresponde a una superficie 10.391 km² siendo este un ámbito de escala regional en cuanto a su relación con los estudios de referencia.

Para la aplicación práctica de los modelos entrenados se ha elegido una ventana de análisis (Figura 10) localizada en las siguientes coordenadas UTM:

607688.863,4711873.041 // 607688.863,4713032.064 // 609045.132,4713032.064 // 609045.132,4713032.064

Se trata de un espacio localizado entre los municipios de Puyo y Tafalla en un ámbito geomorfológico con formaciones arcillosas en forma de sierras, colinas, cabezos y llanuras.

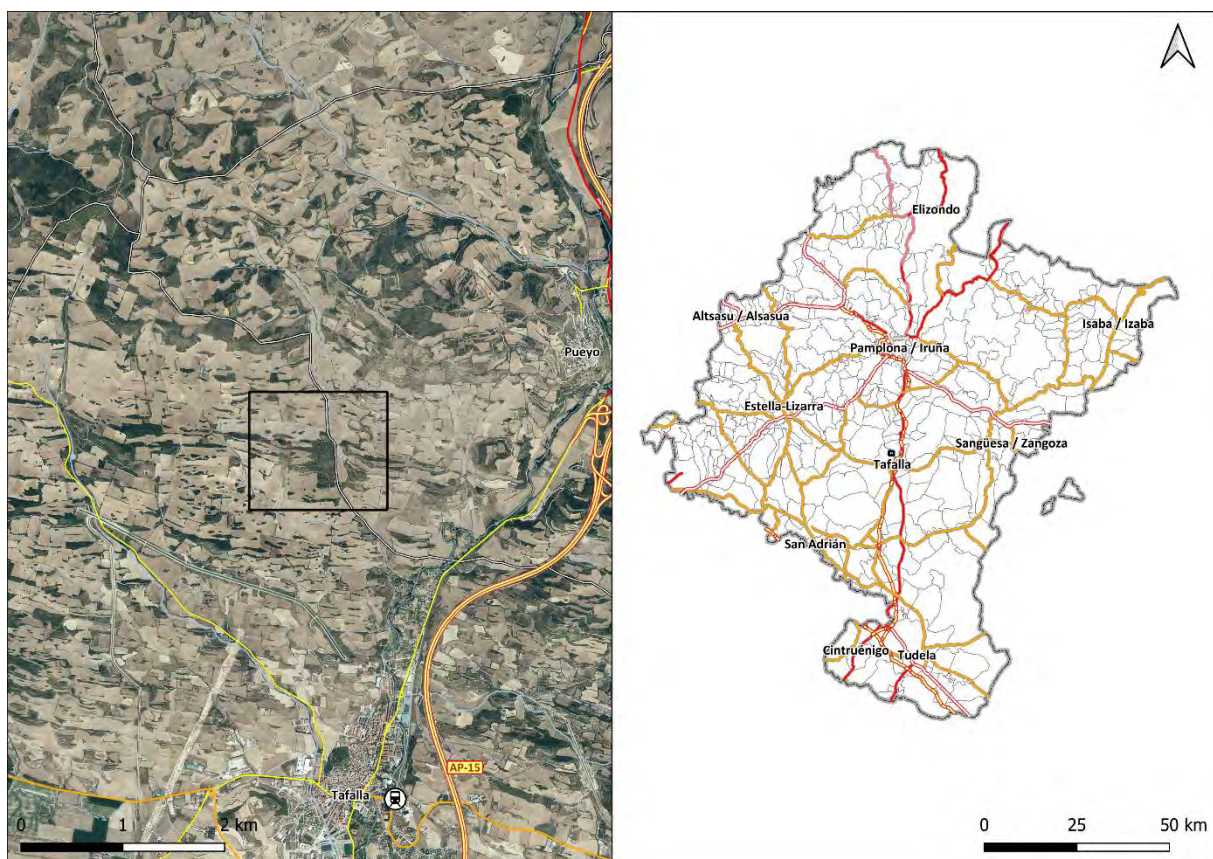


Figura 10. Localización de la ventana de análisis dentro del ámbito territorial de la Comunidad Foral. Fuente: Servicios wms de IDENA: REFERE_Lay_mapabase y IDENA: REFERE_Lay_baseorto Elaboración propia.

2.2.- Creación del *dataset*

El *dataset* de entrenamiento está compuesto de la información espacial, categórica y descriptiva del Inventario Arqueológico de Navarra y de la información espectral adquirida de diferentes tipos de imágenes producidas a partir modelo digital de elevación creado en 2017 por el Gobierno de Navarra.

El *dataset* se ha construido en cuatro pasos:

- Elaboración de los productos derivados de LiDAR (Figura 11).
- Selección de yacimientos arqueológicos del IAN (Figura 18).
- Preprocesamiento (Figura 19)
- Montaje del *dataset* (Figura 19)

2.2.1.- Elaboración de los productos derivados de LiDAR

En el marco del proyecto PNOA-LiDAR del Instituto Geográfico Nacional (IGN), el año 2017 se realizó la segunda cobertura completa de la Comunidad Foral de Navarra mediante el uso de un sensor aéreo *Single Photon LiDAR*, obteniéndose una densidad media de 14 puntos/m².

Todos los ficheros de este vuelo se encuentran a disposición pública, en el repositorio de cartografía del Gobierno de Navarra³, clasificados automáticamente con procesos de aprendizaje automático y depuración manual de los resultados. En el mismo repositorio se pueden encontrar los principales productos derivados de esta nube de puntos como son los modelos digitales de elevación (MDE). Por tanto, no es necesario procesar específicamente la nube de puntos sino utilizar los MDE que sean de interés. Actualmente, en el repositorio de cartografía de Gobierno de Navarra están a disposición los siguientes MDE:

- Modelo Digital del Terreno (MDT)
- Modelo Digital de Superficies (MDS)
- MDS menos MDT

Estos productos, se encuentran divididos en 400 archivos correspondientes con las hojas de la malla 1:10.000 del IGN⁴ y se distribuyen en alta resolución a 0,5 m/px o en una resolución menor de 2 m/px. Para este trabajo se han elegido los archivos del MDT de alta resolución (0,5 m/px) de toda la comunidad foral; reseñar que en el estudio de referencia [51] se utilizó un MDT con una resolución de 0,25 m/px.

2.2.1.1.- El producto de multiescala

El estudio de Guyot, Hubert-Moy y Lorho se basa en el uso de los valores de multiescala para entrenar un modelo de tipo *Random Forest* que discrimine los entornos arqueológicos de tipo túmulo de los no arqueológicos [51]. No obstante, la metodología que han empleado se fundamenta en el estudio desarrollado por Lindsay, Cockburn y Russel que analiza el uso de imágenes integrales para medir la desviación métrica de la posición topográfica relativa, respecto a la elevación media (DEV) [61]; este procedimiento ha resultado adecuado, y un

3 https://filescartografia.navarra.es/5_LIDAR/

4 https://filescartografia.navarra.es/6_MDE/6_3_AÑO_2017/6_3_2_MDT/6_3_2_1_ASCII_Grid_EPS_G25830/6_3_2_1_1_50cm/

éxito a nivel de clasificador, debido a que el objeto de estudio es una elevación antrópica sobre el terreno, por tanto, es medible su desviación métrica respecto a la elevación media (DEV). Entendida esta como la diferencia entre la elevación de cada pixel de una imagen y la elevación media de su vecindad más próxima, dividida por la desviación estándar [51].

Ecuación 2.1

$$DEV(D) = \frac{(z_0 - \bar{z}_D)}{\sigma_D}$$

D = tamaño de la ventana iteradora; Z_0 = elevación del pixel central de la ventana; Z_D = Elevación media en la ventana. σ_D = Desviación estándar de las elevaciones de la ventana.

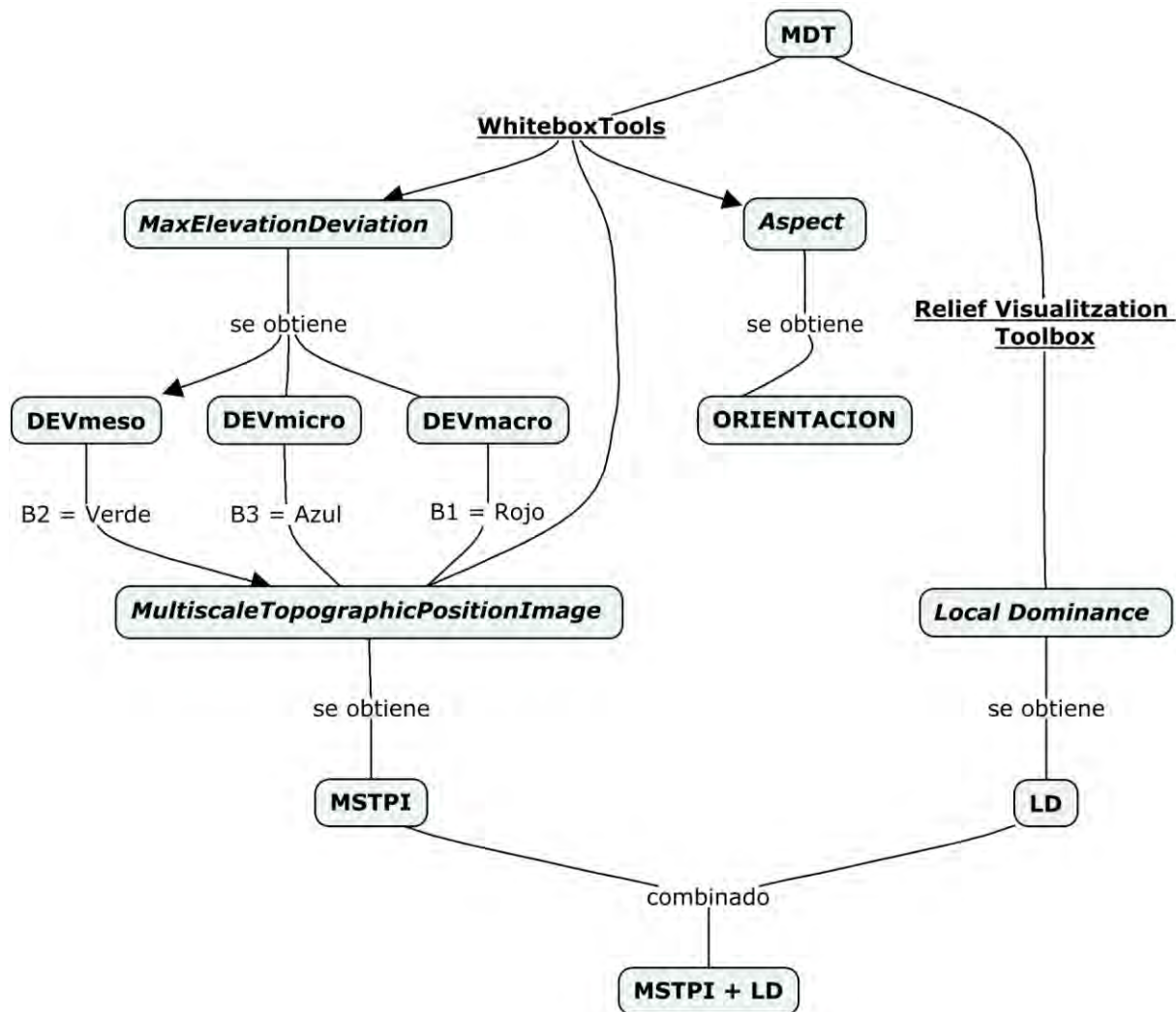


Figura 11. Diagrama de flujo de la metodología para obtener los productos derivados de LiDAR y sus modos de visualización. Elaboración propia.

Detrás de este análisis se encuentra el enfoque de imagen integral, que definió Franklin C. Crow en 1984 como una estructura de datos y un algoritmo que calcula de manera eficiente la suma de valores en un subconjunto rectangular de una malla [51], [61]. En el análisis multiescalar, la dimensión del subconjunto rectangular sobre la imagen integral se define

como escala o ventana de observación, mientras que los valores de DEV se obtienen de transformar el MDE en una imagen integral.

Los tamaños de la ventana elegidos coinciden con los valores planteados en el estudio de referencia [51] asociándose los valores de microescala de 1-10 m, mesoescala 10 – 100 m y macroescala de 100 – 1000 m.

En el estudio de referencia se justifica la elección de este tipo de escalas según los datos empíricos de la dimensión de los túmulos, siguiendo esta premisa se ha calculado el área de los yacimientos catalogados en el IAN, de tal manera que se pueda determinar si las superficies de los yacimientos están representadas dentro del análisis multiescalar. Según las dimensiones mencionadas (Figura 12), los yacimientos menores de 100 m² están representados en la microescala. Los yacimientos entre 100 m² y 10.000 m² corresponden a la mesoescala y los enclaves más grandes 10.000 m² se recogen en la macroescala.

Atendiendo a esta distribución, entorno al 20% de los yacimientos catalogados en Navarra se encuentran el ámbito de la macroescala; el 54% asociados a la mesoescala y un 25% a la microescala. De todo el conjunto un 1% se consideran valores perdidos o fuera de este rango de escalas.

La aplicación práctica de esta metodología se desarrolla ejecutando dos algoritmos integrados en la herramienta **WhiteboxTools**⁵ creada por Lindsay [62] como una plataforma de análisis espacial avanzado desarrollada en el seno del Grupo de Investigación en Geomorfometría y Hidrogeomática de la Universidad de Guelph's (Canadá).

La primera herramienta corresponde al algoritmo **MaxElevationDeviation** que se utiliza para calcular la desviación máxima de la elevación media o DEVmax [61] devolviendo un índice residual de la elevación, midiendo la posición topográfica relativa como una fracción del relieve local, por lo que se normaliza a la rugosidad de la superficie local. Este algoritmo se ha ejecutado, para cada una de las escalas reseñadas, devolviendo un archivo DEVmax normalizado en un rango de -3 a 3, rango en el que los valores de DEV tienden a estar bien representados [61] y cuya información hay que utilizarla como una signature espectral.

El segundo algoritmo, **MultiscaleTopographicPositionImage**, está enfocado para crear un archivo de visualización que se genera combinando los tres productos DEVmax obtenidos anteriormente. Esta herramienta utiliza los valores normalizados a 8 bits (rango de 0 a 255) de la macroescala, mesoescala y microescala para colocarlos en un sólo archivo, en el que la banda roja representa los valores macro, la verde usando valores de la mesoescala y la banda azul para los valores de la microescala. Se trata, por tanto, de una composición RGB de 24 bits que se ha demostrado que es útil para visualizar el carácter topográfico de la variante escalar del paisaje [61], así como ha resultado ser eficaz e informativa en relación a características morfológicas locales asociada a estructuras arqueológicas [51].

2.2.1.2.- El producto *Local Dominance*

Así bien, aunque la imagen de multiescala (*MSTPimage* [51] o *MSTPCC* [61]) es una composición apta para interpretar visualmente las anomalías en el relieve, no hay que descartar la combinación de la imagen de multiescala con otra técnica de visualización que los últimos años han demostrado ser muy útiles para la interpretación arqueológica del territorio [22].

5 https://www.whiteboxgeo.com/manual/wbt_book/intro.html

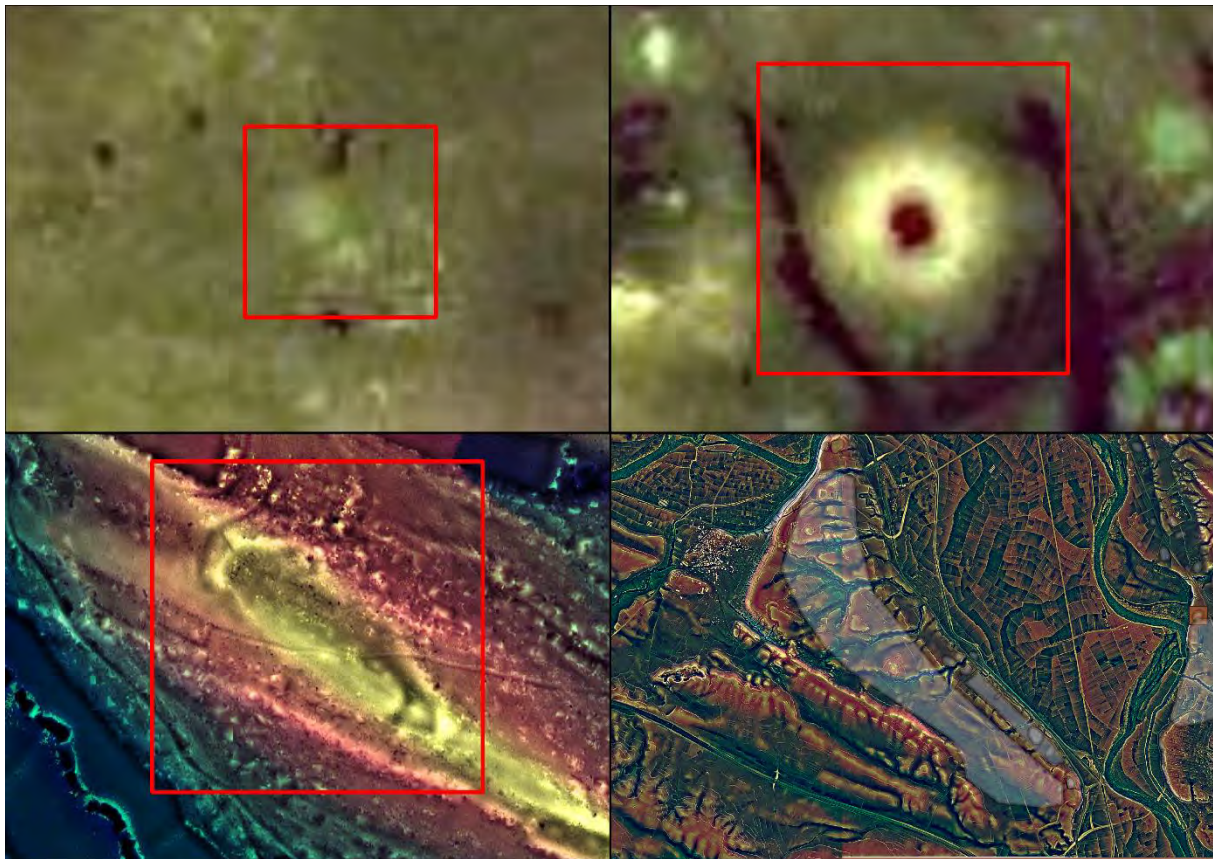


Figura 12. Composición de cuatro imágenes de yacimientos que presentan características topográficas que los definen como de microescala (arriba izquierda), mesoescala (arriba derecha); macroescala (abajo izquierda) y fuera de rango (abajo derecha). Elaboración propia.

Se utiliza el producto *Local Dominance* [22] creado con la aplicación **Relief Visualización**

El procedimiento **Local Dominance** calcula para cada pixel del MDE lo dominante que sería un observador parado en ese punto para un área circundante a escala local. La dominancia corresponde en este caso a la inclinación media del ángulo con el que el quien observa mira hacia la superficie circundante [22]. Para ejecutar el cálculo es necesario pasar al algoritmo un radio máximo, en este caso 20 píxeles (10 m) y un radio mínimo, en este caso 10 píxeles (5 m), para eliminar el ruido de las características topográficas de la superficie en la microescala, mientras que la elevación de quien observa corresponde a la elevación de cada pixel del MDE.

Esta metodología es muy idónea para representar características topográficas de elementos arqueológicos que tengan este tipo de sección [22] (Figura 13).

2.2.1.3.- El producto de Orientación

Otro de los productos que se utiliza es la orientación. Esta corresponde a la dirección de la línea de máxima pendiente y se expresa como el ángulo medido en dirección horaria desde el norte (acimut) que se calcula para cada celda de un MDE a partir de aplicar la siguiente ecuación 2.2. [69].

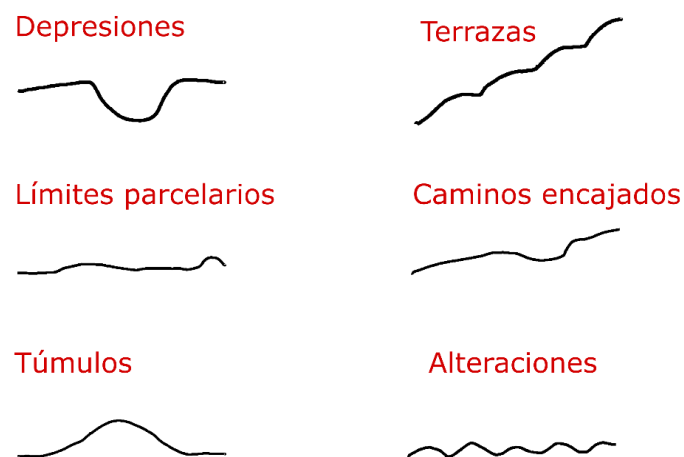


Figura 13. Representación gráfica de anomalías topográficas de origen arqueológico que quedan resaltadas mediante el uso de las técnicas de visualización descritas en [22]

Ecuación 2.2.

$$\varphi = 180^{\circ} - \arctan\left(\frac{q}{p}\right) + 90^{\circ}\left(\frac{p}{|p|}\right)$$

Siendo φ la orientación, \arctan la función inversa de la tangente de las derivadas parciales q y p en las dos direcciones x e y .

Se trata, junto con el cálculo de la pendiente, de uno de los parámetros geométricos más básicos que parten de las derivadas parciales de primer orden de la superficie, y se obtiene a partir del gradiente de la superficie. Dado un campo escalar tal como el representado por el MDE, el gradiente es un campo vectorial orientado en la dirección en la que el campo escalar experimenta una mayor variación. Las dos propiedades principales que podemos obtener del gradiente son dos: su longitud y su dirección. Estas propiedades, considerando la interpretación geomorfométrica que se le da por calcularse a partir del MDE, constituyen dos de los parámetros más importantes que pueden extraerse de este: la pendiente y la orientación [70]. La pendiente como la variación máxima de la elevación y la orientación como la línea de máxima pendiente. La aplicación práctica de esta metodología se ha ejecutado mediante el algoritmo **Aspect** integrado en la herramienta **WhiteboxTools**⁶ creada por Lindsay [62].

2.2.1.4.- Creación de diferentes modos de visualización

Los productos principales que se han expuesto se deben interpretar como firmas espectrales de las áreas de interés que se van a analizar, es decir, de los yacimientos arqueológicos. No obstante, como ya se ha reseñado en varias ocasiones, en el ámbito de estudio del reconocimiento arqueológico del territorio, es muy importante la interpretación

⁶ https://www.whiteboxgeo.com/manual/wbt_book/intro.html

visual [22] y, por tanto, cobran especial relevancia los métodos de visualización (Tabla 2) de las firmas espectrales.

En este estudio se utilizan las siguientes combinaciones⁷:

Tabla 2 Relación de los diferentes modos de visualización creados a partir de los cuales se han extraído valores estadísticos para entrenar los modelos

CÓDIGO	NOMBRE	OBSERVACIONES
1	MSTPI - B1	Valores de la banda roja (macroescala) de la imagen de multiescala.
2	MSTPI - B2	Valores de la banda verde (mesoescala) de la imagen de multiescala.
3	MSTPI - B3	Valores de la banda azul (microescala) de la imagen de multiescala.
4	ORIENTACION	Valores de orientación, sin normalizar.
5	MSTPI + LD - B1	Valores de la banda roja (macroescala) de la imagen de multiescala fusionada con la imagen de Local Dominance.
6	MSTPI + LD - B2	Valores de la banda verde (mesoescala) de la imagen de multiescala fusionada con la imagen de Local Dominance.
7	MSTPI + LD - B3	Valores de la banda azul (microescala) de la imagen de multiescala fusionada con la imagen de Local Dominance.
8	LD	Valores de Local Dominance sin normalizar.
9	DEV local	Imagen en rango 0-255 cuyos valores se han normalizado entre -3 (0) y 3 (255). Valores de MICROESCALA.
10	DEV meso	Imagen en rango 0-255 cuyos valores se han normalizado entre -3 (0) y 3 (255). Valores de MESOESCALA.
11	DEV macro	Imagen en rango 0-255 cuyos valores se han normalizado entre -3 (0) y 3 (255). Valores de MACROESCALA.

2.2.2.- Selección de yacimientos arqueológicos del IAN.

Como se ha comentado el acceso al inventario arqueológico de navarra se realiza mediante un visor cartográfico *on line* y de registro supervisado, se trata del visor SIGIAN.

Para la realización de este trabajo se ha asignado a quien suscribe este documento un usuario de tipo catalogador con acceso al 100% de los registros del inventario. No obstante, SIGIAN está concebido como una herramienta para gestionar el inventario arqueológico, por tanto, no está pensada como una herramienta para análisis ni investigaciones.

⁷ El índice correlativo que figura es el que se utilizara en la tabla de atributos del *dataset* para identificar de qué tipo de imagen o combinación de ellas proviene la información estadística.

2.2.2.1.- Obtención de la información

Desde SIGIAN se pueden descargar dos tipos de datos: los yacimientos en formato vectorial (puntos o polígonos) y las fichas de inventario.

Para este trabajo se han descargado todos los registros asociados a cada una de las entidades locales de Navarra que figuran en el SIGIAN, es decir, municipios y facerías. Los concejos no figuran como unidades de clasificación, ya que todos se engloban en un municipio, mientras que las facerías⁸ suelen coincidir con límites de varios municipios, por tanto, hay que entenderlas como entidades locales a efectos de prospección arqueológica.

Estos archivos se descargan en formato vectorial de tipo .kml al que se asocian un conjunto de valores de la ficha de inventario.

No obstante, las fichas de inventario completas no están volcadas sobre los datos vectoriales. Para poder trabajar con este conjunto de información se ha procedido a descargar ficha a ficha todos los registros de la comunidad foral en formato .html.

Una vez obtenidas todas las fichas, estas han sido leídas automáticamente mediante la librería de python *Beautiful Soup*⁹; esta librería está compuesta de un conjunto de clases y funciones que permiten entender la estructura gramatical de los archivos con formato .html o .xml y almacenar su información en objetos que se volcaran en listas de Python. Finalmente, se estructuran como una tabla en formato tipo .csv con la librería para python Pandas¹⁰.

Para adquirir la información ha sido necesario crear tantas listas como campos se requieran vincular a los datos vectoriales (nombre, localización, entorno, tipología, etc.). Mediante un iterador de tipo *for* se leerán las posiciones de los datos del .html y se añadirán a su correspondiente lista. Una vez ejecutado, con la librería pandas se creará un diccionario python (lista de listas) en la que se diseñará una especie de base datos con campos y valores (lista de información de los .html) que finalmente será transformada en un archivo de datos tabular en formato .csv.

Obtenido el archivo .csv con la información categórica, ésta se ha unido a la información vectorial para gestionarla desde un entorno GIS, en este caso QGIS 3.10 A Coruña.

2.2.2.2.- Información disponible

La información contenida en la ficha de inventario es un reflejo de la propia estructura de la base de datos. No obstante, no es objeto de este trabajo el comprender y analizar la estructura de la base de datos sino conocer de forma descriptiva que es lo que contiene y cómo la información contenida se puede explotar de manera analítica.

La extracción de datos de SIGIAN se concluyó en marzo de 2021, por tanto, se incluyen todos los datos existentes en el inventario hasta esa fecha. Cabe recordar que el inventario se

8 **Según el Diccionario Panhispánico del español jurídico: Civ.** En Navarra, servidumbre recíproca entre varias fincas de propiedad pública o privada. En las facerías los ganados podrán pastar de sol a sol en el término facero, pero no podrán acercarse a los terrenos sembrados o con frutos pendientes de recolección. <https://dpej.rae.es/lema/facer%C3%ADa> Consultado en 21 de agosto de 2021.

9 <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

10 <https://pandas.pydata.org/>

alimenta constantemente, ya sea por las empresas que han resultado adjudicatarias de un lote de prospección, por equipos de investigación que plasman sus descubrimientos, o por la propia administración que revisa y gestiona el inventario.

Las fichas del inventario arqueológico de navarra se dividen en tres bloques: geometría, adjuntar archivos y tabla de atributos.

El bloque de **edición de geometrías** permite habilitar la edición vectorial de un polígono o desplazar un punto. Las coordenadas se adquieren en proyección UTM ETRS89 huso 30 N (EPSG:25830) que se reflejan numéricamente en las fichas del inventario.

En cuanto a la opción de **adjuntar archivos** se pueden agregar archivos tipo imagen (jpeg, .png y .gif) o bien archivos vectoriales en formato .gml o .kml.

La información contenida en la **tabla de atributos** se divide en nueve apartados: identificación (Figura 14), localización (Figura 14), general (Figura 14), tipología (Figura 15), descripción (Figura 15), actuaciones (Figura 16), conservación (Figura 16), materiales (Figura 17), situación legal (Figura 16) y bibliografía (Figura 16). En los siguientes diagramas se resumen las variables categóricas que pueden integrarse en el inventario.

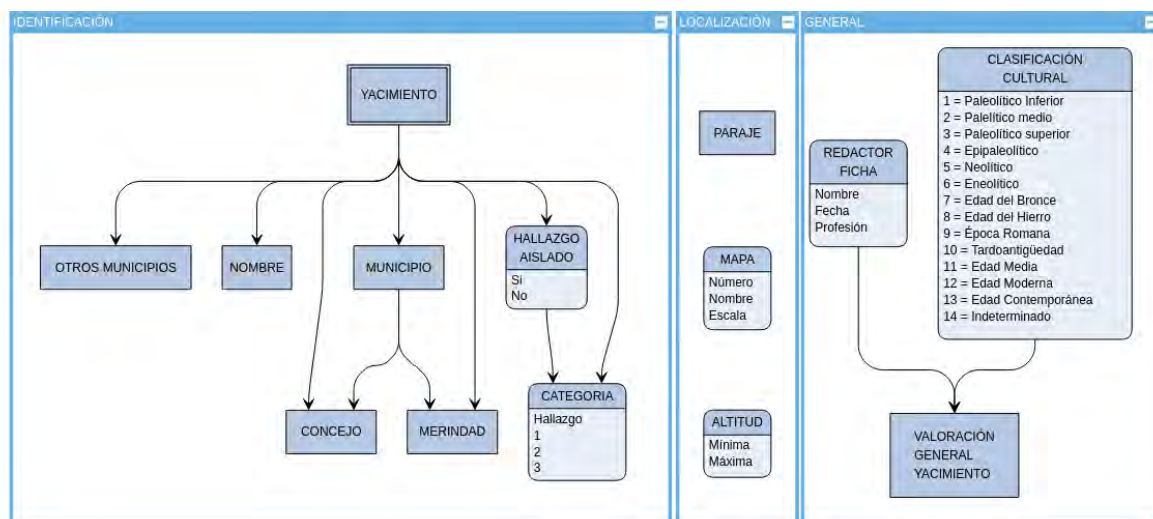


Figura 14. Diagrama conceptual de la base de datos del inventario arqueológico de Navarra. Elaboración propia.

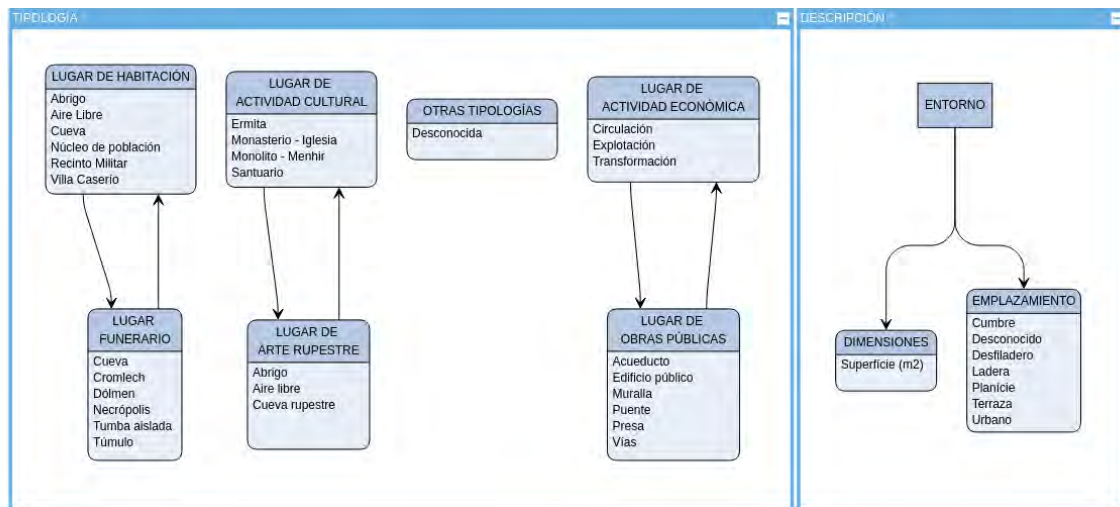


Figura 15. Diagrama conceptual de la base de datos del inventario arqueológico de Navarra. Elaboración propia.

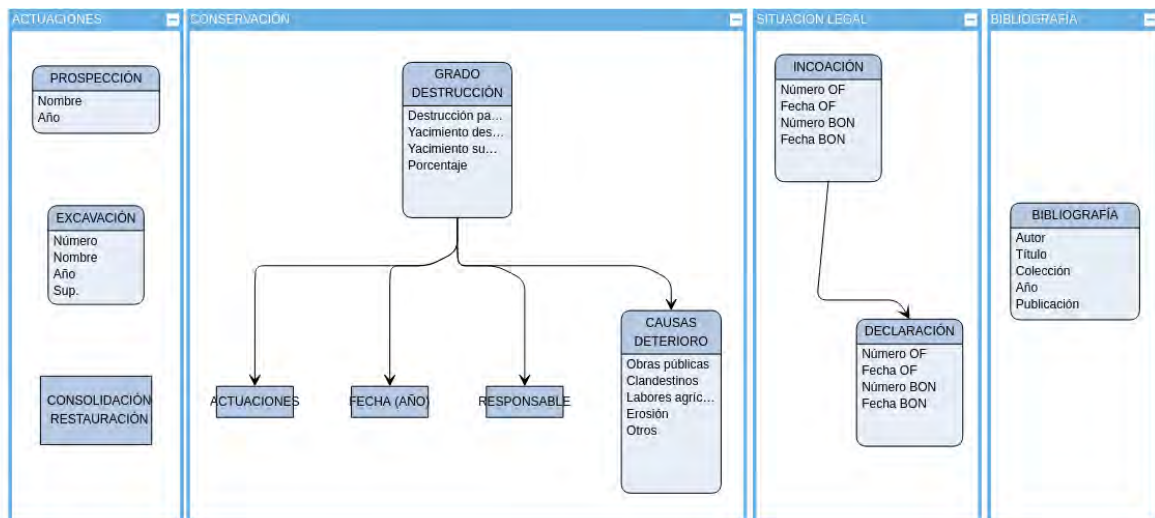


Figura 16. Diagrama conceptual de la base de datos del inventario arqueológico de Navarra. Elaboración propia.

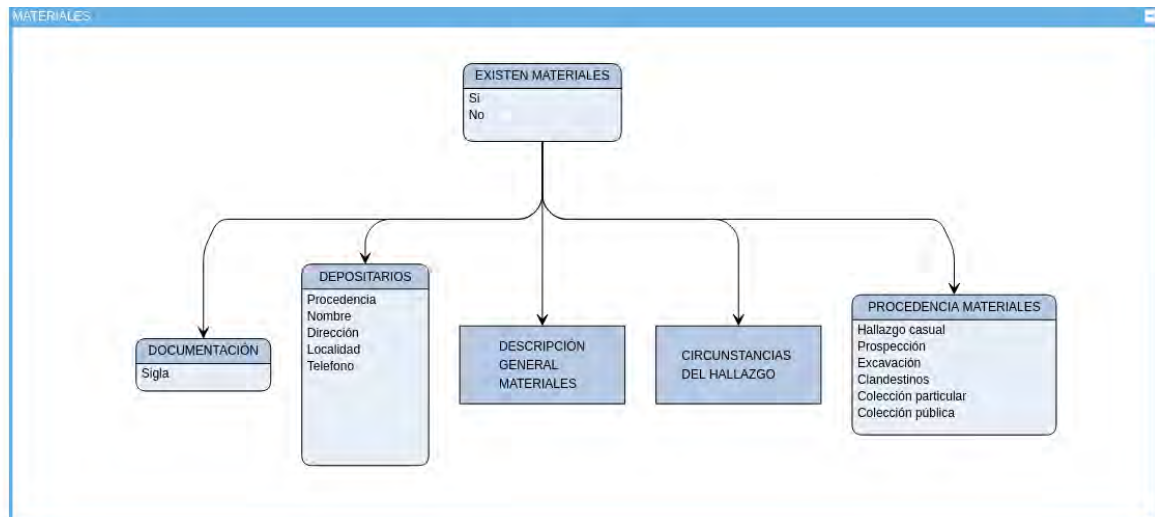


Figura 17. Diagrama conceptual de la base de datos del inventario arqueológico de Navarra. Elaboración propia.

2.2.2.3.- Información utilizada

Las áreas de interés o ventanas utilizadas tienen su origen en un conjunto de archivos vectoriales, ya sea en forma de polígono o punto, a los que se les ha vinculado la información categórica mediante el uso del código de yacimiento como punto de unión. Este código está compuesto de un valor alfanumérico que se expresa de la siguiente manera 09-31-006-0003 o país-región-municipio-yacimiento.

Los atributos principales que se han vinculado permiten identificar espacialmente el yacimiento, cuál es su nombre, quien lo ha identificado, cuál es su tipología, su cronología y que superficie tiene. El resto de información no se ha considerado útil para el propósito de esta investigación. Sin embargo, en el caso de ser necesario acceder a la misma se puede hacer mediante el uso del código de yacimiento, ya que este actúa como clave secundaria del Inventario Arqueológico de Navarra.

El esquema (Figura 18) de la información utilizada es el siguiente:

- Información vectorial:
 - Polígonos
 - Puntos
- Información categórica:
- Identificación:
 - Código de yacimiento
 - Nombre
 - Municipio
- General:
 - Redactor ficha

- Nombre
- Clasificación cultural (cronología)
- Tipología: toda la información
- Descripción:
 - Superficie

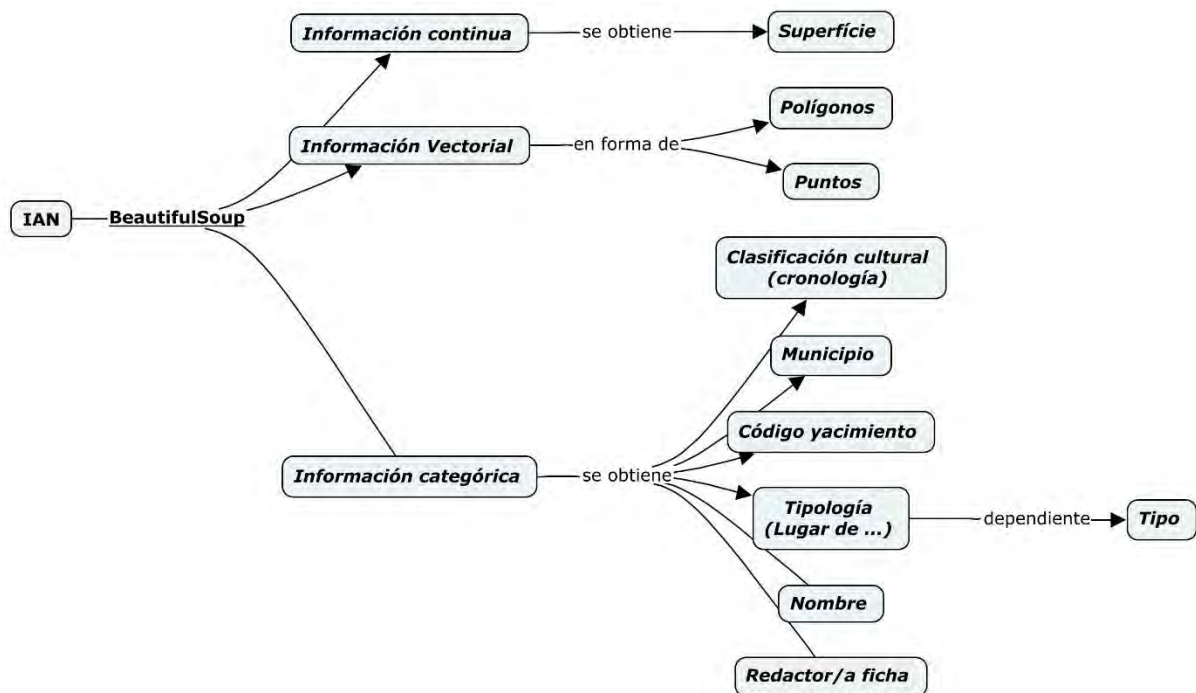


Figura 18. Diagrama de flujo del proceso de adquisición de datos del Inventario Arqueológico de Navarra.

2.2.3.- Preprocesamiento

Como se ha definido anteriormente el concepto de yacimiento es algo que se concibe como no homogéneo, estando su representación espacial condicionada a la percepción y conocimientos de quien la dibuja (edición de vectores en el SIGIAN) y, a su vez, la percepción actual del yacimiento es un producto de la evolución temporal del mismo donde inciden factores endógenos (generación del registro arqueológico) y exógenos (alteraciones naturales relacionadas con la geomorfología del enclave).

Por tanto, si se parte del principio de que las representaciones de los yacimientos albergados en el inventario arqueológico de Navarra son arbitrarias y no homogéneas. ¿Cómo convertir la base de datos en objetiva y homogénea para poder ser segmentada y clasificada automáticamente?

2.2.3.1.- Limpieza previa

Los registros extraídos de la base de datos del Inventario Arqueológico de Navarra son 9.869.

De estos se puede afirmar que existen 388 registros duplicados, los cuales se han eliminado, quedando 9.481 registros. Estos registros representan la suma de 7.773 yacimientos arqueológicos y 1.708 hallazgos aislados.

Para esta investigación sólo se tienen en cuenta los valores de yacimientos arqueológicos.

2.2.3.2.- Resolviendo la heterogeneidad de la base de datos

Partiendo de la premisa de que la signatura espectral que se utiliza es de carácter topográfico (valor DEVmax) lo que se realiza es dividir la base de datos en dos conjuntos:

- uno que agrupe los yacimientos con anomalías topográficas visibles en los diferentes modos de visualización
- y otro con el resto de los yacimientos del IAN sin anomalías topográficas visibles.

Este primer problema sólo se puede resolver manualmente realizando una inspección visual de todos los yacimientos sobre las imágenes que combinan la información MSTPI y LD (Figura 20).

La base de datos ha quedado dividida en 1.787 yacimientos seleccionados con algún tipo de anomalía topográfica visible y 5.933 sin anomalía. Hay que destacar que en el proceso de revisión manual de los yacimientos se han eliminado 53 registros que correspondían a elementos arqueológicos dentro de ciudades o que han sido afectados por grandes obras públicas (Autovía A-12, Canal de Navarra o TAV) y, actualmente, no existen físicamente como yacimientos, aunque sí administrativamente.

Una vez separados, se ha procedido a realizar una reducción de la dimensionalidad de los datos asociados a yacimientos con anomalías topográficas. El procedimiento utilizado se conoce como **Self-Organizing Map (SOM)**, se trata de un tipo de red neuronal artificial que se entrena mediante aprendizaje no supervisado¹¹ para producir una representación bidimensional y discreta de los datos a partir de utilizar la función de vecindad para preservar las propiedades topológicas del espacio de entrada. Este tipo de análisis permite descubrir la correlación entre los datos.

El procedimiento fue introducido por Teuvo Kohonen en 1990 [71] el cual desarrolló una arquitectura basada en dos capas: la de entrada y la de salida, esta última entendida como un mapa de características (Figura 18).

Esta arquitectura no utiliza el descenso del gradiente para actualizar los pesos de las neuronas, como es habitual en las redes neuronales, sino que le pasa directamente los pesos a la capa de salida. A cada neurona en un SOM se le asigna un vector de peso con la misma dimensionalidad D que el del espacio de entrada, el cual se actualiza mediante el proceso de aprendizaje competitivo. El aprendizaje competitivo se fundamenta en tres procesos: la competición, la cooperación y la adaptación.

¹¹ Los algoritmos de clasificación no supervisada no requieren de un conocimiento previo de cómo se comportan las categorías o clases a discriminar. Se basa en procedimientos estadísticos que valores con similar comportamiento, asignado correspondencias entre grupos de valores y categorías.

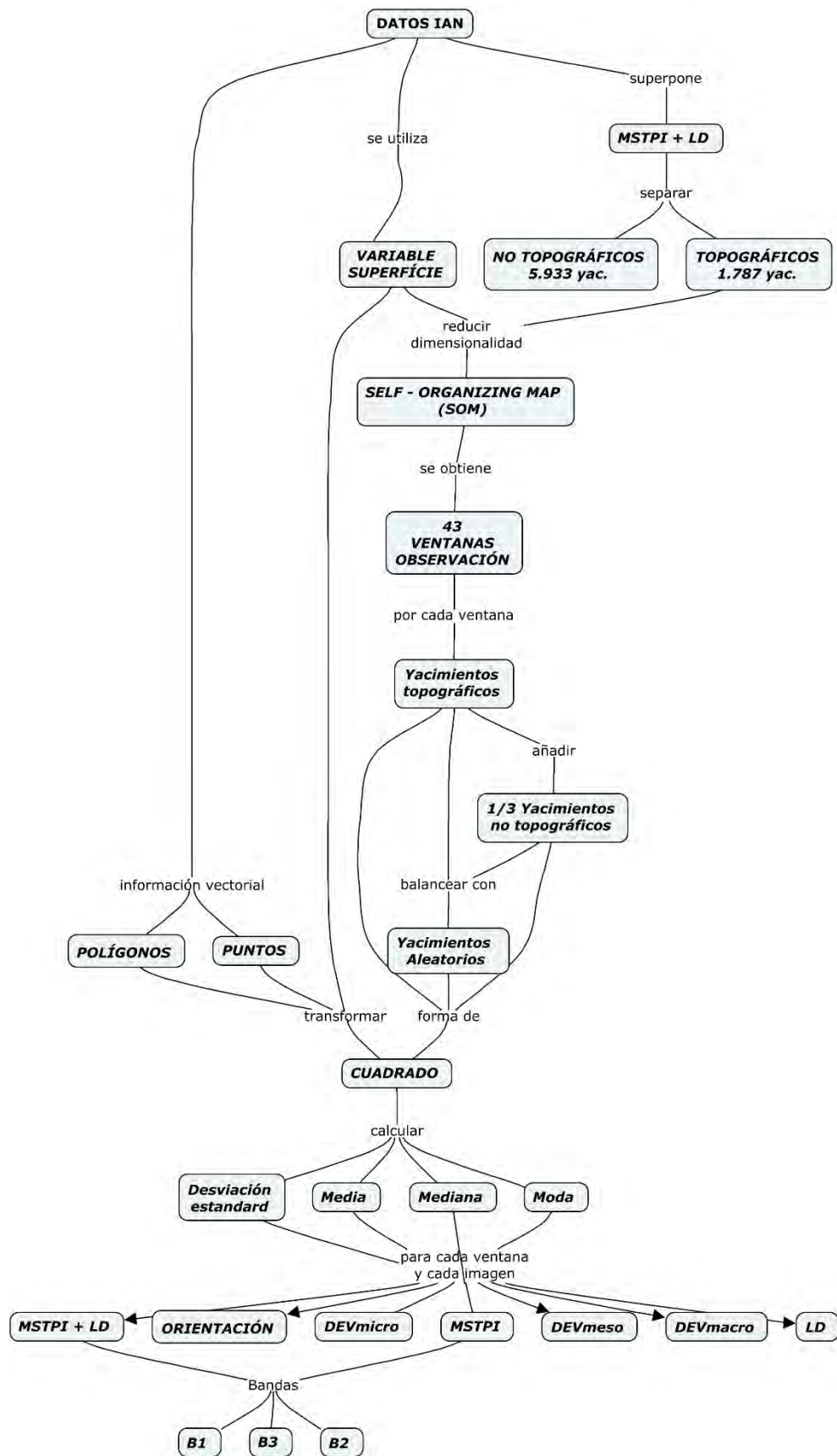


Figura 19. Diagrama de flujo del preprocesamiento de los datos y montaje del dataset.



Figura 20. Representación de cuatro yacimientos arqueológicos del municipio de Beire (Navarra) sobre una imagen de multiescala realzada con información de dominancia local. Se observa como los yacimientos de San Julián y El Cerco son entornos topográficamente separables, mientras que los yacimientos de Cardete II y Cardete III ocupan zona llanas difícilmente separables por criterios topográficos. Elaboración propia.

Ecuación 2.3

$$i(x^n) = \arg \min_j \|x^n - w_j\|$$

Donde x^n corresponde al siguiente patrón de entrada de la base de datos. Y w_j corresponde a la unidad que mejor encaja con x^n

Durante el proceso de competición se calcula la distancia entre cada neurona de la capa de salida y los datos de entrada, siendo elegida (ganadora de la competición) la que tenga una distancia euclidiana más baja. El proceso de cooperación se fundamenta en no sólo elegir la neurona ganadora sino también a sus vecinas mediante la función de vecindad, es decir, serán vecinas aquellas que según las variables distancia y tiempo estén más cerca de la neurona ganadora del proceso competitivo.

Elegidas las neuronas, se actualizan los pesos, pero no de manera lineal, sino en función a la distancia entre la neurona y los datos de entrada, es decir, a mayor distancia menor ajuste. [62]. Este proceso finaliza cuando la tasa de aprendizaje converge en cero y ya no hay más actualizaciones para la neurona ganadora.

Ecuación 2.4

$$w_k = w_k + \eta(t) \cdot h_{ik}(t) \cdot (x^{(n)} - w_k)$$

Ecuación por la que se actualizan los pesos de la neurona ganadora w_i y los de todos sus vecinos w_k .

Para la aplicación de este procedimiento se ha utilizado el software **Orange Data Mining 3.29**, que se trata de una aplicación de ciencia de datos desarrollada por el Laboratorio de Bioinformática de la Universidad de Ljubljana (Eslovenia) siendo un proyecto de código abierto basado en Python [73].

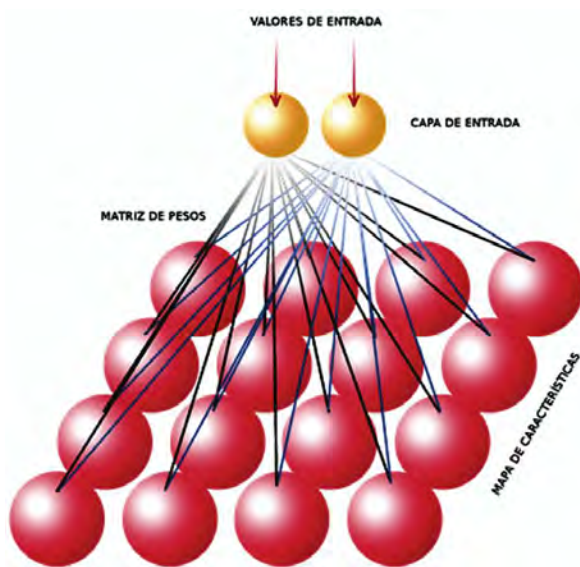


Figura 21. Estructura de un mapa de características de una red SOM en las neuronas adyacentes a la capa de salida están conectadas entre sí por una relación de vecindad. Fuente [72]

A partir de esta aplicación se ha configurado una red SOM inicializada con valores preprocesados mediante un análisis de componentes principales (PCA) y una asignación de dimensionalidad automática sobre una malla cuadrangular. Los valores sobre los que se calculan los vectores de peso se basan en la variable de superficie obtenida de los datos vectoriales que previamente se habían clasificado como yacimientos arqueológicos con anomalía topográfica.

Del proceso de reducción de la dimensionalidad se han obtenido 43 grupos de yacimientos similares por su superficie (Tabla 3), es decir, se ha podido homogenizar la base de datos en base a un variable continua distinta a las variables categóricas que utiliza el inventario.

Tabla 3. Relación del número de ventanas diferenciadas y el rango en metros cuadrados que representan.

VENTANA	RANGO (m2)	VENTANA	RANGO (m2)
G1	1047 - 1586	G23	9909 - 10577
G2	547 - 569	G24	4018 - 6577
G3	590 - 667	G25	8537 - 9725
G4	802 - 921	G26	15983 - 16875

Tabla 3 (continuación) Relación del número de ventanas diferenciadas y el rango en metros cuadrados que representan.

VENTANA	RANGO (m2)	VENTANA	RANGO (m2)
G5	939 - 1036	G27	12971 - 13837
G6	457 - 520	G28	11723 - 12923
G7	397 - 454	G29	17051 - 18032
G8	237 - 318	G30	18105 - 19231
G9	31 - 138	G31	13988 - 15837
G10	142 - 234	G32	28269 - 30481
G11	319 - 451	G33	22263 - 24333
G12	675 - 794	G34	19541 - 22100
G13	4724 - 5118	G35	24493 - 27933
G14	1677 - 2059	G36	64264 - 72269
G15	1143 - 1648	G37	46772 - 51510
G16	2088 - 2458	G38	41324 - 46543
G17	3226 - 3950	G39	30684 - 37118
G18	2326 - 3129	G40	37202 - 41197
G19	5189 - 5899	G41	56448 - 63121
G20	6603 - 7550	G42	51638 - 19558
G21	7564 - 8433	G43	202448 - 392313
G22	10612 - 11429		

2.2.3.3.- Resolviendo la arbitrariedad de la base de datos

El problema de la arbitrariedad de la base de datos se expresa gráficamente en una multiplicidad de formas geométricas para yacimientos que corresponden a las mismas variables categóricas (Figura 22).

Esta arbitrariedad se resuelve unificando las geometrías de tipo polígono y tipo punto hacia una nueva geometría: el cuadrado, al igual que se ha aplicado en el estudio de referencia [51].

Así bien, se utiliza el cuadrado porque este análisis se basa en ventanas de observación sobre píxeles cuadrados. El tamaño del cuadrado corresponderá al valor máximo de superficie de cada uno de los 43 grupos que se han separado anteriormente.

Para realizar esta transformación es necesario utilizar la herramienta de geoprocésamiento de QGIS 3.1x conocida como área de influencia o buffer. El proceso se basa en extraer inicialmente los centroides de cada una de las geometrías de los yacimientos a utilizar. Una vez conocidos todos los centroides, se aplicará sobre los mismos un geoprocésamiento de tipo buffer por el cual se crea un área de influencia basada en el valor máximo de cada uno de los 43 grupos en los que se ha dividido la base de datos. Para ello será muy importante que se determine que la forma de esta área de influencia sea cuadrada. Por tanto, una vez finalizado este proceso lo que se obtiene es un conjunto de 43 tipos de ventanas de observación de forma cuadrada.

2.2.4.- Montaje del dataset

Hasta el momento lo que se ha obtenido es un conjunto de 1.787 yacimientos con anomalías topográficas divididos en 43 grupos. Cada uno de estos grupos está representado espacialmente como un cuadrado cuya área corresponde al valor máximo de superficie del grupo al que corresponda, y cada una de estas formas está asociada a las diferentes variables categóricas obtenidas del inventario arqueológico de Navarra.

Para garantizar un buen aprendizaje de los modelos, cada uno de los grupos se ha constituido como un *dataset* binario y balanceado, es decir, 50% de yacimientos de la clase positiva, y 50% de yacimientos de la clase negativa.

La clase positiva ha sido constituida con un mínimo de 25 yacimientos de anomalía topográfica, más un tercio de los yacimientos del mismo rango de superficie, pero que no tienen anomalía topográfica, elegidos aleatoriamente del conjunto de yacimientos sin anomalía topográfica. Con este aporte de información extra se garantiza un mejor aprendizaje del modelo, ya que debe diferenciar dentro de la clase positiva aquello que es relevante para ser considerado un yacimiento con anomalía topográfica.

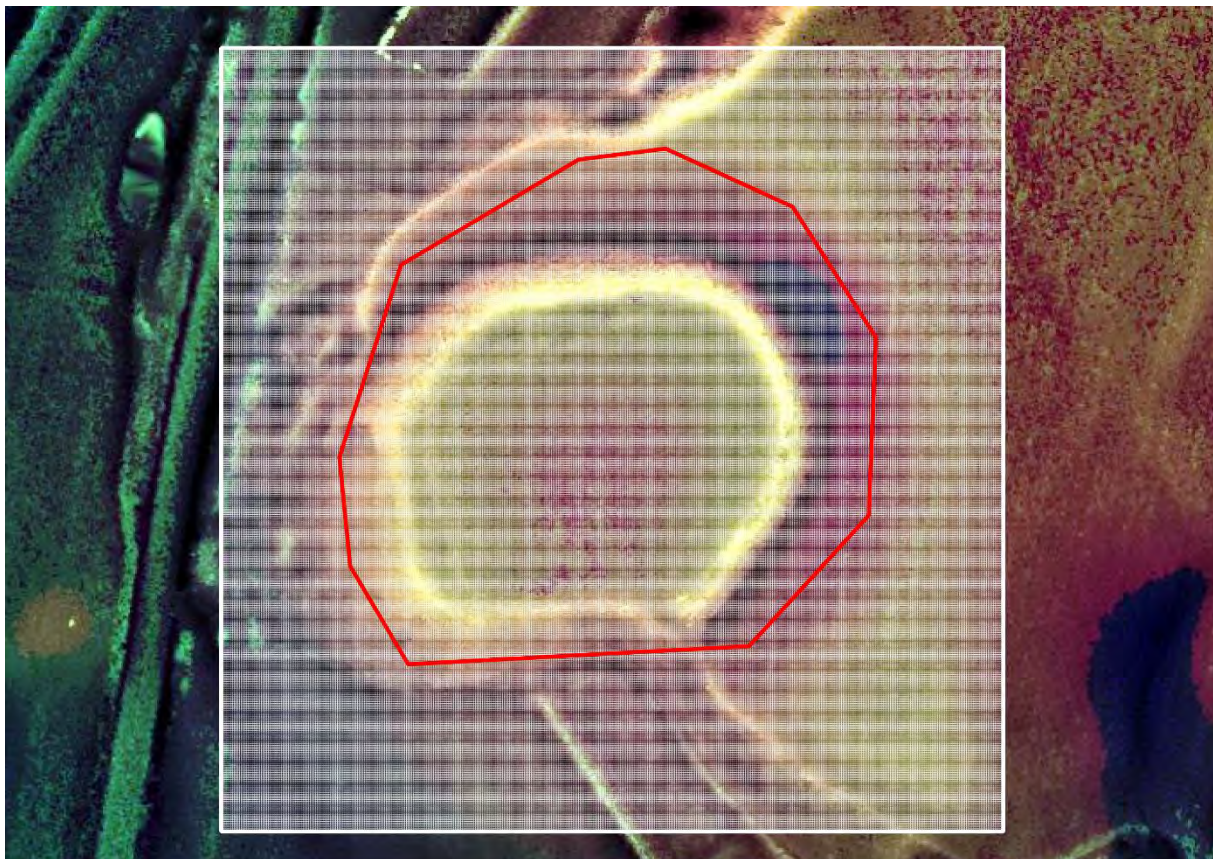


Figura 22. Ejemplo del yacimiento El Cerco (Beire) que corresponde a uno de los enclaves de la ventana G38. En rojo la delimitación del IAN, en blanco la dimensión de la ventana y la representación de los píxeles asociados a la ventana. Se observa como las ventanas regulares recogen mejor la forma de los enclaves y de la información estadística que se recoge de cada píxel.

La clase negativa se ha constituido de manera totalmente aleatoria utilizando el algoritmo nativo de QGIS 3.1x *Random points in polygons* que crea un conjunto de datos aleatorios según una delimitación espacial dada. En este caso los límites de la comunidad foral. Cada

uno de estos puntos se ha transformado en un cuadrado mediante el procedimiento descrito anteriormente para geoproceso buffer, asignándole el área correspondiente al grupo que perteneciese.

Por tanto, un grupo con 25 yacimientos topográficamente significativos tendrá 8 yacimientos más del mismo rango de superficie, pero sin anomalía topográfica. A los que se sumaran 33 yacimientos completamente aleatorios, es decir, que la unidad mínima de entrenamiento será un grupo de 66 registros, completamente balanceados, con una misma área y forma. Reseñar que en el estudio de referencia la unidad mínima de entrenamiento corresponde a un *dataset* no balanceado compuesto por 50 registros (88% aleatorios y 12% yacimientos topográficamente visibles).

Finalmente, a estos registros se les asigna un conjunto de variables continuas obtenidas de extraer las estadísticas de zona (media, mediana, moda y desviación estándar) sobre los diferentes tipos de imágenes que se ha aludido anteriormente. La extracción de estas estadísticas se ha realizado mediante el algoritmo de estadísticas de zona de QGIS 3.10 que permite crear un atributo para cada estadístico sobre un archivo vectorial existente. Reseñar que en el estudio de referencia los valores continuos utilizados corresponden a los valores espectrales de las imágenes DEVmax en cada una de las escalas mencionadas: macro, meso y microescala.

2.3.- Proceso de aprendizaje

Fundamentalmente, para producir un nuevo conocimiento que se pueda utilizar es necesario construir un modelo basado en los datos recopilados y preprocesados, que sea una descripción de los patrones y relaciones entre los datos con los que se puedan hacer predicciones, entender mejor los datos o explicar situaciones pasadas [53].

La resolución del problema se plantea como un problema de clasificación binario que prediga zonas positivas y zonas negativas. Para resolver este problema se utiliza la librería de python **scikit-learn 0.24.2** [74] compuesta de una serie de herramientas para el análisis predictivo de datos.

Aunque el origen de las variables que se utilizaran en el proceso se genera en imágenes, la resolución del problema se plantea como una estructura de datos tabular. De tal manera, que no es necesario crear una única imagen de toda el área de estudio, sino que se extraen los valores estadísticos necesarios de las imágenes planteadas, según las diferentes áreas de interés. Es decir, los datos de partida están contenidos en 43 tablas diferentes, una por cada grupo.

Para la validación de los sistemas de predicción se deben crear ejemplos de entrenamiento y test, por tanto, cada grupo se divide en un porcentaje del 70% para entrenamiento y el 30% restante para test. La división de la base de datos se realiza mediante la función **sklearn.model_selection.train_test_split** que devuelve la base de datos dividida en dos subconjuntos aleatorios para test y entrenamiento.

Este procedimiento se realiza para evitar el sobreajuste, es decir, para evitar aprender los parámetros de una función de predicción y probarlos con los mismos datos, ya que entonces el modelo solo repetiría lo mismo que ha aprendido.

2.3.1.- Modelos

Una vez dividida la base de datos se procede a seleccionar cual es el mejor modelo (algoritmo) para predecir nuevos datos. Para ello se han utilizado tres tipos de modelos: *Random Forest*, *Gradient Boosting* y *Regresión Logística* que se explican a continuación.

En el proceso de selección del modelo más adecuado es necesario poder configurar adecuadamente los hiperparámetros de cada modelo. Para ello se ha utilizado la función `sklearn.model_selection.GridSearchCV` que genera una búsqueda exhaustiva de candidaturas de valores de hiperparámetros especificados previamente, de tal manera que devuelve la mejor configuración.

2.3.1.1 *Random Forest*

El modelo *Random Forest*, diseñado por Leo Breiman en 2001 [75], corresponde a un algoritmo de aprendizaje supervisado basado en la combinación de diferentes variantes de un mismo clasificador: el **árbol de decisión** (Figura 21).

Un árbol de decisión es un clasificador que en función de un conjunto de atributos puede determinar a qué clase pertenece el caso objeto de estudio [76].

La estructura del modelo se compone de hojas y nodos (Figura 23). Las hojas corresponden a una categoría o clase del atributo objeto de clasificación. Los nodos corresponden a la pregunta que se les realiza a los datos a partir de la cual se toma la decisión. Después de cada pregunta los datos de esa parte del árbol se dividen entre una rama Si y otra No.

El aprendizaje finaliza cuando se cumple alguno de estos requisitos:

- Todos los ejemplos de entrenamiento pendientes pertenecen a la misma clase.
- El número de ejemplos de un nodo es menor que un umbral dado.

El número de ejemplos que se derivan a una rama es menor que un umbral dado.

Un *Random Forest* (Figura 24) es un modelo que promedia (*averaging method*) las predicciones de una gran cantidad de árboles de decisión, que se generan variando aleatoriamente varios parámetros que especifican qué datos se utilizan para entrenar el árbol y otros parámetros del árbol [77].

A nivel práctico en este método cada árbol del conjunto se construye a partir de una muestra extraída con reemplazo (*bootstrap sample*) del conjunto de entrenamiento. Al dividir cada nodo durante la construcción del árbol, la mejor división se encuentra entre todas las características de entrada, o un subconjunto aleatorio del tamaño del parámetro *max_features* que se le haya asignado.

El propósito de estas dos fuentes de aleatoriedad es disminuir la varianza del estimador. De hecho, los árboles de decisión individuales suelen presentar una gran variación y tienden a sobreajustarse. La aleatoriedad inyectada en los *Random Forest* produce árboles de decisión con errores de predicción algo desacoplados. Al tomar un promedio de esas predicciones, algunos errores pueden anularse. Los *Random Forest* logran una variación reducida al combinar diversos árboles, a veces a costa de un ligero aumento del sesgo [77]. En la práctica, la reducción de la varianza es a menudo significativa, por lo que se obtiene un modelo mejor en general. A diferencia de la publicación original [75], la implementación de *scikit-learn* combina clasificadores promediando su predicción probabilística, en lugar de dejar que cada clasificador vote por una sola clase [74].

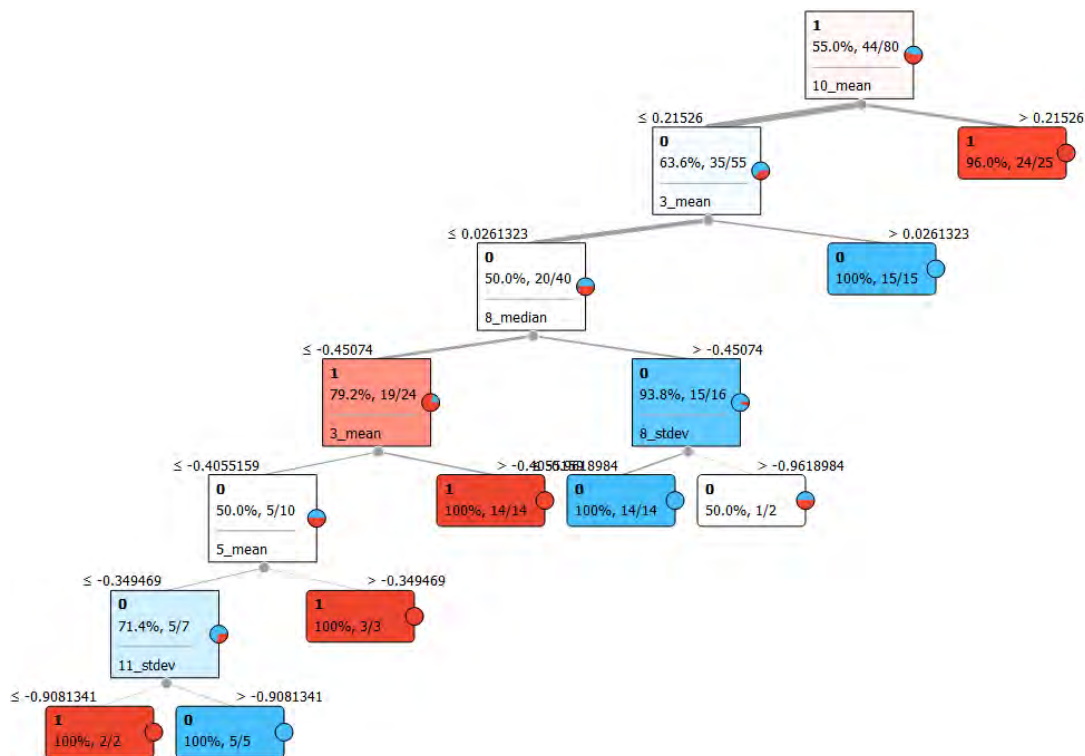


Figura 23. Esquema de funcionamiento de un árbol de decisión aplicado a los datos de la ventana de observación G6. Obsérvese como a partir de los valores dados separa los datos entre clase positiva (yacimientos) o negativa (valor aleatorio). Fuente: Elaboración propia a partir de la herramienta Tree Viewer de Orange Data Mining 3.29

Para la mejor selección de hiperparámetros se ha construido un modelo **RandomForestClassifier** con los siguientes parámetros estáticos:

- `min_samples_leaf = 1`
- `min_samples_split = 2`
- `random_state = 98`

Y se ha pasado un diccionario python con los siguientes parámetros para que la función **GridSearchCV** comprobase la mejor configuración:

- `criterion = entropy` (método C4.5), `gini` (método CART)
- `n_estimators = 10, 50, 100, 150, 200`
- `max_features = sqrt, log2, None`

2.3.1.2 Gradient Tree Boosting

Se trata de un algoritmo de aprendizaje supervisado basado en la combinación de diferentes variantes de un mismo clasificador: el **árbol de decisión**. Así bien, mientras *Random Forest* corresponde a un *ensemble* basado en un método de promedio, *Gradient Tree Boosting* (Figura 24) corresponde también a un *ensemble* basado en un método de impulso (*boosting method*), por el que se generaliza el aumento de la función de pérdida (*loss function*).

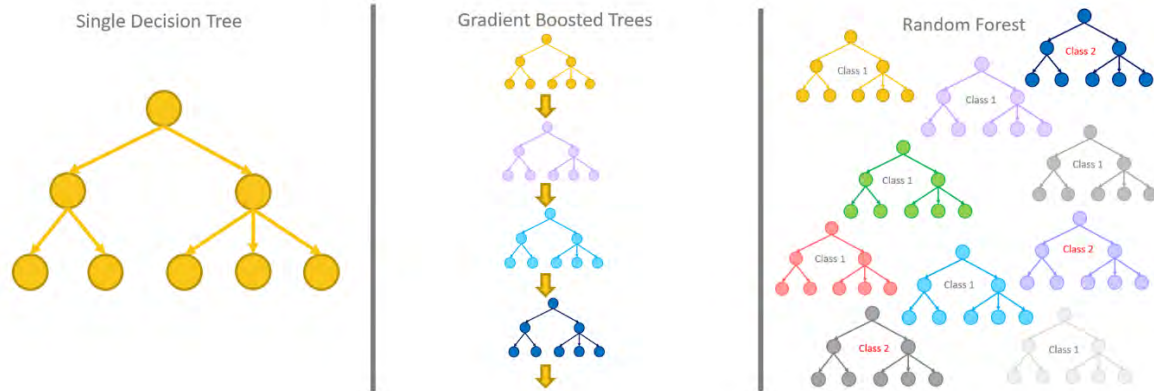


Figura 24. Representación gráfica del funcionamiento de los diferentes métodos basados en los árboles de decisión. Fuente [78]

El método *boosting* asigna pesos de manera uniforme a todos los ejemplos. Una vez evaluado el primer árbol se actualizan los pesos en función del error, aumentando la ponderación de las observaciones que son difíciles de clasificar y bajándolas en aquellas que son fáciles de clasificar. El nuevo árbol es una suma de los dos primeros sobre el que se vuelve a calcular el error de clasificación para crear un tercer árbol que combina los pesos residuales obtenidos [79]. Por tanto, las predicciones del modelo de conjunto final corresponden a la suma ponderada de las predicciones realizadas por los modelos de árbol anteriores [80]. El modelo *Gradient Boosting* se caracteriza por usar la función de pérdida (*loss function*) la cual se puede parametrizar para adaptarla a los propósitos del problema planteado.

Para la mejor selección de hiperparámetros se ha construido un modelo **GradientBoostingClassifier** con los siguientes parámetros estáticos:

- *learning_rate* = 1
- *max_depth* = 1
- *loss* = *deviance*
- *random_state* = 0

Y se ha pasado un diccionario python con los siguientes parámetros para que la función **GridSearchCV** comprobase la mejor configuración:

- *n_estimators* = 10, 50, 100, 150, 200
- *max_features* = sqrt, log2, None

2.3.1.3 Logistic Regression

La Regresión Logística, desarrollada por David Cox en 1958, es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa.

Es importante tener en cuenta que, aunque la regresión logística permite clasificar, se trata de un modelo de regresión que modela el logaritmo de la probabilidad de pertenecer a cada clase [81] a partir de una función de coste:

- Hipótesis de probabilidad en una regresión logística

Ecuación 2.5

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Función de coste en una regresión logística

Ecuación 2.6

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m l_{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))$$

Así bien, es necesario ajustar los parámetros *theta* a partir de minimizar el error de la función de coste. Este proceso se realiza mediante el algoritmo del descenso por gradiente por el que se asignan valores aleatorios o a ceros a los parámetros *theta* repitiendo el proceso hasta conseguir la convergencia.

Para la mejor selección de hiperparámetros se ha construido un modelo **LogisticRegression** con los siguientes parámetros estáticos:

- *multi_class* = *ovr* (sólo para problemas binarios)
- *random_state* = 48

Y se ha pasado un diccionario *python* con los siguientes parámetros para que la función **GridSearchCV** comprobase la mejor configuración:

- *penalty* = *l1, l2, elasticnet, none*
- *solver* = *newton-cg, lbfgs, liblinear, sag, saga*
- *max_iter* = 100, 500, 1000, 3000

2.3.2.- Rendimiento

La evaluación de los resultados es el proceso por el que se comprueba si existe un aprendizaje y el modelo es aplicable a nuevos datos, realizándose las diferentes comprobaciones sobre el subconjunto de test que se había separado previamente.

No obstante, en el proceso de evaluación es necesario preguntarse en qué se tiene interés, si en la exactitud (en obtener un modelo que clasifique rápidamente), en que el modelo sea escalable según la dimensionalidad del conjunto de datos, o bien en que nuestro modelo sea interpretable [54].

Para este caso de estudio es muy importante la exactitud e interpretabilidad del modelo, dejando a un lado la velocidad y escalabilidad del mismo. Se considera que ajustando el aprendizaje a la mejor exactitud el modelo evitará la presencia de falsos positivos, algo que ha sido muy habitual en las investigaciones relacionadas con la detección automática de entornos arqueológicos [47]. Así bien, hay que valorar que los modelos ajustados puedan ser interpretados y conocer qué conjuntos de variables son las que aportan mayor peso a la resolución del problema [77], de tal manera que pueda reducirse la dimensionalidad de los

datos a utilizar en el momento de realizar una predicción sobre un conjunto de instancias no conocidas por el modelo.

Tabla 4. Tabla interpretativa para comprender el funcionamiento de las métricas de evaluación de un clasificador binario. Fuente [54]

		Clasificación como	
		Si	No
Clase real	Si	Verdadero Positivo (VP)	Falso Negativo (FN)
	No	Falso Positivo (FP)	Verdadero Negativo (VN)

Las principales medidas de rendimiento utilizadas se basan en la interpretación de los datos recogidos en la matriz de confusión (Tabla 4). Por definición (en un problema binario), la entrada i,j en una matriz de confusión es el número de observaciones que realmente están en el grupo i , pero que se predice que estarán en el grupo j .

Para evaluar los resultados se han utilizado las siguientes medidas de rendimiento:

- **Recall:** *True positive rate* o **Sensibilidad**. Corresponde a los ejemplos de la clase positiva clasificados correctamente.

Ecuación 2.7

$$\circ TPR = \frac{VP}{VP+FN}$$

- **Precision:** Proporción de ejemplos clasificados en la clase positiva que son realmente de la clase positiva.

Ecuación 2.8

$$\circ Precision = \frac{VP}{VP+FP}$$

- **Especificidad:** TNR. Corresponde a los ejemplos de la clase negativa clasificados correctamente.

Ecuación 2.9

$$\circ TNR = \frac{VN}{VN+FP}$$

- **F1 Score:** Media armónica entre las medidas de rendimiento *precision* y *recall*.

Ecuación 2.10

$$\circ F1\ score = 2 \frac{(Precision*TPR)}{(Precision+TPR)}$$

- **Balanced Accuracy:** Corresponde a la media aritmética de la sensibilidad y la especificidad.

Ecuación 2.11

$$\circ Balanced\ Accuracy = \frac{TPR+TNR}{2}$$

- **Media Geométrica:** Permite obtener un balance entre los porcentajes de ejemplos bien clasificados de las dos clases, aplicándose la media geométrica entre la sensibilidad y la especificidad.

Ecuación 2.12

$$\circ \quad GM = \sqrt{TPR * TNR}$$

- **Curva Precision – Recall:** Esta curva es el resultado de dibujar la gráfica entre las medidas *precision* y *recall*. La gráfica permite ver a partir de qué sensibilidad existe una degradación de la precisión y viceversa.
- **Curva ROC – AUC:** Se trata de una visualización gráfica que relaciona la sensibilidad con el ratio de falsos positivos. Es decir, relaciona la sensibilidad del modelo con los fallos optimistas (clasificar los negativos como positivos).
- **Indice Kappa:** Mide el grado de fiabilidad de la exactitud de la clasificación, prescindiendo de factores aleatorios.
 - $K = (\text{observado} - \text{esperado}) / (1 - \text{esperado})$
 - Según este estadístico los valores resultantes definen un clasificador de la siguiente manera:
 - <0,20: Pobre
 - 0,21 - 0,40: Débil
 - 0,41 - 0,60: Moderado
 - 0,61 - 0,80: Bueno
 - 0,81 - 1: Muy bueno

Para evaluar la interpretabilidad de los modelos se ha utilizado el atributo de *sklearn feature_importances_*, así bien este atributo solo es válido para estimadores basados en arboles de decisión. Esta herramienta analiza cada rama, en busca de ver qué característica se usó para dicha división y cuánto mejoró el modelo como resultado de esa división. La mejora (ponderada por el número de filas en ese grupo) se agrega a la puntuación de importancia para esa característica. Esto se suma en todas las ramas de todos los árboles y, finalmente, las puntuaciones se normalizan de manera que suman 1 [77]. El resultado se muestra en formato de gráfica de barras donde se puede valorar cuáles son las variables que más han influido en la toma de decisiones del aprendizaje del modelo, de tal manera que se pueda reajustar el mismo reduciendo la cantidad de variables para el entrenamiento del modelo.

2.4.- Ventana de análisis

En última instancia se ha procedido a poner en producción los mejores modelos aprendidos, es decir, se han obtenido nuevos datos con la misma estructura que los usados para entrenar a los que se les ha pasado los modelos devolviendo unos valores de probabilidad, a partir de los cuales crear unos mapas de probabilidad que informan sobre la presencia de anomalías arqueológicas.

El procedimiento utilizado ha consistido en crear, para cada uno de los modelos elegidos, una malla o *grid* cuyo tamaño de celda fuese igual a la ventana de observación del modelo. Esta malla se ha construido mediante el algoritmo nativo de QGIS 3.1x "Crear Cuadrícula" asignándole una forma cuadrada y una superposición del 50% entre cada una de las celdas de la malla. A partir de este archivo vectorial se han extraído las estadísticas de zona usando el mismo procedimiento descrito anteriormente para la creación del *dataset*.

Una vez creado este nuevo conjunto de datos se ha procedido a evaluar las probabilidades que dichas observaciones correspondan a anomalías arqueológicas.

Para ello se ha utilizado la función ***predict_proba*** de *scikit learn* que predice la probabilidad de pertenecer a una clase o a otra. Estas predicciones se han unido a la malla vectorial creada. A partir de aquí se ha disuelto la geometría hacia los valores de probabilidad mayores del 75%, de tal manera, que se pueda establecer una escala de color graduada que refleje las más altas probabilidades.

3.- Resultados y discusión

Se muestran a continuación los principales resultados más relevantes para el objeto de la investigación, si bien el grueso de los resultados se muestra organizado en los Anexos I, II y III.

El **Anexo I** se compone de nueve cuadernos donde en cada uno de ellos figura el código para realizar las correspondientes tareas:

Cuaderno 1: crear los valores de multiescala.

Cuaderno 2: crear los valores de orientación.

Cuaderno 3: extraer los valores del Inventario Arqueológico de Navarra.

Cuaderno 4: localizar la mejor configuración de los diferentes clasificadores sobre todas las ventanas de observación.

Cuaderno 5: ejecutar individualmente la mejora configuración de un clasificador tipo *Random Forest* para todas las ventanas de observación.

Cuaderno 6: ejecutar individualmente la mejora configuración de un clasificador tipo *Gradient Boosting* para todas las ventanas de observación.

Cuaderno 7: ejecutar individualmente la mejora configuración de un clasificador tipo *Logistic Regresion* para todas las ventanas de observación.

Cuaderno 8: aplicar de los valores de los mejores modelos sobre la ventana de análisis y predecir las probabilidades de la existencia de anomalías arqueológicas.

Cuaderno 9: comparar los mejores modelos contra todas las ventanas de observación.

Las imágenes con valores de *Local Dominance* se han obtenido ejecutando directamente la aplicación *Relief Visualization Toolbox*.

En el **Anexo II** se incluye una ficha para cada una de las 43 ventanas en las que se ha dividido el *dataset*. En las fichas figura información de dos tipos: del tipo de ventana de observación y del proceso de entrenamiento y test. La ficha incluye los siguientes campos para cada ventana: el nombre de la ventana, el rango en metros cuadrados de la ventana y tres imágenes representativas de los elementos que la componen, una imagen para los yacimientos topográficos, otra para los no topográficos y otra para las observaciones aleatorias.

Las medidas de evaluación que figuran en las fichas se han obtenido del fragmento del *dataset* obtenido para testear.

En cada ficha (Figura 25) se recoge la mejor configuración del clasificador obtenido por el procedimiento de *GridSearchCV*. Se incluyen los valores de media geométrica, de la curva *ROC AUC*, del modelo *AUC PR*, de *F1 score* y el índice *Kappa*.

El apartado gráfico incluye información visual de la matriz de confusión, de los modelos *ROC AUC* y *AUC PR* y una gráfica de barras en la que se determina cual ha sido la variable más importante en el proceso de entrenamiento.

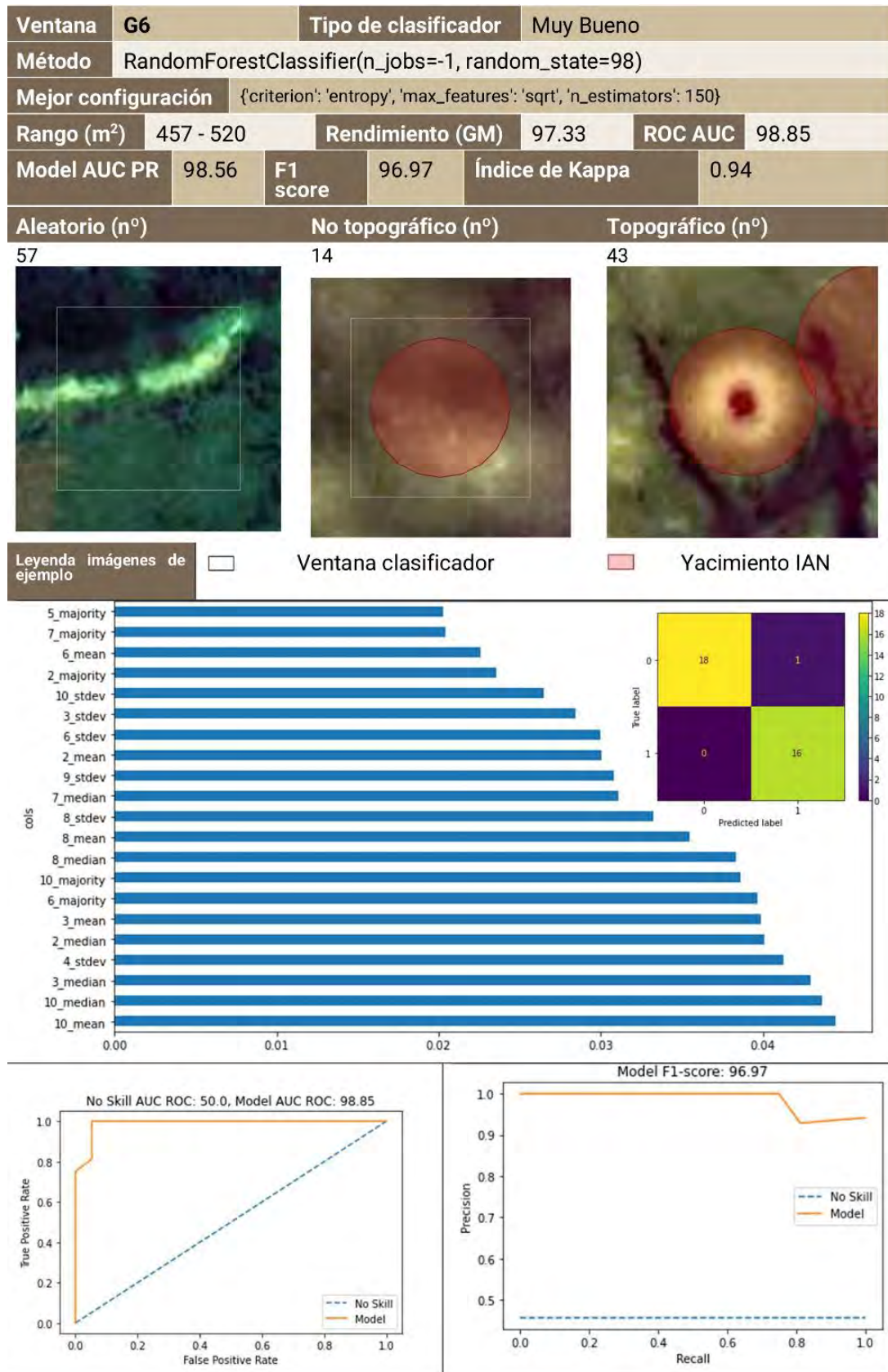


Figura 25. Modelo de ficha en el que se observan los datos de la ventana G6, que ha resultado ser un excelente clasificador. Elaboración propia.

En el **Anexo III** se recoge la información estadística de las variables categóricas tipo (dolmen, túmulo, aire libre, núcleo de población, etc.) y tipología (lugar funerario, lugar de habitación, lugar de actividad económica, etc.) de las 43 ventanas de observación.

3.1.- Proceso de entrenamiento y evaluación de los modelos

En resumen, se puede afirmar que las mejores configuraciones obtenidas para entrenar los modelos corresponden al *ensemble Random Forest*, aunque las variaciones en su configuración están altamente condicionadas por cada *dataset*. Se ha encontrado la mejor configuración en un *Random Forest* para 36 ventanas de observación, mientras que 6 ventanas se han resuelto satisfactoriamente mediante el método *Gradiente Boosting* y en un caso la mejor configuración se ha obtenido a través de una Regresión Logística. Estos resultados coinciden con el estudio de referencia, que afirma que el clasificador tipo *Random Forest* es capaz de resolver eficientemente los problemas de clasificación binaria asociados a valores de multiescala relacionados con entornos arqueológicos [51].

En cuanto a la interpretabilidad de los datos (Figura 26 y tabla 5), se ha encontrado que los valores de media de la variable DEV meso, es decir, los registros obtenidos de la signatura espectral que informa sobre la elevación media en el rango de la mesoescala (100 a 10.000 m²) han sido los que con más frecuencia se han utilizado para discriminar las clases.

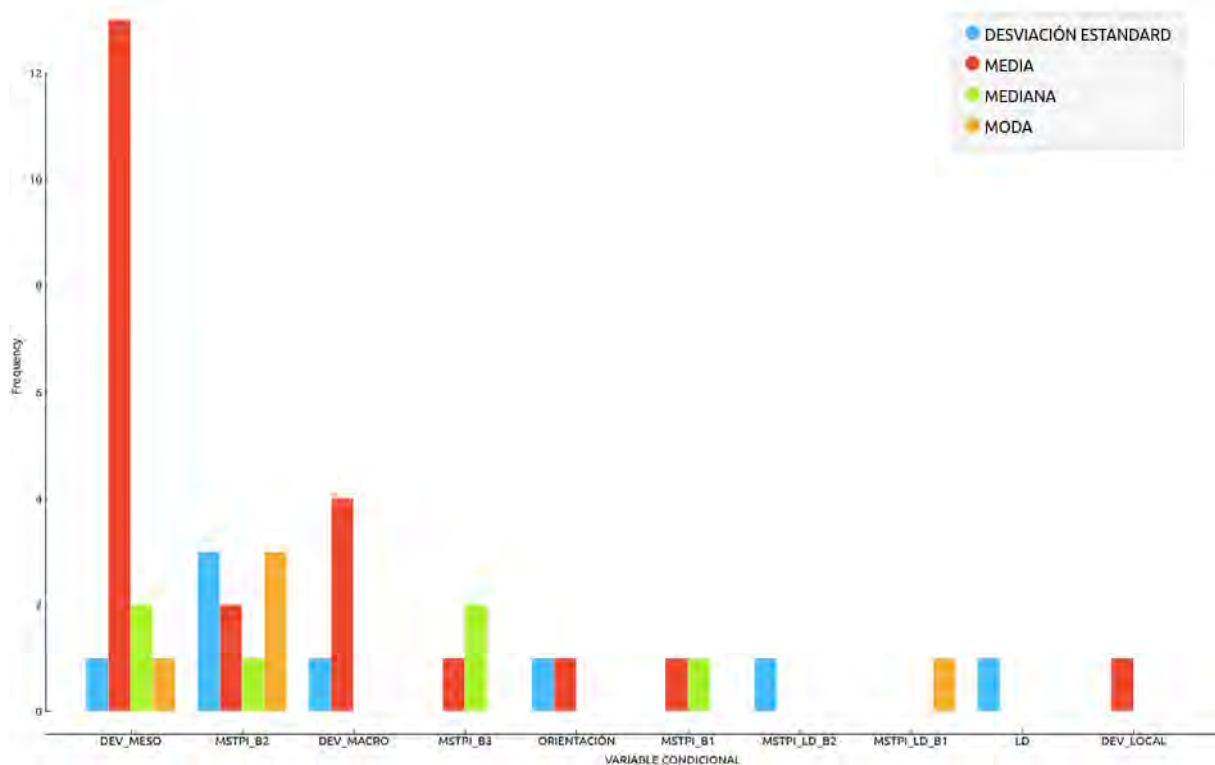


Figura 26. Gráfica de barras que relaciona la frecuencia que ha sido utilizada cada una de las variables condicionales y cuál ha sido el estadístico más relevante en cada caso. Elaboración propia.

Tabla 5. Relación de las métricas de evaluación de los diferentes modelos entrenados. Elaboración propia.

VENTANA	RANGO (m2)	Método	GM	ROC AUC	AUC PR	F1	VARIABLE CONDI-CIONAL
G1	1047 - 1586	GB	81.65	88.89	90.04	80.0	MSTPI_B2
G2	547 - 569	RF	79.33	88.11	83.21	78.26	DEV_MESO
G3	590 - 667	RF	79.21	84.64	86.15	79.01	MSTPI_B3
G4	802 - 921	RF	70.35	85.06	84.37	70.0	ORIENTACIÓN
G5	939 - 1036	GB	73.79	87.22	88.57	73.68	DEV_LOCAL
G6	457 - 520	RF	97.33	98.85	98.56	96.97	DEV_MESO
G7	397 - 454	GB	69.29	74.6	65.54	68.75	DEV_MESO
G8	237 - 318	RF	83.0	94.98	96.42	81.58	MSTPI_B3
G9	31 - 138	GB	76.03	84.62	91.09	81.58	DEV_MESO
G10	142 - 234	RF	78.83	88.04	91.12	79.45	DEV_MESO
G11	319 - 451	RF	80.81	88.61	88.9	78.05	DEV_MESO
G12	675 - 794	RF	85.84	90.95	84.4	84.85	MSTPI_B2
G13	4724 - 5118	RF	85.28	94.44	94.63	84.21	DEV_MACRO
G14	1677 - 2059	RF	49.72	68.96	71.93	58.06	MSTPI_B2
G15	1143 - 1648	GB	88.8	90.1	92.14	87.8	MSTPI_B2
G16	2088 - 2458	RF	70.16	81.54	71.7	66.67	MSTPI_B2
G17	3226 - 3950	RF	72.06	84.86	81.75	69.23	LD
G18	2326 - 3129	RF	72.06	84.38	81.74	69.23	DEV_MESO
G19	5189 - 5899	RF	60.86	79.63	86.33	63.16	DEV_MESO
G20	6603 - 7550	RF	83.21	91.61	90.28	84.62	MSTPI_B2
G21	7564 - 8433	RF	79.33	91.96	91.44	80.0	MSTPI_B2
G22	10612 - 11429	RF	70.35	84.34	78.53	70.0	MSTPI_B2
G23	9909 - 10577	RF	70.16	84.62	85.36	69.57	DEV_MESO
G24	4018 - 6577	RF	75.58	84.55	83.16	75.0	DEV_MESO
G25	8537 - 9725	RF	87.71	93.36	90.42	88.0	DEV_MESO
G26	15983 - 16875	RF	75.21	88.38	88.14	73.68	DEV_MESO
G27	12971 - 13837	RF	68.03	78.1	82.58	69.57	DEV_MESO
G28	11723 - 12923	RF	74.8	86.01	83.0	72.73	DEV_MESO
G29	17051 - 18032	RF	60.3	52.02	67.26	53.33	MSTPI_B2
G30	18105 - 19231	LR	67.42	70.0	57.27	76.92	
G31	13988 - 15837	RF	75.69	80.56	77.65	73.33	DEV_MESO
G32	28269 - 30481	RF	73.85	95.87	96.7	81.48	MSTPI_LD_B1
G33	22263 - 24333	GB	80.92	85.71	76.87	80.0	MSTPI_B1
G34	19541 - 22100	RF	73.5	80.58	82.33	73.33	DEV_MESO
G35	24493 - 27933	RF	74.34	76.97	72.8	75.68	MSTPI_LD_B2
G36	64264 - 72269	RF	73.75	82.14	83.62	72.0	DEV_MESO
G37	46772 - 51510	RF	64.78	91.26	92.89	71.43	ORIENTACIÓN
G38	41324 - 46543	RF	79.77	98.35	98.54	77.78	DEV_MACRO
G39	30684 - 37118	RF	74.35	87.5	82.52	70.27	DEV_MACRO
G40	37202 - 41197	RF	87.29	84.05	93.91	85.71	DEV_MACRO
G41	56448 - 63121	RF	66.8	52.48	46.85	72.0	DEV_MACRO
G42	51638 - 19558	RF	76.75	85.31	84.19	76.32	MSTPI_B1
G43	202448 - 392313	RF	52.22	49.09	50.38	54.55	MSTPI_B3

También se ha observado que la segunda variable condicional es una variable derivada de DEV meso. Se trata de los registros obtenidos en la banda verde de una imagen de multiescala, lo que significa que son los valores DEV meso, pero integrados dentro de una composición RGB. Por tanto, la importancia de los valores de mesoescala son relevantes a la hora de discriminar las clases, hasta tal punto, que lo son no sólo en los yacimientos cuyo rango corresponde a la mesoescala (ventanas: G1, G2, G6, G7, G10, G11, G12, G14, G15, G16, G18, G19, G20, G21 y G24), sino también en los yacimientos de rango de microescala (ventana G9) y en gran parte de los yacimientos de macroescala (ventanas: G22, G23, G26, G27, G28, G29, G31, G34 y G36).

Según los métodos de evaluación del rendimiento de los clasificadores se ha observado que los valores de media geométrica (GM) y F1 score están altamente correlacionados (+0.932 correlación de Pearson). Existe una dependencia lineal entre dichas variables (Figura 27). Los mejores modelos son los que se ubican en la esquina superior derecha y más cerca de la línea de regresión que relacione estas dos variables. Según esta gráfica los modelos G6, G8, G12, G13, G15, G20, G25, G40 tienen valores superiores a 80, tanto en media geométrica (GM) como en F1 score. De la misma manera, los modelos G14, G19, G29 y G43 se encuentran en lado opuesto y, por tanto, son clasificadores no eficientes.

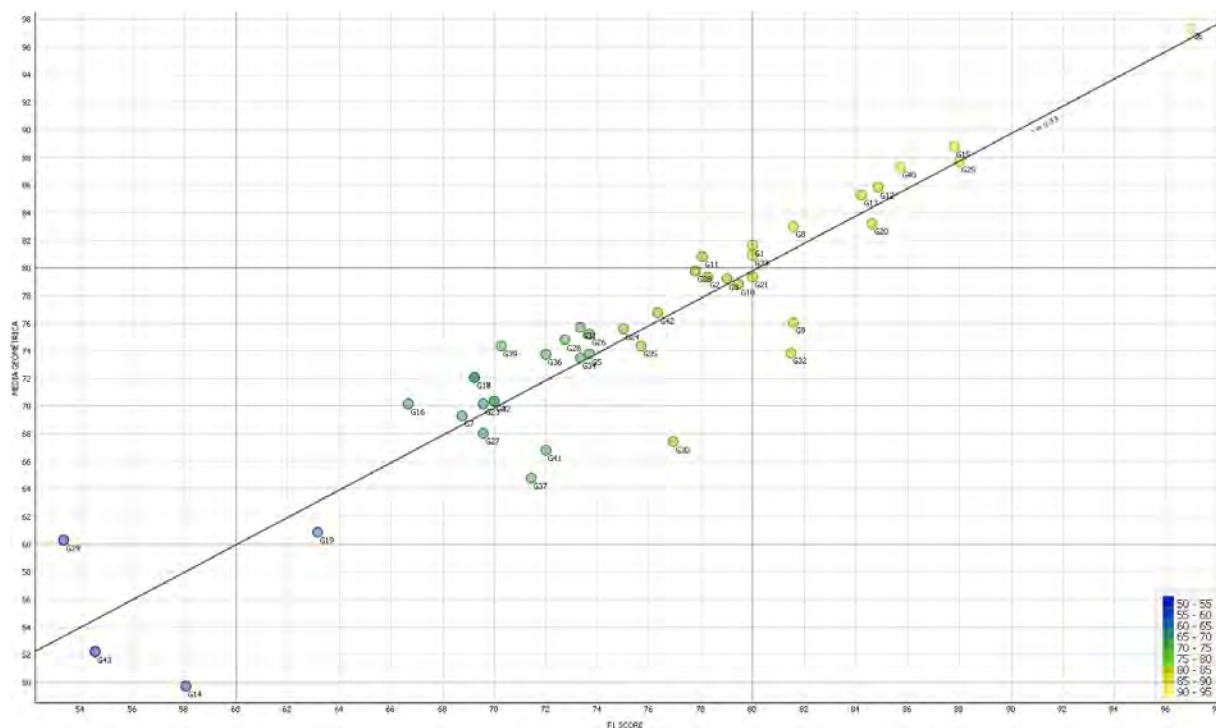


Figura 27. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score. La escala de colores de la leyenda corresponde a los valores de F1 score. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

3.2 Comparativa entre modelos

El mejor modelo obtenido corresponde al definido por la ventana de observación G6, mientras que el peor corresponde al G43. Sus diferencias radican en que en la ventana G6 está compuesta de más de un 54% de yacimientos de tipo "Lugar Funerario" en un rango de 457 m² a 520 m², mientras que la ventana G43 está compuesta de un 80% de yacimientos tipo

“Lugar de habitación” en un rango de 202.448 m² a 392.313 m². La excesiva superficie de la ventana G43 implica que no se pueda diferenciar entre formaciones geomorfológicas naturales de lo que son los núcleos de población que han ocupado esos espacios geomorfológicos tan extensos y, por tanto, no se pueda producir una separación efectiva de los datos. De manera contraria, la ventana G6 consigue entender que los lugares funerarios presentan formas circulares u ovaladas que son las que comúnmente definen los lugares funerarios de tipo dolmen o túmulo. El hecho de que la ventana G6 corresponda al mejor clasificador se encuentra en línea con las conclusiones de la bibliografía específica consultada en este trabajo [51], [57] que refiere a clasificadores cuyos valores de *True positive rate* son superiores a 90%, y valores de *False positive rate* inferiores al 10%, tal y como se muestran en la ventana G6.

No obstante, la bibliografía ya había demostrado que si se segmenta previamente los yacimientos a estudiar por tipos o tipologías basadas en su forma geométrica estos pueden entrenar modelos de forma más exitosa. Con este trabajo se observa que si se segmenta mediante el uso de ventanas regulares asociadas a la extensión de los yacimientos inventariados se puede clasificar también de manera exitosa.

Sin embargo, no se ha encontrado ninguna correlación relevante entre las medidas de rendimiento (Media geométrica, F1 score, ROC AUC o AUC PR) y los valores de dispersión de las tipologías de yacimientos dentro de cada ventana (Figuras 28 a 31), entendida la dispersión como un reflejo de la variabilidad de tipologías y tipos de yacimientos que forman parte de cada ventana.

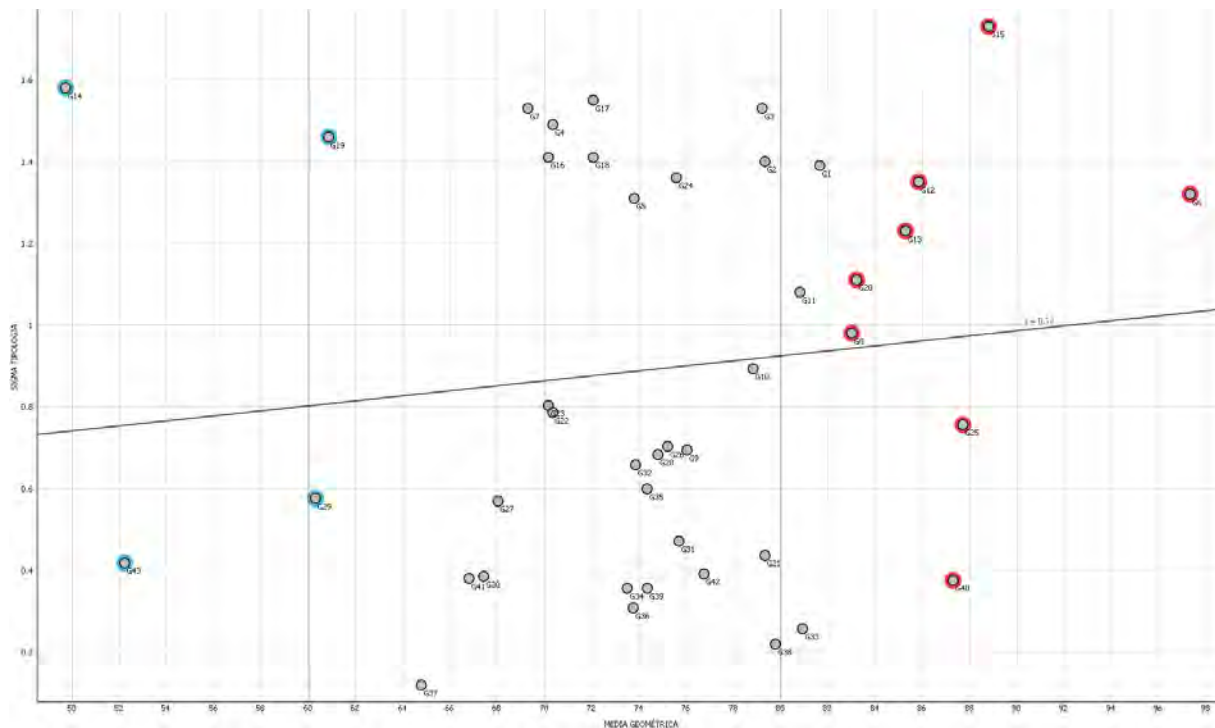


Figura 28. Gráfica de dispersión entre la medida de evaluación Media Geométrica y los valores de dispersión de tipologías de yacimientos usados en cada ventana. En rojo seleccionados los buenos clasificadores y en azul los que aportan una resultado más pobres. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

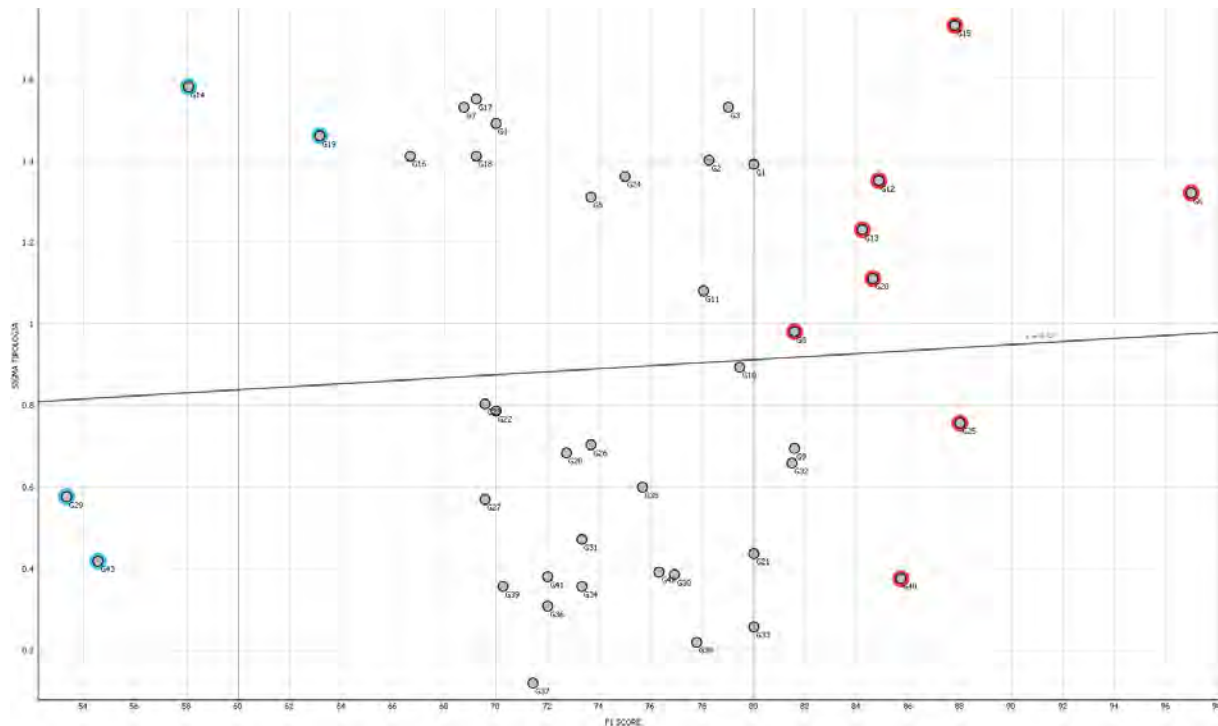


Figura 29. Gráfica de dispersión entre la medida de evaluación F1 score y los valores de dispersión de tipologías de yacimientos usados en cada ventana. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

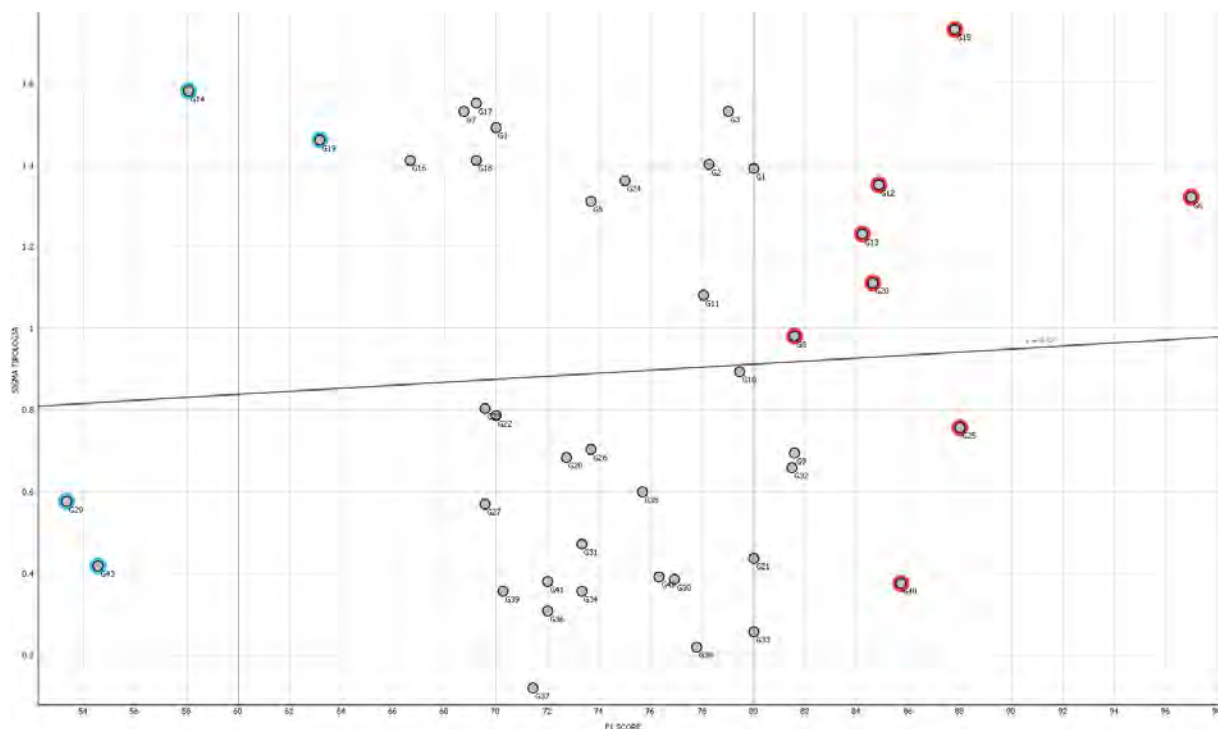


Figura 30. Gráfica de dispersión entre la medida de evaluación ROC AUC y los valores de dispersión de tipologías de yacimientos usados en cada ventana. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

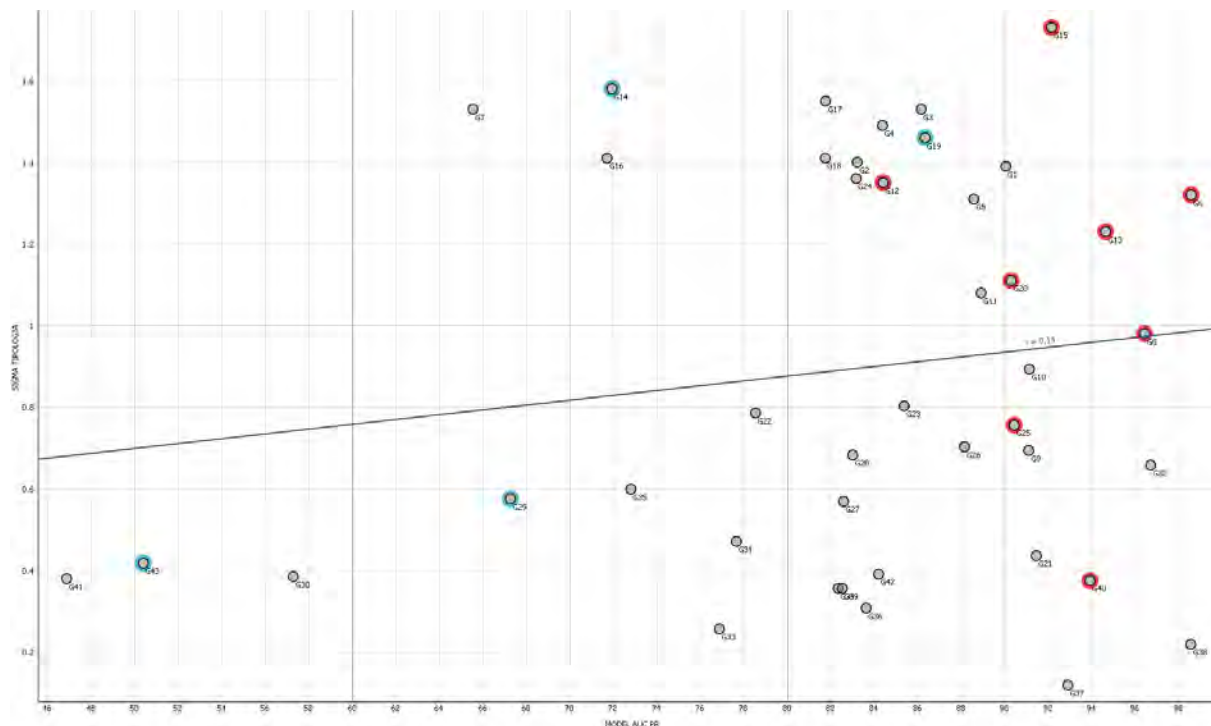


Figura 31. Gráfica de dispersión entre la medida de evaluación AUC PR y los valores de dispersión de tipologías de yacimientos usados en cada ventana. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

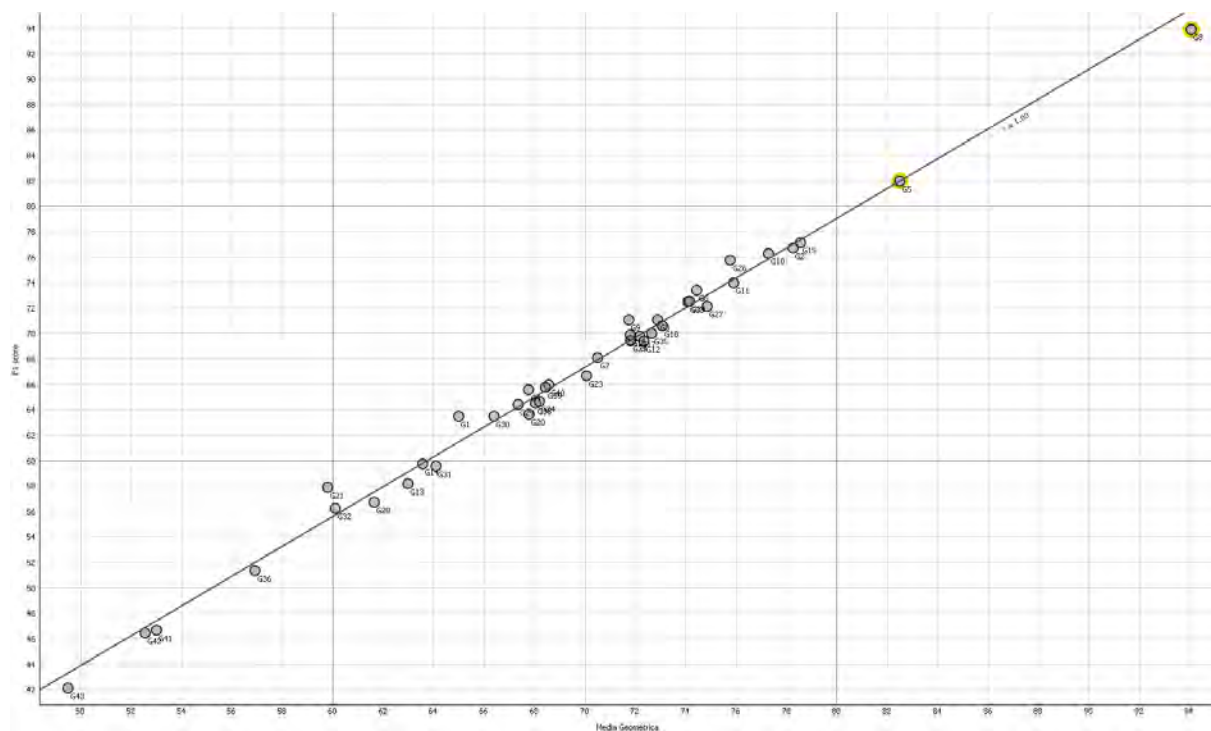


Figura 32. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score para los datos de la ventana G8 aplicados al resto de ventanas diseñadas. Resalta con valores superiores a 80 la ventana G5. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

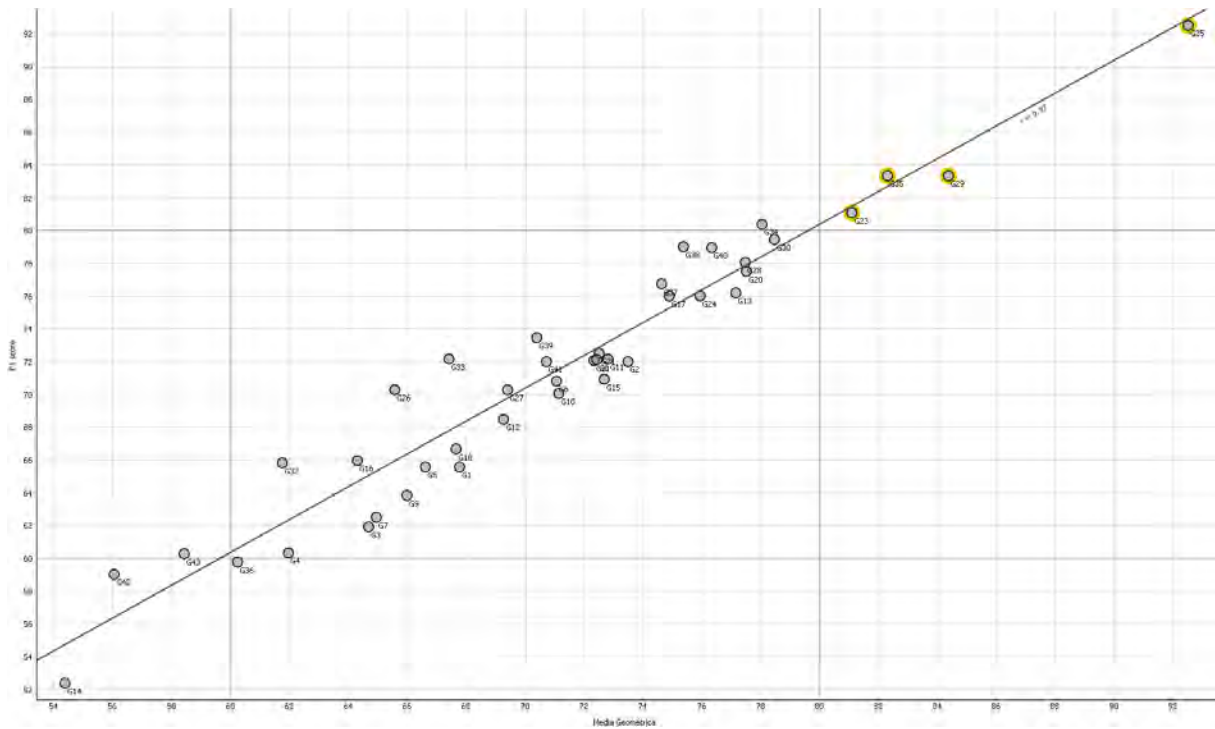


Figura 33. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score para los datos de la ventana G25 aplicados al resto de ventanas diseñadas. Resalta con valores superiores a 80 las ventanas G23, G29 y G35. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

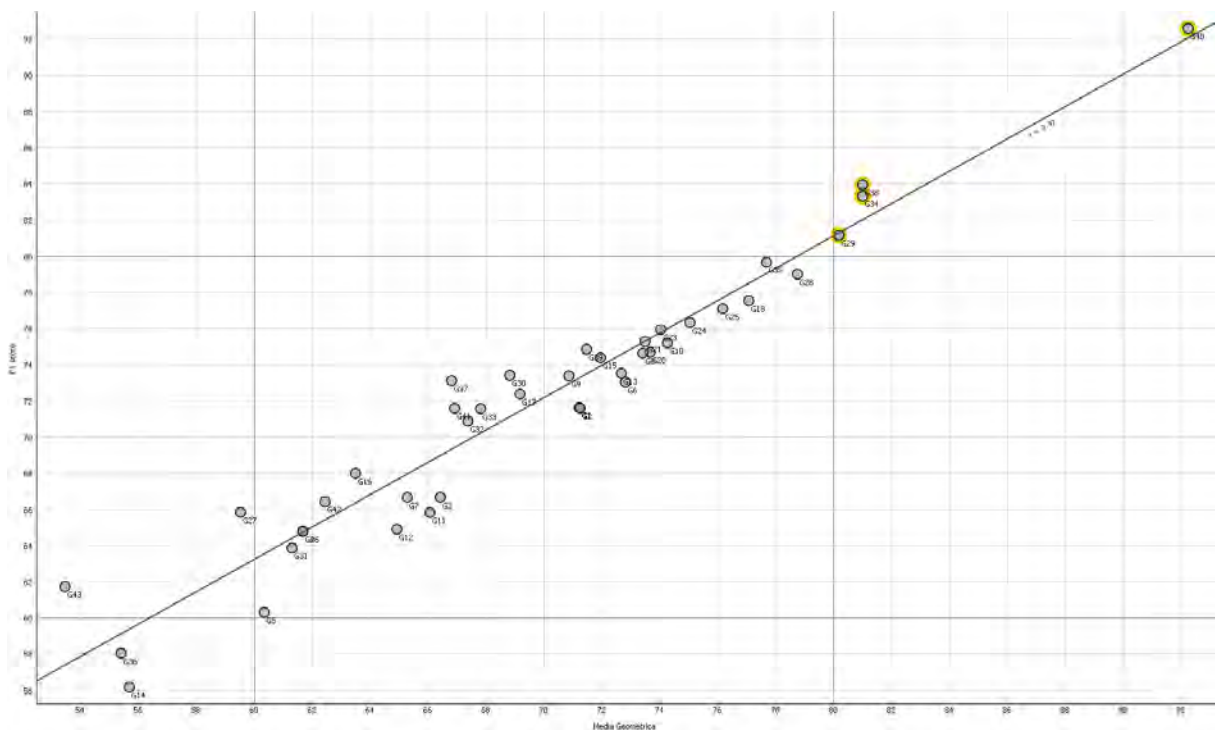


Figura 34. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score para los datos de la ventana G40 aplicados al resto de ventanas diseñadas. Resalta con valores superiores a 80 las ventanas G29, G34 y G38. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

Se reseña que en este proceso se ha observado como la ventana G15 presenta un problema de *overfitting* o *underfitting* ya que en este procedimiento no ha sido capaz de predecir los valores de su propio *dataset*. Por tanto, la ventana G15 se ha eliminado del grupo de buenos clasificadores (Figura 35).

Parece adecuado valorar que observando los datos de los modelos G8, G12, G13, G15, G20, G25 y G40, que al igual que el modelo G6 forman parte del grupo de buenos clasificadores, existe una tendencia a clasificar mejor cuanto más dispersión de tipologías exista en la ventana de observación. No obstante, los datos de los clasificadores moderados y malos informan que esta tendencia no se puede considerar como norma.

¿Pueden los mejores modelos entender los datos de otras ventanas de observación? Otro de los ejercicios planteados ha sido el empleo de los mejores modelos (G6, G8, G12, G13, G15, G20, G25, G40) para predecir los datos del resto de ventanas. Al observar los valores de media geométrica y F1 score resultantes se observa como las ventanas G8, G25 y G40 pueden predecir adecuadamente datos de otras ventanas con un rendimiento mejor (Figuras 33 a 34).

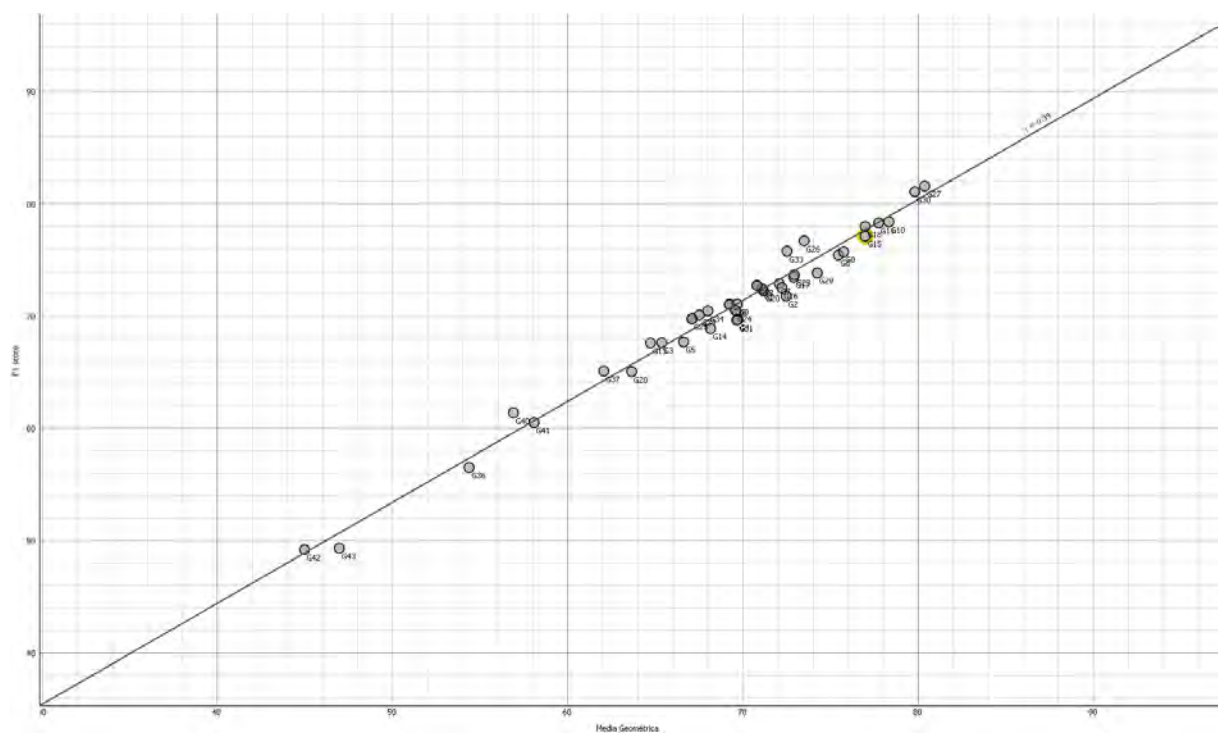


Figura 35. Gráfica de dispersión entre las medidas de evaluación Media Geométrica y F1 score para los datos de la ventana G15 aplicados al resto de ventanas diseñadas. Resalta con valores inferiores a 80 la propia ventana G15. Elaboración propia a partir de las herramientas Correlations y Scatter plot de Orange Data Mining 3.29.

De todo el proceso se deduce que las ventanas G6, G8, G12, G13, G20, G25 y G40 son buenos clasificadores, capaces de diferenciar entre un yacimiento aleatorio de otro topográfico dentro de una ventana de observación dada.

3.3 Predicción de nuevos datos

Llegados a este punto nos surge la pregunta de cómo se comportarán estos modelos aplicados a un caso práctico. Hay que destacar que el procedimiento utilizado está diseñado para plasmar la probabilidad que los datos observados con una ventana determinada pertenezcan o no a una anomalía topográfica de origen arqueológico, es decir, no propiamente a localizar por sí mismo yacimientos, sino a hacer esta búsqueda más eficaz, a través de la combinación de distintos métodos de visualización.

El objetivo es mejorar la visualización de los datos LiDAR para agilizar la búsqueda de zonas susceptibles de ser anomalías de origen arqueológico. Si se observa el MDT de 2017 en una escala determinada (1:6.500) usando diferentes modos de visualización como son el sombreado analítico (Figura 36), el sombreado anisotrópico (Figura 37), la imagen de multiescalar (Figura 38) y una imagen multiescalar fundida con una imagen de dominancia local (Figura 39) se pueden discernir, si se cuenta con experiencia, donde se puede localizar yacimientos arqueológicos.

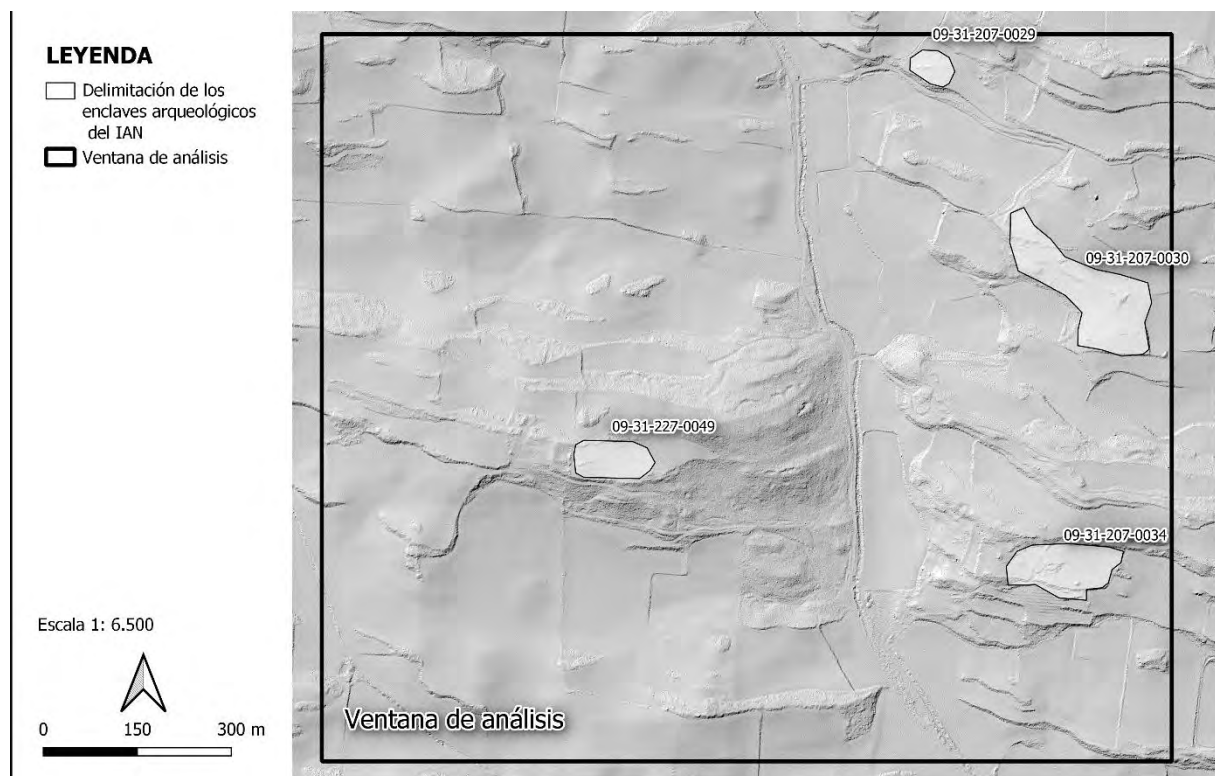


Figura 36. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre el sombreado analítico disponible en IDENA (ELEVAC_Ras_RelieveBN_MDT_50CM_VE2017). Elaboración propia.

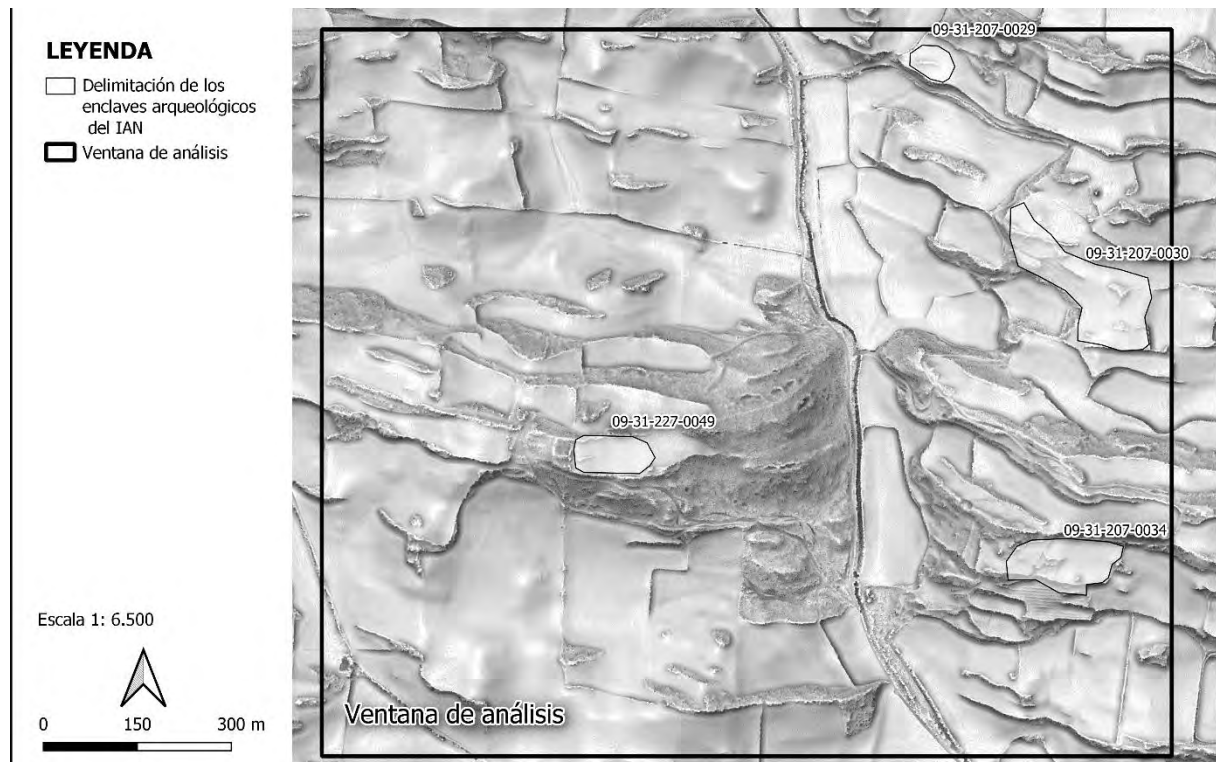


Figura 37. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre el sombreado orográfico basado en iluminación anisotrópica disponible en IDENA (ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017). Elaboración propia.

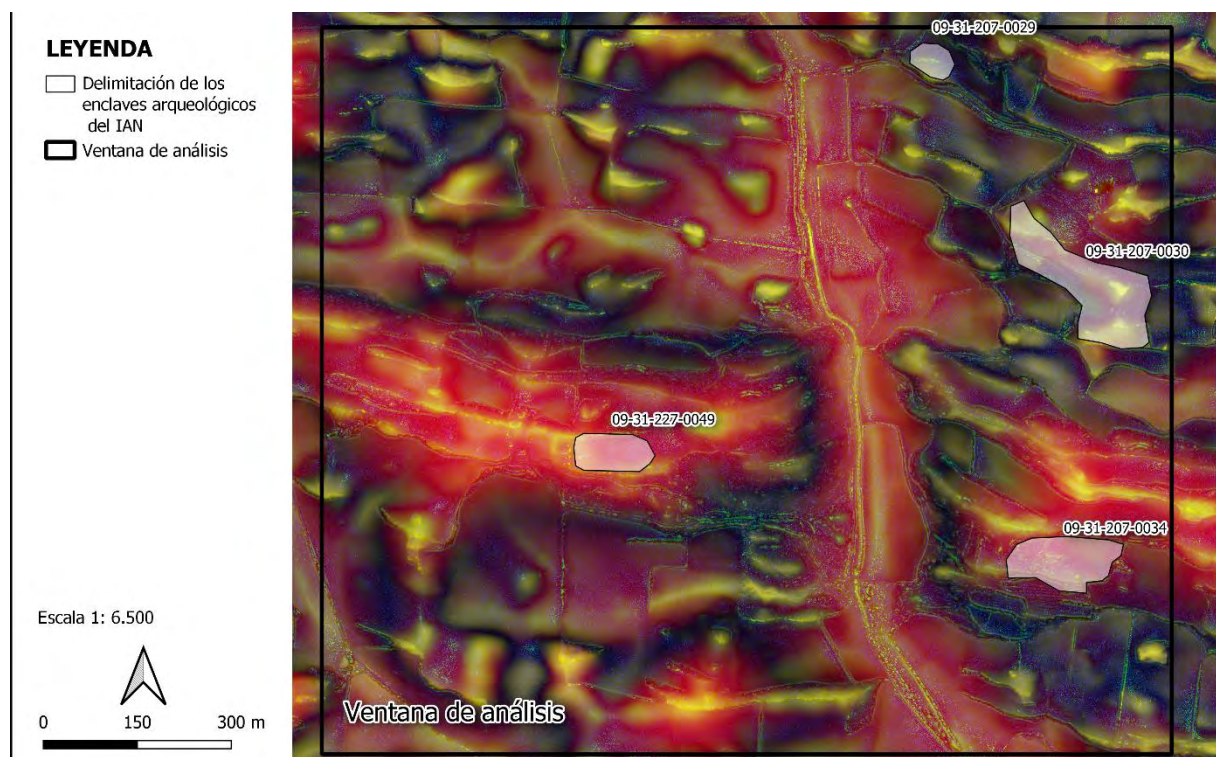


Figura 38. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre la imagen multiescalar elaborada a partir de la composición RGB de los valores de DEVmax. Elaboración propia.

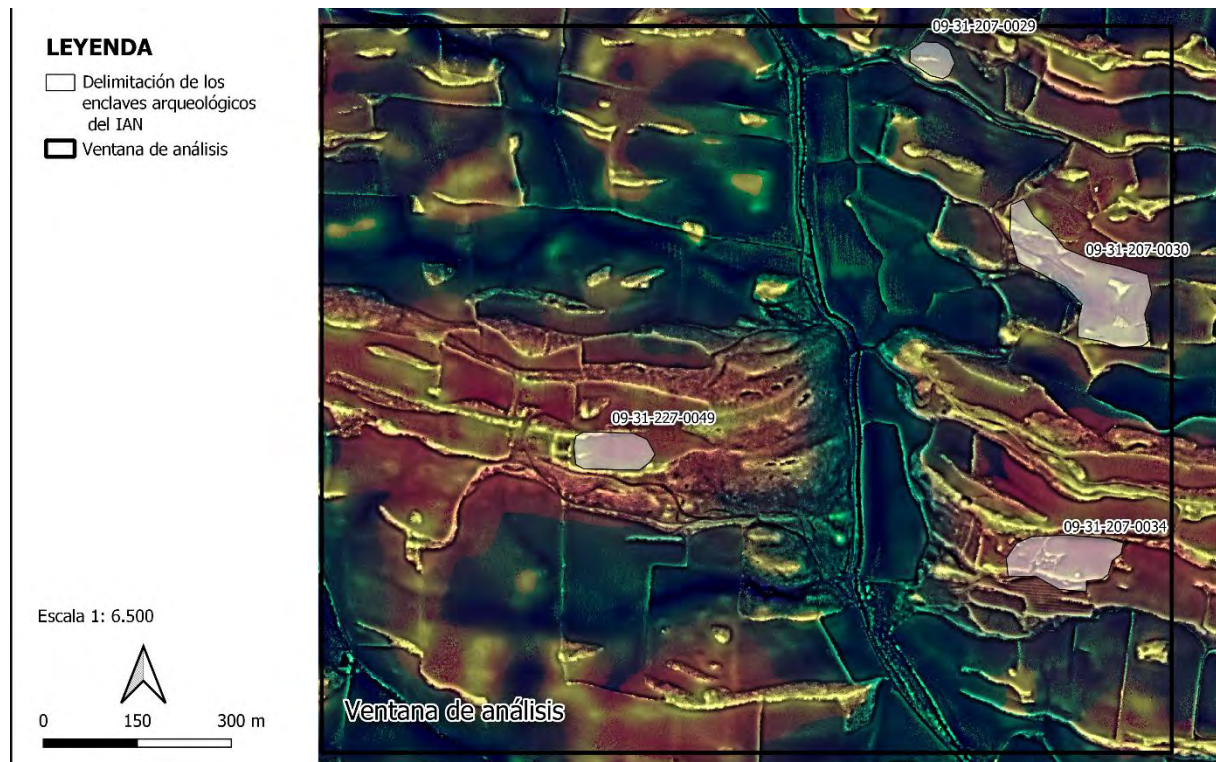


Figura 39. Plano de la ventana de análisis con la ubicación de los yacimientos que figuran en el IAN para dicha zona sobre la imagen multiescalar elaborada a partir de la composición RGB de los valores de DEVmax, fundida con los valores de Local Dominance. Elaboración propia.

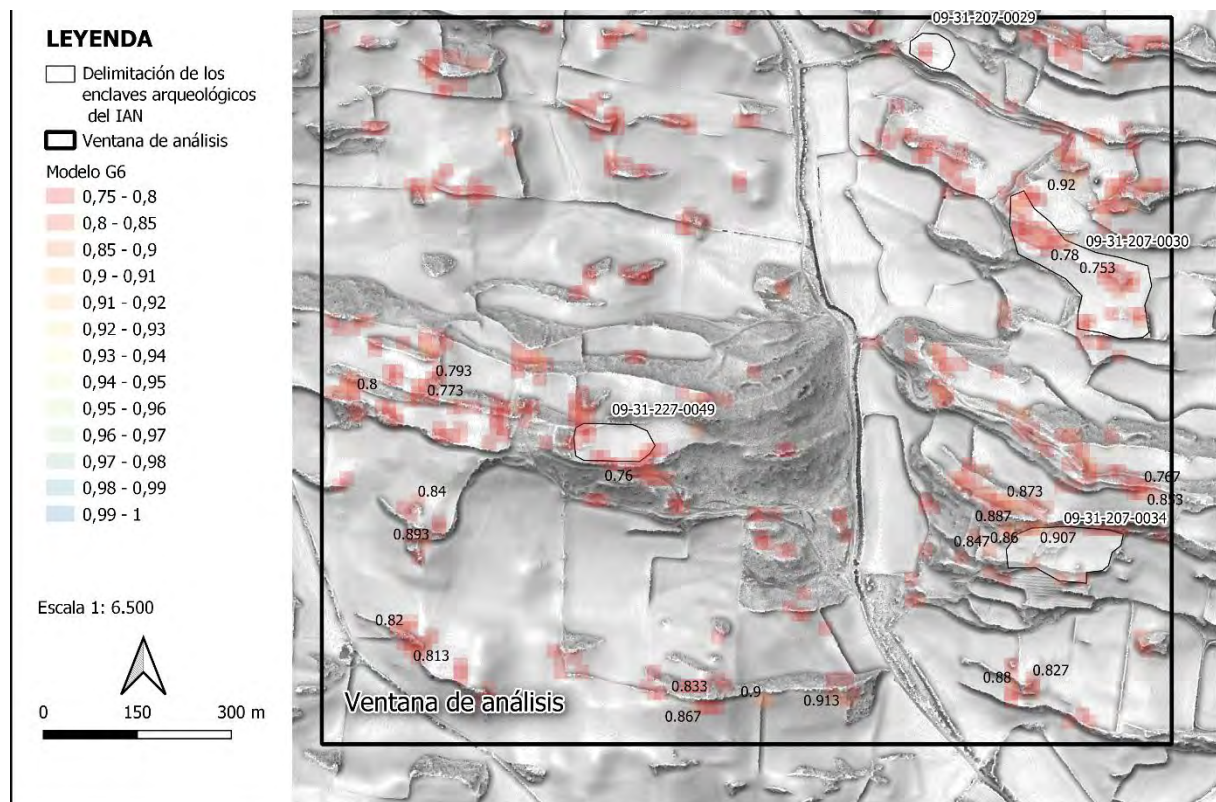


Figura 40. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G6 (rango 457 - 520 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.

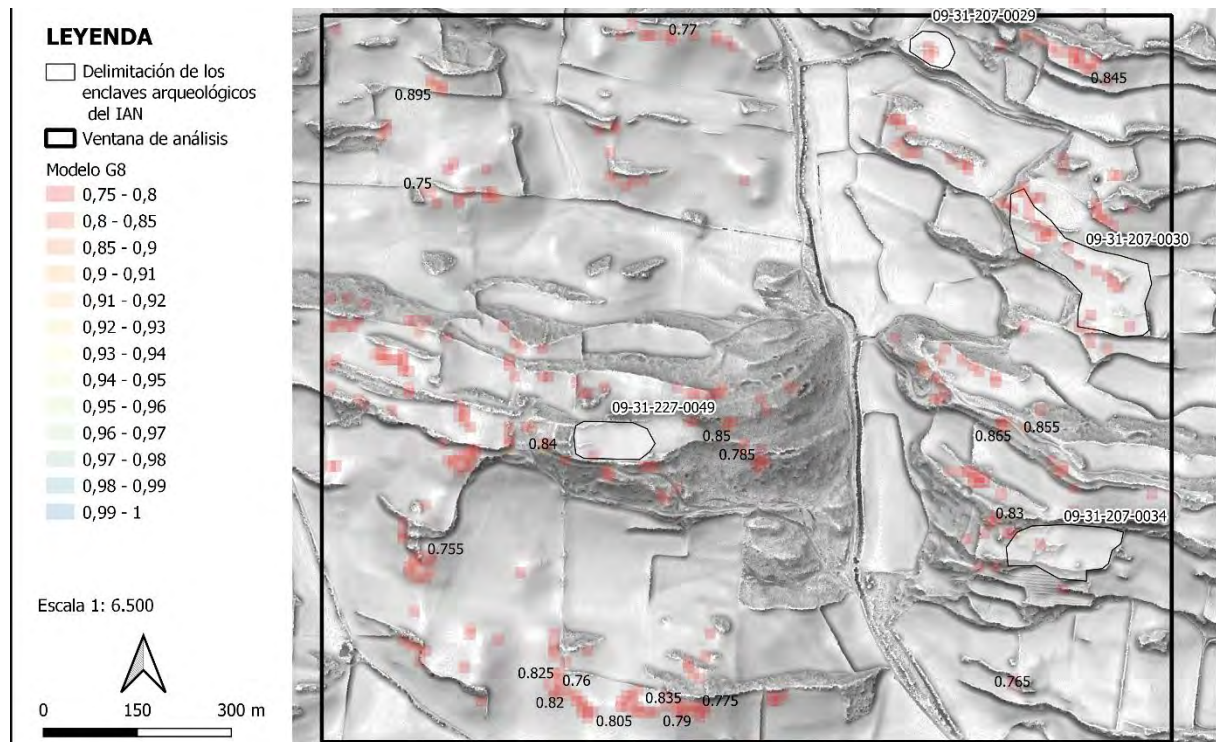


Figura 41. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G8 (rango 237 - 318 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.

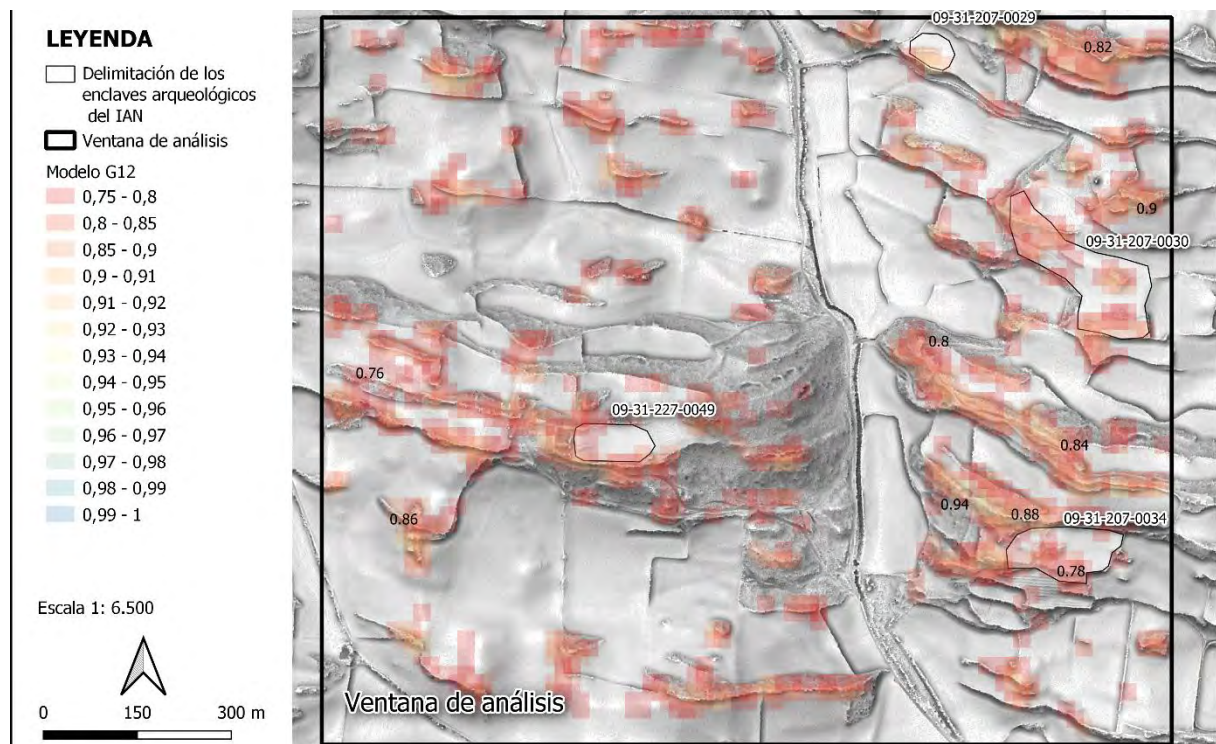


Figura 42. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G12 (rango 675 - 794 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.

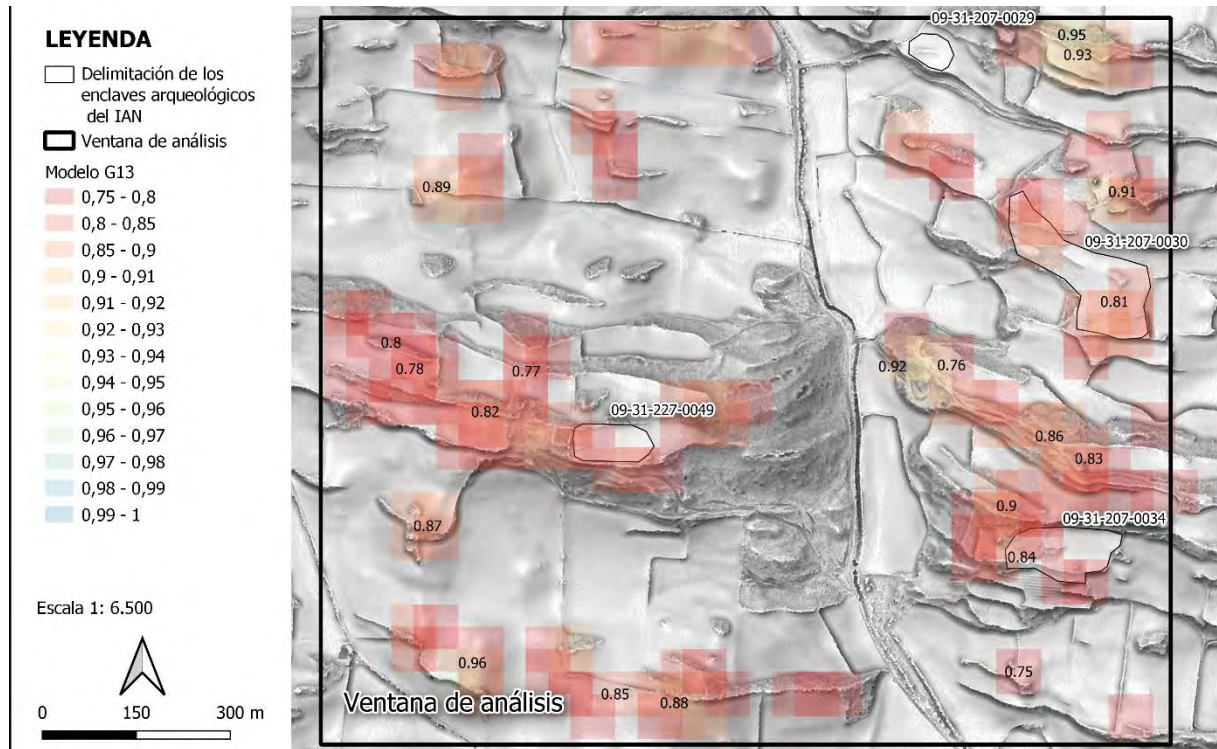


Figura 43. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G13 (rango 4.724 -5.118 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.

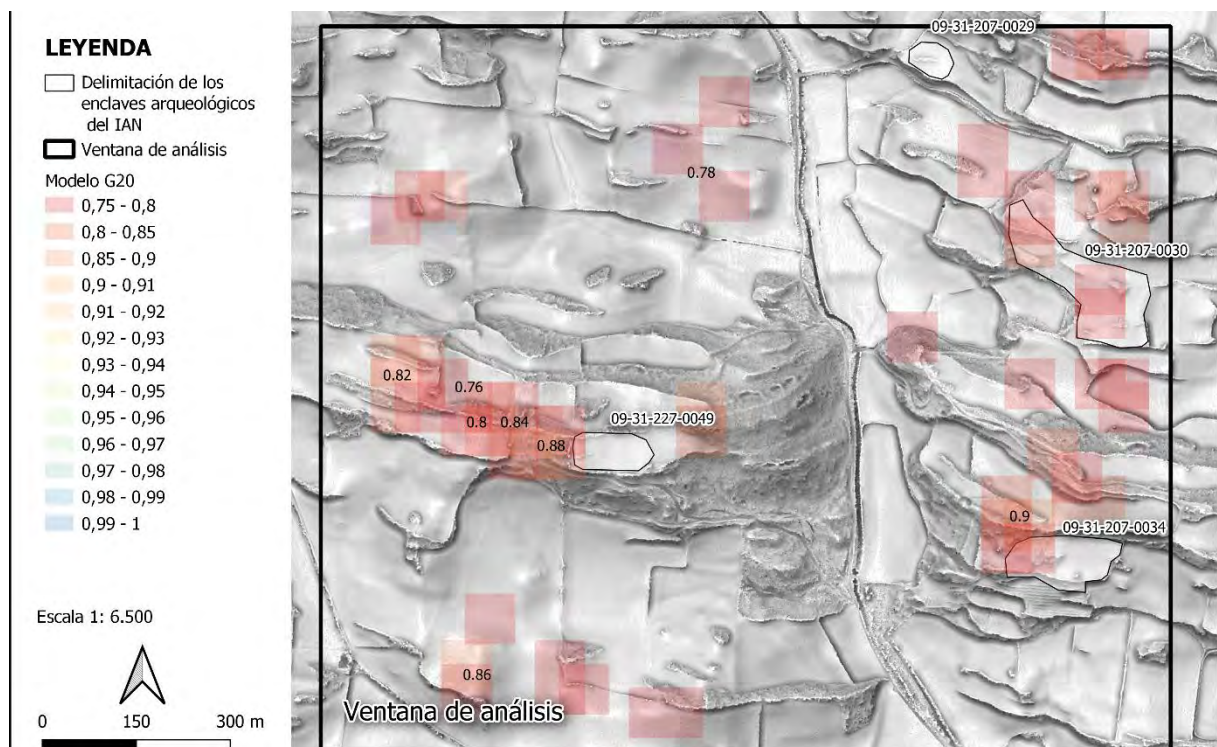


Figura 44. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G20 (rango 6.603 -7.550 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.

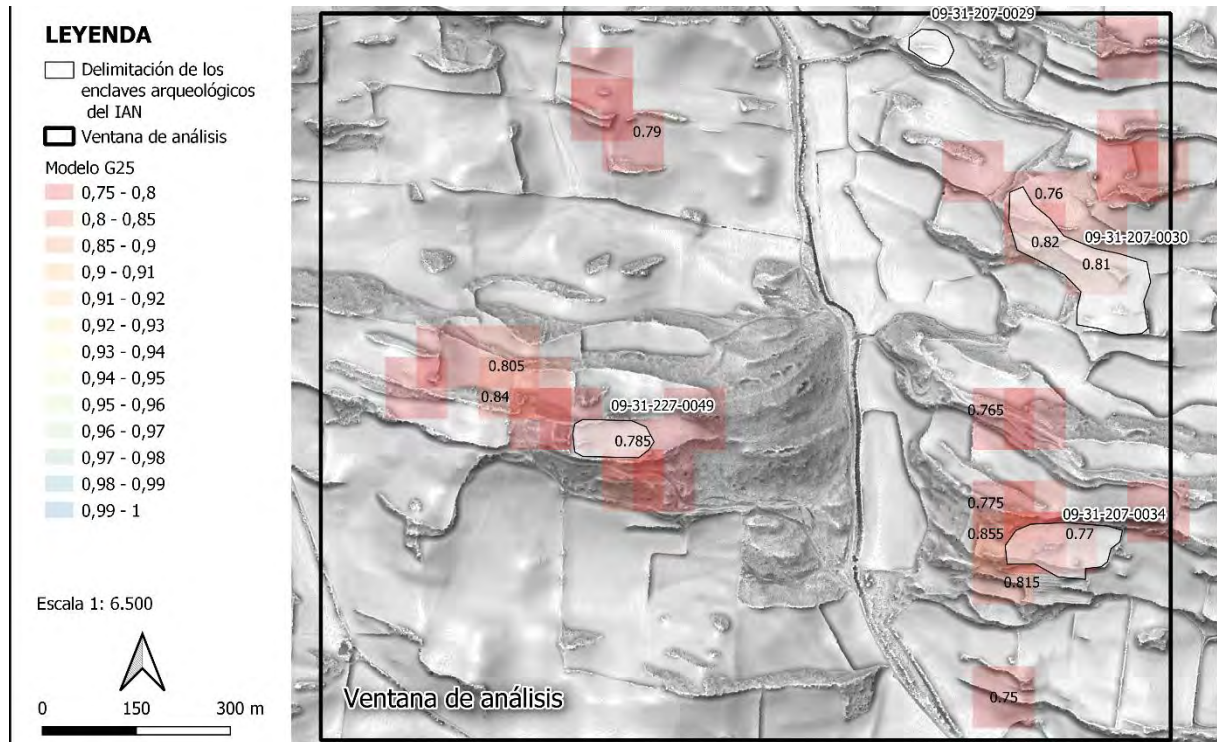


Figura 45. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G25 (rango 8.537 -9.725 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.

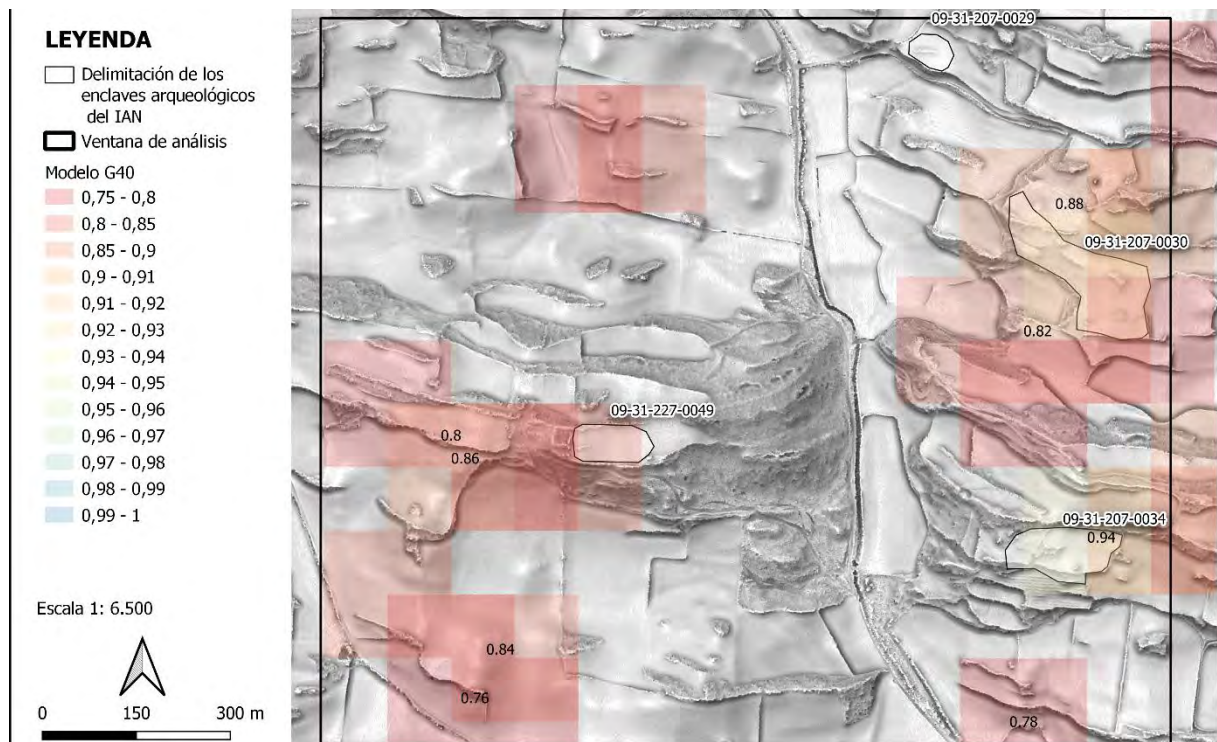


Figura 46. Probabilidades de presencia de anomalías de origen arqueológico relacionadas con el modelo de ventana G40 (rango 37.202 -41.177 m²); los valores se solapan sobre el servicio wms de IDENA: ELEVAC_Ras_RelieveBN_ANI_MDT_50CM_VE2017. Elaboración propia.

Al superponer los valores de probabilidad calculados para cada modelo elegido (G6, G8, G12, G13, G20, G25 y G40) sobre uno de estos modos de visualización se obtiene un valor añadido que ayuda a sesgar nuestra percepción hacia dónde se pueden localizar las anomalías de origen arqueológico (Figuras 40 a 46).

Hay que tener en cuenta que en las imágenes mostradas podría deducirse que existe algo de sobreajuste de los datos. Mostrando más claramente donde existen anomalías topográficas. Estas pueden deberse a cualquier tipo de uso tradicional del paisaje, como por ejemplo los márgenes de las fincas agrícolas, que la Zona Media son desniveles muy pronunciados, y muchas veces de origen antrópico (muros de mampostería).

Los modelos discriminan de forma eficiente donde deben encontrarse anomalías topográficas, por lo tanto, determinan donde las variables topográficas no van a predecir la presencia de un enclave arqueológico.

5.- Conclusiones

En el desarrollo de este trabajo se ha visto que una adecuada clasificación depende de una correcta segmentación de los datos de partida con los que se va a entrenar el modelo.

El método de segmentación empleado en este trabajo se basa en el uso de ventanas de observación definidas por la superficie de los yacimientos del Inventario Arqueológico de Navarra. Este tipo de segmentación es novedosa y consigue explicar adecuadamente los datos que presentan algún tipo de anomalía topográfica que se pueda vincular a un origen arqueológico.

Otra de las aportaciones de este trabajo, es la desvinculación durante el proceso de segmentación de las tipologías y tipos con los que se clasifican habitualmente los yacimientos arqueológicos, centrándose únicamente en variables continuas como la superficie del yacimiento.

Se confirma lo descrito en la bibliografía especializada en cuanto al uso de los métodos de visualización basados en combinaciones de productos derivados de LiDAR para la mejor identificación de entornos arqueológicos. Se aporta en este ámbito la combinación de las imágenes de multiescala con los valores de *Local Dominance* como una herramienta válida para sesgar la percepción de quien realiza las tareas de prospección virtual.

También se confirma que el uso de la variable multiescalar, en concreto los valores asociados a la mesoescala (de 100 a 10.000 m²), es determinante para la separación entre la clase positiva (arqueológico) de la clase negativa (aleatorio).

Finalmente, se considera que no se puede aún sustituir la prospección física *in situ* por modelos matemáticos, ya que la localización se debe dar observando físicamente la materialidad de la evidencia, pero estos modelos sí que suponen un avance significativo en cómo localizar nuevos yacimientos. Las futuras líneas de investigación como continuación de este trabajo pasarán por emplear los mejores modelos de observación como base para entrenar redes neuronales convolucionales preentrenadas para realizar tareas de detección de objetos, en la línea que ya se está trabajando en los ámbitos universitarios centroeuropeos [82], [83].

6.- Lista de referencias

- [1] Gobierno de Navarra, "LEY FORAL 14/2005, DE 22 DE NOVIEMBRE, DEL PATRIMONIO CULTURAL DE NAVARRA," *BON*, vol. 141, no. 25/11/2005, 2005.
- [2] Dirección General de Cultura (institución Príncipe de Viana), "INVENTARIO ARQUEOLÓGICO DE NAVARRA 2018," *Departamento de Cultura, Deporte y Juventud*, 2018. [Online]. Available: <https://hacienda.navarra.es/sicportal/mtoAnunciosModalidad.aspx?Cod=180412134051F75E80E5>.
- [3] L. García Sanjuán, *Introducción al Reconocimiento y Análisis Arqueológico del Territorio*, 1ª Edición. Barcelona: Ariel Prehistoria, 2005.
- [4] L. Luo *et al.*, "Airborne and spaceborne remote sensing for archaeological and cultural heritage applications: A review of the century (1907–2017)," *Remote Sensing of Environment*, vol. 232, 2019.
- [5] A. Schmidt, *Earth resistance for archaeologists: Geophysical methods for archaeology*. Lanham: Altamira Press, 2013.
- [6] "Web of Science." [Online]. Available: <https://www.webofscience.com/wos/alldb/analyze-results/a7106342-9bb3-41de-aeec-c575efdebc6f-06dc7fa7>. [Accessed: 04-Sep-2021].
- [7] O.G.S. Crawford, "Air Survey and Archaeology," *The Geographical Journal*, vol. 61, no. 5 (May, 1923), pp. 342–360, 1923.
- [8] D. Wilson, *Air Photo Interpretation for Archaeologists*, 2ª Edición. The History Press Ltd, 1982.
- [9] R. Lasaponara and N. Masini, "Identification of archaeological buried remains based on the normalized difference vegetation index (NDVI) from quickbird satellite data," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 3, pp. 325–328, 2006.
- [10] R. Lasaponara and N. Masini, "Detection of archaeological crop marks by using satellite QuickBird multispectral imagery," *Journal of Archaeological Science*, vol. 34, no. 2, pp. 214–221, 2007.
- [11] R. Lasaponara and N. Masini, *Satellite Remote Sensing. A new Tool for Archaeology*. New York: Springer, 2012.
- [12] A. Agapiou, "Orthogonal equations for the detection of hidden archaeological remains demystified," *Journal of Archaeological Science: Reports*, vol. 14, pp. 792–799, 2017.
- [13] A. Agapiou, D. D. Alexakis, A. Sarris, and D. G. Hadjimitsis, "Orthogonal equations of multi-spectral satellite imagery for the identification of un-excavated archaeological sites," *Remote Sensing*, vol. 5, no. 12, pp. 6560–6586, 2013.
- [14] C. Atzberger, M. Wess, M. Doneus, and G. Verhoeven, "ARCTIS - A MATLAB® toolbox for archaeological imaging spectroscopy," *Remote Sensing*, vol. 6, no. 9, pp. 8617–8638, 2014.
- [15] M. Tzouvaras, D. Kouhartsiouk, A. Agapiou, C. Danezis, and D. G. Hadjimitsis, "The use of Sentinel-1 synthetic aperture radar (SAR) images and open-source software for cultural heritage: An example from paphos area in Cyprus for mapping landscape changes after a 5.6 magnitude earthquake," *Remote Sensing*, vol. 11, no. 15, 2019.
- [16] A. Wehr and U. Lohr, "Airborne laser scanning - An introduction and overview," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2–3, pp. 68–82, 1999.
- [17] J. A. Mozos and A. M. de Aguirre, "Tema 6. Introducción al LiDAR," in *Teledetección RADAR y LiDAR*, Máster Universitario en Sistemas de Información Geográfica y Teledetección

- (MUSIGT), Ed. Pamplona: ETSIAB. Universidad Pública de Navarra, 2019.
- [18] “How LiDAR works.” [Online]. Available: <https://www.yellowscan-lidar.com/es/knowledge/how-lidar-works/>. [Accessed: 05-Sep-2021].
- [19] J. Fernandez-Diaz, W. Carter, R. Shrestha, and C. Glennie, “Now You See It... Now You Don’t: Understanding Airborne Mapping LiDAR Collection and Data Product Generation for Archaeological Research in Mesoamerica,” *Remote Sensing*, vol. 6, no. 10, pp. 9951–10001, Oct. 2014.
- [20] Ž. Kokalj and M. Somrak, “Why not a single image? Combining visualizations to facilitate fieldwork and on-screen mapping,” *Remote Sensing*, vol. 11, no. 7, 2019.
- [21] Ž. Kokalj, K. Zakšek, K. Oštir, P. Pehani, and K. Čotar, “Relief Visualization Toolbox, ver. 1.3 Manual,” p. 9, 2016.
- [22] Z. & H. Kokalj R., *Airborne Laser Scanning Raster Data Visualization*. 2017.
- [23] A. Guyot, M. Lennon, and L. Hubert-Moy, “Objective comparison of relief visualization techniques with deep CNN for archaeology,” *Journal of Archaeological Science: Reports*, vol. 38, no. April, p. 103027, 2021.
- [24] A. Montufo Martin, “Aplicaciones de la teledetección en arqueología. Una revisión crítica,” *Cuadernos de Prehistoria*, vol. 100, pp. 425–451, 1991.
- [25] E. Cerrillo-Cuenca and A. López López, “Evaluación y perspectivas del uso del LiDAR en la arqueología española,” *Boletín del Museo Arqueológico Nacional*, no. 39, pp. 221–238, 2020.
- [26] J. M. Costa García and R. Casal García, “Fotografía aérea histórica, satelital moderna y LiDAR aéreo en algunos recintos militares romanos de Castilla y León,” *Portugalia, Nova Serie*, vol. 36, pp. 143–158, 2015.
- [27] J. M. Costa-García, J. Fonte, and M. Gago, “The reassessment of the roman military presence in Galicia and northern Portugal through digital tools: Archaeological diversity and historical problems,” *Mediterranean Archaeology and Archaeometry*, vol. 19, no. 3, pp. 17–49, 2019.
- [28] J. M. Vidal Encinas, J. M. Costa García, D. González Álvarez, A. Menéndez Blanco, X. de G. Secretaría Xeral de Universidades, and X. de G. GAIN-Axencia Galega de Innovación, “La presencia del ejército romano en las montañas de El Bierzo (León): novedades arqueológicas,” 2019.
- [29] L. Berrocal-Rangel, P. Paniego Díaz, L. Ruano, and G. R. Manglano Valcárcel, “Aplicaciones LiDAR a la topografía arqueológica: El Castro de Iruña (Fuenteguinaldo, Salamanca) / LiDAR applications to the archaeological topography: The Iruña Hillfort (Fuenteguinaldo, Salamanca),” *Cuadernos de Prehistoria y Arqueología*, vol. 43, no. 2017, 2017.
- [30] J. Fernández-Lozano, G. Gutiérrez-Alonso, and M. Á. Fernández-Morán, “Using airborne LiDAR sensing technology and aerial orthoimages to unravel roman water supply systems and gold works in NW Spain (Eria valley, León),” *Journal of Archaeological Science*, vol. 53, pp. 356–373, 2015.
- [31] E. Cerrillo-Cuenca, “An approach to the automatic surveying of prehistoric barrows through LiDAR,” *Quaternary International*, vol. 435, pp. 135–145, 2017.
- [32] E. Cerrillo-Cuenca and P. Bueno-Ramírez, “Counting with the invisible record? The role of LiDAR in the interpretation of megalithic landscapes in south-western Iberia (Extremadura, Alentejo and Beira Baixa),” *Archaeological Prospection*, vol. 26, no. 3, pp. 251–264, 2019.
- [33] M. Carrero-Pazos, B. Vilas Estévez, E. Romaní Fariña, and A. A. Rodríguez Casal, “La

- necrópolis del Monte de Santa Mariña revisitada: aportaciones del LIDAR aéreo para la cartografía megalítica de Galicia,” *Gallaecia: revista de arqueología e antigüidade*, vol. 33, no. 0, pp. 39–57, 2015.
- [34] A. Monterroso-Checa, “Remote sensing and archaeology from Spanish LiDAR-PNOA: Identifying the amphitheatre of the roman city of torreparedones (Córdoba-Andalucía-Spain),” *Mediterranean Archaeology and Archaeometry*, vol. 17, no. 1, pp. 15–22, 2017.
- [35] J. G. Rejas, F. Burillo, R. López, and M. Farjas, “Aplicación de teledetección hiperespectral en la ciudad celtíbera de Segeda,” in *From Space to Place: 2nd[1] J. G. Rejas, F. Burillo, R. López, and M. Farjas, “Aplicación de teledetección hiperespectral en la ciudad celtíbera de Segeda,” in From Space to Place: 2nd International Conference on Remote Sensing in Archaeology, 2006. Inter, 2006.*
- [36] J. J. Rejas, J. G.; Burillo, Zancajo, “Teledetección Pasiva y Activa en Arqueología. Caso de estudio de la ciudad celtibérica de Segeda,” in *Teledetección: Agua y desarrollo sostenible. XIII Congreso de la Asociación Española de Teledetección. Rejas, J.C.; Burillo, F.; Zancajo, J.J. “Teledetección Pasiva y Activa en Arqueología. Caso de estudio de la ciudad celtibérica de Segeda,” 2009, pp. 497–500.*
- [37] J. J. F. González and F. R. V. Hernández, “NDVI identification and survey of a Roman road in the Northern Spanish province of Álava,” *Remote Sensing*, vol. 11, no. 6, pp. 1–20, 2019.
- [38] J. J. Fuldain González and J. I. Fuldain González, “Prospección arqueológica en NDVI con drones. El uso de geoEuskadi como herramienta de ponderación de un nuevo método,” *Mapping (1131-9100)*, vol. 27, no. 192, pp. 24–24–29, 2018.
- [39] H. A. Orenge and A. Garcia-Molsosa, “A brave new world for archaeological survey: Automated machine learning-based potsherd detection using high-resolution drone imagery,” *Journal of Archaeological Science*, vol. 112, p. 105013, Dec. 2019.
- [40] A. Duró Cazorla, “El Castillo de Gallipienzo. Memoria de intervención arqueológica,” Orísoain, 2019.
- [41] D. J. Bescoby, “Detecting Roman land boundaries in aerial photographs using Radon transforms,” *Journal of Archaeological Science*, vol. 33, no. 5, pp. 735–743, 2006.
- [42] M. Jahjah, C. Ulivieri, A. Invernizzi, and R. Parapetti, “Archaeological remote sensing application pre-post war situation of Babylon archaeological site-Iraq,” *Acta Astronautica*, vol. 61, no. 1–6, pp. 121–130, 2007.
- [43] V. De Laet, E. Paulissen, and M. Waelkens, “Methods for the extraction of archaeological features from very high-resolution Ikonos-2 remote sensing imagery, Hisar (southwest Turkey),” *Journal of Archaeological Science*, vol. 34, no. 5, pp. 830–841, 2007.
- [44] P. Verhagen and L. Drăguț, “Object-based landform delineation and classification from DEMs for archaeological predictive mapping,” *Journal of Archaeological Science*, vol. 39, no. 3, pp. 698–703, 2012.
- [45] D. S. Davis, “Object-based image analysis: a review of developments and future directions of automated feature detection in landscape archaeology,” *Archaeological Prospection*, vol. 26, no. 2, pp. 155–163, 2019.
- [46] D. S. Davis, “Defining what we study: The contribution of machine automation in archaeological research,” *Digital Applications in Archaeology and Cultural Heritage*, vol. 18, no. July, p. e00152, 2020.
- [47] W. B. Verschoof-van der Vaart and K. Lambers, “Learning to Look at LiDAR: The Use of R-CNN in the Automated Detection of Archaeological Objects in LiDAR Data from the Netherlands,” *Journal of Computer Applications in Archaeology*, vol. 2, no. 1, pp. 31–40, 2019.

- [48] W. B. Verschoof-van der Vaart and J. Landauer, "Using CarcassonNet to automatically detect and trace hollow roads in LiDAR data from the Netherlands," *Journal of Cultural Heritage*, vol. 47, pp. 143–154, 2021.
- [49] Ø. D. Trier, D. C. Cowley, and A. U. Waldeland, "Using deep neural networks on airborne laser scanning data: Results from a case study of semi-automatic mapping of archaeological topography on Arran, Scotland," *Archaeological Prospection*, vol. 26, no. 2, pp. 165–175, Apr. 2019.
- [50] D. S. Davis, "Geographic Disparity in Machine Intelligence Approaches for Archaeological Remote Sensing Research," *Remote Sensing*, vol. 12, no. 6, p. 921, Mar. 2020.
- [51] A. Guyot, L. Hubert-Moy, and T. Lorho, "Detecting Neolithic burial mounds from LiDAR-derived elevation data using a multi-scale approach and machine learning techniques," *Remote Sensing*, vol. 10, no. 2, Feb. 2018.
- [52] M. Somrak, S. Džeroski, and Ž. Kokalj, "Learning to classify structures in ALS-derived visualizations of ancient Maya settlements with CNN," *Remote Sensing*, vol. 12, no. 14, 2020.
- [53] J. A. Sanz Delgado, "Tema 1. Introducción a la minería de datos. El proceso KDD," in *Sistemas Inteligentes para la extracción de información*, Máster en Sistemas de Información Geográfica y Teledetección (MUSIGT), Ed. Pamplona: ETSIAB. Universidad Pública de Navarra, 2020.
- [54] J. A. Sanz Delgado, "Tema 2. Definición de los problemas de clasificación. Evaluación de los resultados," in *Sistemas Inteligentes para la extracción de información*, Máster en Sistemas de Información Geográfica y Teledetección (MUSIGT), Ed. Pamplona: ETSIAB. Universidad Pública de Navarra, 2020.
- [55] E. Chuvieco, *Teledetección Ambiental*. Barcelona, 2002.
- [56] N. Acevedo, "Principales técnicas del machine learning." [Online]. Available: <https://nataliaacevedo.com/principales-tecnicas-de-machine-learning/>. [Accessed: 04-Sep-2021].
- [57] C. Sevara, M. Pregesbauer, M. Doneus, G. Verhoeven, and I. Trinks, "Pixel versus object - A comparison of strategies for the semi-automated mapping of archaeological features using airborne laser scanning data," *Journal of Archaeological Science: Reports*, vol. 5, pp. 485–498, 2016.
- [58] L. Magnini and C. Bettineschi, "Theory and practice for an object-based approach in archaeological remote sensing," *Journal of Archaeological Science*, vol. 107, pp. 10–22, 2019.
- [59] M. D. Hossain and D. Chen, "Segmentation for Object-Based Image Analysis (OBIA): A review of algorithms and challenges from remote sensing perspective," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, no. November 2018, pp. 115–134, 2019.
- [60] A. Guyot, M. Lennon, T. Lorho, and L. Hubert-Moy, "Combined Detection and Segmentation of Archeological Structures from LiDAR Data Using a Deep Learning Approach," *Journal of Computer Applications in Archaeology*, vol. 4, no. 1, p. 1, 2021.
- [61] J. B. Lindsay, J. M. H. Cockburn, and H. A. J. Russell, "An integral image approach to performing multi-scale topographic position analysis," *Geomorphology*, vol. 245, pp. 51–61, 2015.
- [62] Dr. John B. Lindsay © 2017-2021, "WhiteboxTools Version 1.5.0." [Online]. Available: https://jblindsay.github.io/wbt_book/preface.html.
- [63] C. Xu, H. S. He, Y. Hu, Y. Chang, X. Li, and R. Bu, "Latin hypercube sampling and geostatistical modeling of spatial uncertainty in a spatially explicit forest landscape model simulation," *Ecological Modelling*, vol. 185, no. 2–4, pp. 255–269, 2005.

- [64] B. Gao, Y. Pan, Z. Chen, F. Wu, X. Ren, and M. Hu, "A Spatial Conditioned Latin Hypercube Sampling Method for Mapping Using Ancillary Data," *Transactions in GIS*, vol. 20, no. 5, pp. 735–754, 2016.
- [65] M. Niculiță, "Geomorphometric methods for burial mound recognition and extraction from high-resolution LiDAR DEMs," *Sensors (Switzerland)*, vol. 20, no. 4, 2020.
- [66] A. Bonhage, M. Eltaher, T. Raab, M. Breuß, A. Raab, and A. Schneider, "A modified Mask region-based convolutional neural network approach for the automated detection of archaeological sites on high-resolution light detection and ranging-derived digital elevation models in the North German Lowland," *Archaeological Prospection*, no. November 2020, pp. 1–10, 2021.
- [67] M. Bundzel, M. Jaščur, M. Kováč, T. Lieskovský, P. Sinčák, and T. Tkáčik, "Semantic segmentation of airborne lidar data in maya archaeology," *Remote Sensing*, vol. 12, no. 22, pp. 1–22, 2020.
- [68] D. Davis, "Theoretical Repositioning of Automated Remote Sensing Archaeology: Shifting from Features to Ephemeral Landscapes," *Journal of Computer Applications in Archaeology*, vol. 4, no. 1, p. 94, 2021.
- [69] J. Gallant and J. Wilson, "TAPES-G: a grid-based terrain analysis program for the environmental sciences," *Computers and Geosciences*, vol. 22 (7), pp. 713–722, 1996.
- [70] V. Olaya, "Sistemas de Información Geográfica." [Online]. Available: <https://volaya.github.io/libro-sig/>. [Accessed: 20-Jul-2021].
- [71] T. Kohonen, "Self-Organizing Map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [72] F. A. Salazar-vasquez, C. Osorio-serna, and M. A. Caicedo-giraldo, "Identificación de la delimitación administrativa de la malaria usando redes neuronales artificiales Boundary Delimitation of Malaria using Artificial Neural Networks Identificação da Delimitação Administrativa da Malária usando Redes," pp. 11–19.
- [73] J. Demšar *et al.*, "Orange: Data Mining Toolbox in Python," *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013.
- [74] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in {P}ython," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [75] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
- [76] J. A. Sanz Delgado, "Tema 4. Sistemas de clasificación. Árboles de decisión.," in *Sistemas Inteligentes para la extracción de información*, Máster en Sistemas de Información Geográfica y Teledetección (MUSIGT), Ed. Pamplona: ETSIAB. Universidad Pública de Navarra, 2020.
- [77] J. Howard and S. Gugger, *Deep Learning for Coders with fastai and PyTorch*, 1st ed. Sebastopol, 2020.
- [78] Rosaria Silipo, "Ensemble models: Bagging - Boosting," 2020. [Online]. Available: <https://medium.com/analytics-vidhya/ensemble-models-bagging-boosting-c33706db0b0b>. [Accessed: 05-Sep-2021].
- [79] J. A. Sanz Delgado, "Tema 6. Sistemas de clasificación múltiples," in *Sistemas Inteligentes para la extracción de información*, Máster en Sistemas de Información Geográfica y Teledetección (MUSIGT), Ed. Pamplona: ETSIAB. Universidad Pública de Navarra, 2020.
- [80] H. Singh, "Understanding Gradient Boosting Machines," 2018. [Online]. Available: <https://towardsdatascience.com/understanding-gradient-boosting-machines->

- 9be756fe76ab. [Accessed: 22-Aug-2021].
- [81] J. Amat Rodrigo, "Regresión Logística simple y múltiple," 2016. [Online]. Available: https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple. [Accessed: 22-Aug-2021].
- [82] A. Guyot *et al.*, "Combined Detection and Segmentation of Archeological Structures from LiDAR Data Using a Deep Learning Approach To cite this version : HAL Id : hal-03166011 Combined Detection and Segmentation of Archeological Structures from LiDAR Data Using a Deep Learni," 2021.
- [83] W. B. Verschoof-Van Der Vaart, K. Lambers, W. Kowalczyk, and Q. P. J. Bourgeois, "Combining deep learning and location-based ranking for large-scale archaeological prospection of LiDAR data from the Netherlands," *ISPRS International Journal of Geo-Information*, vol. 9, no. 5, 2020.

ANEXOS

Anexo I Cuadernos de código Phyton (Notebooks)

NOTEBOOK 1

Este cuaderno esta construido para llamar a la libreria WhiteboxTools y elaborar para todos los MDT de Navarra una imagen de multiescala basada en los valores DEVmax de cada MDT.

LIBRERÍAS

```
In [1]: from whitebox import WhiteboxTools
import os
import shutil
import imageio
import matplotlib.pyplot as plt
%matplotlib inline
from time import time
```

Construir WhiteboxTools

```
In [2]: wbt = WhiteboxTools()
print(wbt.version())
data_dir = os.path.dirname("/media/alex/DADES2/REPOSITORIO_TFM/MDT/")
wbt.set_working_dir(data_dir)
wbt.verbose = False

data_dir
```

WhiteboxTools v1.4.0 by Dr. John B. Lindsay (c) 2017-2020

WhiteboxTools is an advanced geospatial data analysis platform developed at the University of Guelph's Geomorphometry and Hydrogeomatics Research Group (GHRG). See <https://jblindsay.github.io/ghrG/WhiteboxTools/index.html> for more details.

```
Out[2]: '/media/alex/DADES2/REPOSITORIO_TFM/MDT'
```

Crear una lista para descargar individualmente hojas del MDE del repositorio oficial del Gobierno de Navarra.

```
In [3]: csv = "/home/alex/TFM/listaMDE2017.csv"
file = open(csv, 'r')
contenido = file.readlines()
file.close()
descarga = []
for linea in contenido [0:]:
    linea = linea[:-1]
    celda = linea.split(',')
    hoja = celda[0]
    descarga.append(hoja)
print (len(descarga))
print (descarga[0])
```

400

https://filescartografia.navarra.es/6_MDE/6_3_A%C3%910_2017/6_3_2_MDT/6_3_2_1_ASCII_Grid_EPSG25830/6_3_2_1_1_50cm/MDT_0171_11_2017_EPSG25830_50cm.asc.zip

Preparar las herramientas para iterar la lista de MDT completa o sólo un archivo, reduciendo la longitud del nombre del archivo.

```
In [5]: lista = os.listdir(data_dir)
modificador = []
cont = 0
print (len(lista))
for i in lista:
    if i == 'MDT_0065_21_2017_EPSG25830_50cm.tif':
        print (cont)
        cont += 1
```

401

```
In [6]: for list in lista:
dentro = list[4:11]
modificador.append (dentro)
```

Iterador que llama a las funciones de Whitebox para construir cada una de las imagenes DEVmax, normalizarlas a -3 a 3 y finalmente construir la imagen de multiescala.

```
In [ ]: for i in range (0,len(lista)):
inicio = time()
min_val = -3
max_val = 3

wbt.max_elevation_deviation(lista[i],
"micro_mag.tif",
```

```

        "micro_dev.tif",
        1,
        10,
        step=90)
local = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/micro_mag.tif"
local_dev = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/micro_mag.tif"
local_save = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/"+str(modificador[i])+".tiff"

wbt.min_max_contrast_stretch(
    local_dev,
    local_save,
    min_val,
    max_val,
    num_tones=256)#normaliza a 8 bits
local_fin = "/media/alex/DADES2/REPOSITORIO_TFM/DEV_LOCAL/"+str(modificador[i])+".tiff"
shutil.move(local_save,local_fin)
os.remove (data_dir+"/micro_dev.tif")

wbt.max_elevation_deviation(lista[i],
    "meso_mag.tif",
    "meso_dev.tif",
    10,
    100,
    step=90)
meso = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/meso_mag.tif"
meso_dev = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/meso_mag.tif"
meso_save = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/"+str(modificador[i])+".tiff"

wbt.min_max_contrast_stretch(
    meso_dev,
    meso_save,
    min_val,
    max_val,
    num_tones=256)#normaliza a 8 bits
meso_fin = "/media/alex/DADES2/REPOSITORIO_TFM/DEV_MESO/"+str(modificador[i])+".tiff"
#shutil.copy(meso_dev,meso_save)
shutil.move(meso_save,meso_fin)
os.remove (data_dir+"/meso_dev.tif")

wbt.max_elevation_deviation(lista[i],
    "macro_mag.tif",
    "macro_dev.tif",
    100,
    1000,
    step=90)

broad = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/macro_mag.tif"
macro_dev = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/macro_mag.tif"
macro_save = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/"+str(modificador[i])+".tiff"

wbt.min_max_contrast_stretch(
    macro_dev,
    macro_save,
    min_val,
    max_val,
    num_tones=256)#normaliza a 8 bits
macro_fin = "/media/alex/DADES2/REPOSITORIO_TFM/DEV_MACRO/"+str(modificador[i])+".tiff"
#shutil.copy(macro_dev,macro_save)
shutil.move(macro_save,macro_fin)
os.remove (data_dir+"/macro_dev.tif")

output = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/output.tif"

wbt.multiscale_topographic_position_image(
    local,
    meso,
    broad,
    output,
    lightness=1.2)

mstpi_save = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/"+str(modificador[i])+".tiff"
mstpi_fin = "/media/alex/DADES2/REPOSITORIO_TFM/MSTPI/"+str(modificador[i])+".tiff"
shutil.copy(output,mstpi_save)
shutil.move(mstpi_save,mstpi_fin)
os.remove (data_dir+"/output.tif")
os.remove (data_dir+"/micro_mag.tif")
os.remove (data_dir+"/meso_mag.tif")
os.remove (data_dir+"/macro_mag.tif")

fin = (time() - inicio)/60

print("Archivo "+str(modificador[i])+" (" +str(i)+"/"+str(len(lista))+")"+" creado en: %.2f minutos" %fin)

```

NOTEBOOK 2

Este cuaderno esta construido para llamar a la libreria WhiteboxTools y elaborar para todos los MDT de Navarra una imagen orientación.

LIBRERÍAS

```
In [1]: import whitebox as WBT
import os
import shutil
import imageio
import matplotlib.pyplot as plt
%matplotlib inline
from time import time
```

Construir WhiteboxTools

```
In [2]: wbt = WBT.WhiteboxTools()
print(wbt.version())
data_dir = os.path.dirname("/media/alex/DADES2/REPOSITORIO_TFM/MDT/")
wbt.set_working_dir(data_dir)
wbt.verbose = False

data_dir
```

WhiteboxTools v1.4.0 by Dr. John B. Lindsay (c) 2017-2020

WhiteboxTools is an advanced geospatial data analysis platform developed at the University of Guelph's Geomorphometry and Hydrogeomatics Research Group (GHRG). See <https://jblindsay.github.io/ghrg/WhiteboxTools/index.html> for more details.

```
Out[2]: '/media/alex/DADES2/REPOSITORIO_TFM/MDT'
```

Preparar las herramientas para iterar la lista de MDT completa y reducir la longitud del nombre del archivo

```
In [11]: lista = os.listdir(data_dir)
modificador = []
cont = 0
print (len(lista))
```

401

```
In [12]: for list in lista:
dentro = list[4:11]
modificador.append (dentro)
```

Iterador que llama a la función de Whitebox para construir las imagenes de orientación.

```
In [ ]: for i in range (0,len(lista)):
inicio = time()
wbt.aspect(lista[i],
"salida.tiff",
zfactor=None)
original = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/salida.tiff"
guardado = "/media/alex/DADES2/REPOSITORIO_TFM/MDT/"+str(modificador[i])+".tiff"
final = "/media/alex/DADES2/REPOSITORIO_TFM/WMS_ORIENTACION/"+str(modificador[i])+".tiff"
shutil.copy(original,guardado)
shutil.move(guardado,final)
os.remove (original)

fin = (time() - inicio)/60

print("Archivo "+str(modificador[i])+" (" +str(i)+"/"+str(len(lista))+")"+" creado en: %.2f minutos" %fin)
```


NOTEBOOK 3

Con este cuaderno se adquiere la información categórica de todas las fichas del Inventario Arqueológico de Navarra. Se trata de un ejercicio de *web scrapping*.

LIBRERÍAS

```
In [2]: import os
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

Declarar la ruta de ubicación de los archivos donde se encuentran la información del Inventario Arqueológico de Navarra.

```
In [3]: path= "/home/alex/TFM/YACIMIENTOS/HTML/"
lista = os.listdir(path)
len(lista)
```

Out[3]: 9375

Declarar todas las variables de las que se requiere adquirir información. Las variables se declaran en formato de lista para luego anexar (append) la información. En algunas de las variables es necesario declarar previamente cual es su contenido para luego simplemente decirle al *script* que la información correcta se encuentran en una u otra posición de dicha lista.

```
In [8]: yacimiento=[]
clasificacion_cultural = []
tipologia = []
actividad = []
superficie = []
emplazamiento = []
entorno = []
grado_destruccion = []
causas_deterioro = []
materiales = []
ficha = []
paraje = []
year = []

causas_control = ['Clandestinos', 'Labores agrícolas', 'Erosión']
clas_control = ['Paleolítico inferior',
                'Paleolítico medio',
                'Paleolítico superior',
                'Epipaleolítico',
                'Neolítico',
                'Eneolítico',
                'Edad del Bronce',
                'Edad del Hierro',
                'Época Romana',
                'Tardoantigüedad',
                'Edad Media',
                'Edad Moderna',
                'Edad Contemporánea',
                'Indeterminado']
emp_control = ['Ladera', 'Planicie', 'Terraza', 'Cumbre', 'Urbano', 'Desfiladero', 'Subacuática']
tipo_control = ['Aire libre',
                'Núcleo de población',
                'Villa-Caserio',
                'Dolmen',
                'Cromlech',
                'Transformación',
                'Túmulo',
                'Explotación',
                'Recinto militar',
                'Monolito - Menhir',
                'Ermita',
                'Necrópolis',
                'Vías',
                'Cueva',
                'Presa',
                'Puente',
                'Abrigo',
                'Circulación',
                'Monasterio-Iglesia',
                'Tumba aislada',
                'Murralla',
                'Santuario',
                'Cueva rupestre',
                'Acueducto',
                'Edificio público']
```

Mediante un iterador *for* se realiza una búsqueda de la información a adquirir. Es necesario convertir en primer lugar la información en un objeto de la librería Beautiful Soup, para luego poder utilizar las funciones de dicha librería y anexar la información a la lista correspondiente.

información a la lista correspondiente.

```
In [4]: for i in lista:
with open (path + str(i)) as fp:
    soup = BeautifulSoup(fp, 'html.parser')

#CODIGO YACIMIENTO
try:
    codigo = soup.find_all('table', class_='encabezado-table')[0].get_text()
    yacimiento.append (codigo[18:32])
except:
    yacimiento.append('Error')
    pass

#PARAJE
try:
    ubicar = soup.find_all('table', class_='encabezado-table')[1]
    ubicado = ubicar.find('td', colspan = '3').get_text()
    paraje.append(ubicado)
except:
    paraje.append('Error')
    pass

#CLASIFICACIÓN CULTURAL
try:
    ubicar = soup.find_all('h2')[1]
    ubicado = ubicar.find_next('p', class_='text').get_text()
    clas = ubicado.split(',')
    if clas[0] in clas_control:
        clasificacion_cultural.append(ubicado)
    else:
        clasificacion_cultural.append('Desconocida')
except:
    clasificacion_cultural.append('Error')
    pass

#ACTIVIDAD
try:
    ubicar = soup.find_all('h2')[2]
    ubicado = ubicar.find_next('h3').get_text()[6:]
    if ubicado == 'Dimensiones':
        actividad.append('Desconocida')
    else:
        actividad.append(ubicado)
except:
    actividad.append('Error')

#TIPOLOGÍA
try:
    ubicar = soup.find_all('h2')[2]
    ubicado = ubicar.find_next('p', class_='text').get_text()
    tipo = ubicado.split(',')
    if tipo[0] in tipo_control:
        tipologia.append(ubicado)
    else:
        tipologia.append('Desconocida')
except:
    tipologia.append('Error')

#SUPERFÍCIE
try:
    sup = soup.find_all('table', class_='encabezado-table')[2].get_text()
    dato = sup.split()
    if dato[0] == 'Superficie':
        superficie.append (dato[1])
    else:
        superficie.append ('Desconocida')
except:
    superficie.append('Error')
    pass

#EMPLAZAMIENTO
try:
    ubicar = soup.find_all('h2')[3]
    ubicado = ubicar.find_next('h3')
    ubicacion = ubicado.find_next('p', class_='text').get_text()
    emp = ubicacion.split(',')
    if emp[0] in emp_control:
        emplazamiento.append(ubicacion)
    else:
        emplazamiento.append('Desconocido')
except:
    emplazamiento.append('Error')
    pass

#ENTORNO
try:
    ubicar = soup.find_all('h3')[3]
    ubicado = ubicar.find_next('p', class_='text').get_text()
    if ubicado.isupper():
```

```

        entorno.append(ubicado)
    else:
        entorno.append('Desconocido')
except:
    entorno.append('Error')
    pass

#GRADO DE DESTRUCCIÓN
try:
    ubicar = soup.find_all('h2')[5]
    ubicado = ubicar.find_next('h3')
    ubicacion = ubicado.find_next('p', class_='text').get_text()
    d = ubicacion.split()
    grado_destruccion.append(d[2])
except:
    grado_destruccion.append('Error')
    pass

#CAUSAS DETERIORO
try:
    ubicar = soup.find_all('h3')[5]
    ubicado = ubicar.find_next('h3')
    ubicacion = ubicado.find_next('p', class_='text').get_text()
    e = ubicacion.split(',')
    if e[0] in causas_control:
        causas_deterioro.append(ubicacion)
    else:
        causas_deterioro.append('Desconocidas')
except:
    causas_deterioro.append('Error')
    pass

#MATERIALES
try:
    comprobar = soup.find_all('h3')
    if len(comprobar) > 11:
        ubicar = soup.find_all('h3')[11]
        ubicado = ubicar.find_next('p', class_='text').get_text()
        if ubicado.isupper():
            materiales.append(ubicado)
        else:
            materiales.append('Desconocidos')
    else:
        materiales.append('Desconocidos')
except:
    materiales.append('Error')
    pass

#FICHA
try:
    ubicar = soup.find_all('h2')[9]
    ubicado = ubicar.find_next('table', class_='encabezado-table')
    ubicacion = ubicado.find_next('td')
    nombre = ubicacion.find_next('td').get_text()
    ficha.append(nombre)
except:
    ficha.append('Error')
    pass

#AÑO
try:
    ubicar = soup.find_all('h2')[9]
    ubicado = ubicar.find_next('table', class_='encabezado-table')
    ubicacion = ubicado.find_next('td')
    nombre = ubicacion.find_next('td')
    fecha = nombre.find_next('td')
    data = fecha.find_next('td').get_text()
    year.append(data[:4])
except:
    year.append('Error')
    pass

```

Adquirida la información se convierte la misma en un *DataFrame* de la librería Pandas. La función *DataFrame* se construye como si fuese un diccionario de Python, es decir, una lista de listas.

```

In [5]: IAN2021 = pd.DataFrame({'Codyaci': yacimiento,
                             'Paraje': paraje,
                             'Clasificacion cultural': clasificacion_cultural,
                             'Actividad': actividad,
                             'Tipologia': tipologia,
                             'Superficie': superficie,
                             'Emplazamiento': emplazamiento,
                             'Entorno': entorno,
                             'Destrucción': grado_destruccion,
                             'Causas destrucción': causas_deterioro,
                             'Materiales': materiales,
                             'Ficha': ficha,
                             'Año': year}).

```

```
index= list(range(0,len(lista)))
```

En última instancia se crea un archivo .csv con la función `to_csv` de Pandas.

```
In [6]: IAN2021.to_csv('/home/alex/TFM/YACIMIENTOS_2020.csv', index=False)
```

NOTEBOOK 4

En este cuaderno se realiza el entrenamiento de diferentes clasificadores para encontrar la mejor configuración.

LIBRERÍAS

```
In [ ]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from pandas import read_csv
from timeit import default_timer as timer

from sklearn.preprocessing import StandardScaler
from sklearn import model_selection
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from imblearn.metrics import geometric_mean_score
from sklearn.metrics import classification_report
from sklearn.metrics import cohen_kappa_score
from sklearn import metrics

from sklearn.metrics import plot_roc_curve
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
from sklearn.metrics import average_precision_score

from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn import neighbors
from sklearn.linear_model import LogisticRegression
from sklearn import tree

import warnings

warnings.filterwarnings('ignore')
```

DATASET

Declarar la localización de los datos

```
In [ ]: data_dir = os.path.dirname('/home/alex/TFM/YACIMIENTOS/CSV/ventanas_balanceadas/')
lista = os.listdir(data_dir)
```

Lista para almacenar resultados

```
In [ ]: resultados = []
```

ENTRENAMIENTO

El entrenamiento se realiza con *scikit learn* y se estructura de la siguiente manera:

- 1.- Leer el dataset correspondiente
- 2.- Separar las variables continuas (X) y las clases (Y)
- 3.- Crear un conjunto de entrenamiento y otro de validación
- 4.- Normalizar los datos de entrenamiento
- 5.- Declarar las métricas de evaluación
- 6.- Buscar la mejor configuración para un clasificador tipo Random Forest
- 7.- Buscar la mejor configuración para un clasificador tipo Gradient Boosting
- 8.- Buscar la mejor configuración para un clasificador tipo Regresión Logística

```
In [ ]: for i in range(0, len(lista)): #Len(Lista)

dataset = pd.read_csv(data_dir+'/' +str(lista[i]), sep=',', decimal = '.', na_values= '')
dataset = dataset.fillna(dataset.mean())

atributos = []

for var in list(dataset.columns):
    if var[0].isdigit():
```

```

    atributos.append(var)

atributos.append('CLASE')

variables = dataset.loc[:,atributos]

X = np.array(variables.loc[:, '1_mean': '11_majority'])
Y = np.array(variables.loc[:, 'CLASE'])

scaler = StandardScaler()

np.random.seed(12)

X_train, X_test, y_train, y_test = model_selection.train_test_split(X, Y, train_size=0.7)
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

np.random.seed(12)

start = timer()
scoring = {'AUC': 'roc_auc', 'F1 score': 'f1', 'Precision': 'precision', 'Recall': 'recall', 'B_Accuracy': 'balanced_accuracy'}

#### RANDOM FOREST #####
model = RandomForestClassifier(random_state=98,
                              min_samples_leaf = 1,
                              min_samples_split = 2,
                              n_jobs = -1
                              )
param_grid = {'criterion': ['entropy', 'gini'],
              'n_estimators': [10, 50, 100, 150, 200],
              'max_features': ['sqrt', 'log2', None]}

RF = model_selection.GridSearchCV(model, param_grid, scoring= scoring, refit = 'Recall', cv=10, return_train_score=True)
RF = RF.fit(X_train, y_train)

RF_Train =RF.predict(X_train)
gmTrain = geometric_mean_score(y_train,RF_Train)

RF_Test = RF.predict(X_test)
gmTest = geometric_mean_score(y_test, RF_Test)

c1_probs = RF.predict_proba(X_test)
c1_probs = c1_probs[:, 1]
c1_auc = metrics.roc_auc_score(y_test, c1_probs)
lr_f1 = metrics.f1_score(y_test, RF_Test)

kappa = cohen_kappa_score(y_test, RF_Test)

indicekappa = ['Pobre', 'Débil', 'Moderado', 'Bueno', 'Muy Bueno']
verindicekappa = []
if kappa < 0.2:
    verindicekappa = indicekappa[0]
elif kappa > 0.2 and kappa < 0.4:
    verindicekappa = indicekappa[1]
elif kappa > 0.4 and kappa < 0.6:
    verindicekappa = indicekappa[2]
elif kappa > 0.6 and kappa < 0.8:
    verindicekappa = indicekappa[3]
else:
    verindicekappa = indicekappa[4]

resultados.append((lista[i],model, RF.best_params_,round(RF.best_score_,2),round(gmTest*100.0, 2),round(c1_auc*100.0, 2), round(lr_f1

tiempo_aprender = timer() - start

print ("Archivo: {} *** Metodo: {} *** Tiempo: {} segundos *** kappa: {}".format(lista[i], model, round(tiempo_aprender,2), verindice

#### GRADIENT BOOSTING #####
start = timer()

model = GradientBoostingClassifier(learning_rate=1,
                                   max_depth=1,
                                   random_state=0)

param_grid = {'n_estimators': [10, 50, 100, 150, 200],
              'max_features': ['sqrt', 'log2', None]}

GB = model_selection.GridSearchCV(model, param_grid, scoring=scoring, refit= 'Recall' , cv=10, return_train_score=True)
GB = GB.fit(X_train, y_train)

GB_Train =GB.predict(X_train)
gmTrain = geometric_mean_score(y_train,GB_Train)

GB_Test = GB.predict(X_test)
gmTest = geometric_mean_score(y_test, GB_Test)

```

```

c1_probs = GB.predict_proba(X_test)
c1_probs = c1_probs[:, 1]
c1_auc = metrics.roc_auc_score(y_test, c1_probs)
lr_f1 = metrics.f1_score(y_test, GB_Test)

kappa = cohen_kappa_score(y_test, GB_Test)

indicekappa = ['Pobre', 'Débil', 'Moderado', 'Bueno', 'Muy Bueno']
verindicekappa = []
if kappa < 0.2:
    verindicekappa = indicekappa[0]
elif kappa > 0.2 and kappa < 0.4:
    verindicekappa = indicekappa[1]
elif kappa > 0.4 and kappa < 0.6:
    verindicekappa = indicekappa[2]
elif kappa > 0.6 and kappa < 0.8:
    verindicekappa = indicekappa[3]
else:
    verindicekappa = indicekappa[4]

resultados.append((lista[i], model, GB.best_params_, round(GB.best_score_, 2), round(gmTest*100.0, 2), round(c1_auc*100.0, 2), round(lr_f1

tiempo_aprender = timer() - start

print ("Archivo: {} *** Metodo: {} *** Tiempo: {} segundos *** kappa: {}".format(lista[i], model, round(tiempo_aprender, 2), verindice

#### LOGISTIC REGRESION ####

start = timer()

model = LogisticRegression(multi_class= 'ovr', n_jobs = -1, random_state = 48)

param_grid = {'penalty':['l1', 'l2', 'elasticnet', 'none'], 'solver':['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'], 'max_iter':[100,

LR = model_selection.GridSearchCV(model, param_grid, scoring=scoring, refit= 'Recall' , cv=10, return_train_score=True)
LR = LR.fit(X_train, y_train)

LR_Train =LR.predict(X_train)
gmTrain = geometric_mean_score(y_train, LR_Train)

LR_Test = LR.predict(X_test)
gmTest = geometric_mean_score(y_test, LR_Test)

c1_probs = LR.predict_proba(X_test)
c1_probs = c1_probs[:, 1]
c1_auc = metrics.roc_auc_score(y_test, c1_probs)
lr_f1 = metrics.f1_score(y_test, LR_Test)

kappa = cohen_kappa_score(y_test, LR_Test)

indicekappa = ['Pobre', 'Débil', 'Moderado', 'Bueno', 'Muy Bueno']
verindicekappa = []
if kappa < 0.2:
    verindicekappa = indicekappa[0]
elif kappa > 0.2 and kappa < 0.4:
    verindicekappa = indicekappa[1]
elif kappa > 0.4 and kappa < 0.6:
    verindicekappa = indicekappa[2]
elif kappa > 0.6 and kappa < 0.8:
    verindicekappa = indicekappa[3]
else:
    verindicekappa = indicekappa[4]

resultados.append((lista[i], model, LR.best_params_, round(LR.best_score_, 2), round(gmTest*100.0, 2), round(c1_auc*100.0, 2), round(lr_f1

tiempo_aprender = timer() - start

print ("Archivo: {} *** Metodo: {} *** Tiempo: {} segundos *** kappa: {}".format(lista[i], model, round(tiempo_aprender, 2), verindice

```

Declarar tantas listas como posiciones tenga la lista resultados donde se han almacenado los valores resultantes del entrenamiento.

```
In [51]: ventana, metodo, config, score, GM, ROC, F1, K, tipo = [], [], [], [], [], [], [], [], []
```

Asignar valores

```
In [53]: for i in range (0, len(resultados)):
    ventana.append(resultados[i][0])
    metodo.append(resultados[i][1])
    config.append(resultados[i][2])
    score.append(resultados[i][3])
    GM.append(resultados[i][4])
    ROC.append(resultados[i][5])
    F1.append(resultados[i][6])
    K.append(resultados[i][7])
    tipo.append(resultados[i][8])
```

```
score.append(resultados [i][3])
GM.append(resultados [i][4])
ROC.append(resultados [i][5])
F1.append(resultados [i][6])
K.append(resultados [i][7])
tipo.append(resultados [i][8])
```

Crear el *DataFrame*

```
In [59]: for i in range (0,len(resultados)):
         DF = pd.DataFrame({'Ventana': ventana,
                           'Método' : metodo,
                           'Mejor configuración': config,
                           'Mejor puntuación' : score,
                           'Media Geométrica': GM,
                           'ROC AUC' : ROC,
                           'F1 score' : F1,
                           'Índice Kappa': K,
                           'Tipo de clasificador': tipo},
                           index= list(range(0,len(resultados))))
```

Guardar como .csv

```
In [61]: DF.to_csv('/home/alex/TFM/DOCUMENTO/entrenamiento.csv')
```


NOTEBOOK 5

Este cuaderno esta diseñado para buscar la mejor configuración de un clasificador basado en Random Forest.

LIBRERÍAS

```
In [1]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from pandas import read_csv
from timeit import default_timer as timer
import warnings

warnings.filterwarnings('ignore')
```

DATASET

Leer la ubicación de los datos

```
In [2]: data_dir = os.path.dirname('/home/alex/TFM/YACIMIENTOS/CSV/ventanas_balanceadas/')
lista = os.listdir(data_dir)
```

Seleccionar el número de la ventana de observación

```
In [3]: f = 15
```

Eligir el archivo para calcular los mejor configuración

```
In [4]: cont = 0
for i in lista:
    if i == 'BALANCED_G'+str(f)+'.csv':
        n = cont
    else:
        cont += 1

lista[n]
```

```
Out[4]: 'BALANCED_G15.csv'
```

Leer el archivo

```
In [5]: dataset = pd.read_csv(data_dir+'/'+str(lista[n]),sep=',', decimal = '.',na_values= '')
dataset.shape
```

```
Out[5]: (152, 55)
```

Convertir los valores de NaN a valores de media

```
In [6]: dataset = dataset.fillna(dataset.mean())
```

Ver la estructura del dataset

```
In [7]: dataset.columns
```

```
Out[7]: Index(['fid', 'Group', 'AREA', 'CLASE', 'NOMBRE', 'COD_YACI', 'REDACTOR_FICHA',
'MUNICIPIO', '1_median', '1_stdev', '1_majority', '2_mean', '2_median',
'2_stdev', '2_majority', '3_mean', '3_median', '3_stdev', '3_majority',
'4_mean', '4_median', '4_stdev', '4_majority', '5_mean', '5_median',
'5_stdev', '5_majority', '1_mean', 'TIPO', 'TIPOLOGIA', 'CRONOLOGIA',
'6_mean', '6_median', '6_stdev', '6_majority', '7_mean', '7_median',
'7_stdev', '7_majority', '8_mean', '8_median', '8_stdev', '8_majority',
'9_mean', '9_median', '9_stdev', '9_majority', '10_mean', '10_median',
'10_stdev', '10_majority', '11_mean', '11_median', '11_stdev',
'11_majority'],
dtype='object')
```

Leer las variables continuas y categóricas implicadas en la clasificación

```
In [8]: atributos = []
for var in list(dataset.columns):
    if var[0].isdigit():
        atributos.append(var)

atributos.append('CLASE')
print(atributos)
```



```

kappa = cohen_kappa_score(y_test, RF_Test)

indicekappa = ['Pobre', 'Débil', 'Moderado', 'Bueno', 'Muy Bueno']
verindicekappa = []
if kappa < 0.2:
    verindicekappa = indicekappa[0]
elif kappa > 0.2 and kappa < 0.4:
    verindicekappa = indicekappa[1]
elif kappa > 0.4 and kappa < 0.6:
    verindicekappa = indicekappa[2]
elif kappa > 0.6 and kappa < 0.8:
    verindicekappa = indicekappa[3]
else:
    verindicekappa = indicekappa[4]

model_probs = RF.predict_proba(X_test)
model_probs = model_probs[:, 1]

ns_probs = [0 for _ in range(len(y_test))]
ns_auc = metrics.roc_auc_score(y_test, ns_probs)*100.0

model_auc = metrics.roc_auc_score(y_test, model_probs)*100.0
ns_fpr, ns_tpr, _ = metrics.roc_curve(y_test, ns_probs)
model_fpr, model_tpr, _ = metrics.roc_curve(y_test, model_probs)

no_skill = len(y_test[y_test==1]) / len(y_test)
model_precision, model_recall, _ = metrics.precision_recall_curve(y_test, model_probs)
model_f1 = metrics.f1_score(y_test, RF_Test)*100.0
model_auc_PR = metrics.auc(model_recall, model_precision)

tiempo_RF_aprender = timer() - start

print ("Calculados los mejores hiperparámetros en %f segundos" % tiempo_RF_aprender )

```

Calculados los mejores hiperparámetros en 188.068458 segundos

RESULTADOS

MULTIPLE EVALUATION METRICS

Este fragmento del cuaderno forma parte del manual de *scikit learn*

```
In [14]: results = RF.cv_results_
```

```
In [15]: plt.figure(figsize=(13, 13))
plt.title("GridSearchCV evaluating using multiple scorers simultaneously",
          fontsize=16)

plt.xlabel("n_estimators")
plt.ylabel("Score")

ax = plt.gca()
ax.set_xlim(0, 250)
ax.set_ylim(0.5, 1.10)

# Get the regular numpy array from the MaskedArray
X_axis = np.array(results['param_n_estimators'].data, dtype=float)

for scorer, color in zip(sorted(scoring), ['g', 'k', 'b', 'r', 'y']):
    for sample, style in (('train', '--'), ('test', '-')):
        sample_score_mean = results['mean_%s_%s' % (sample, scorer)]
        sample_score_std = results['std_%s_%s' % (sample, scorer)]
        ax.fill_between(X_axis, sample_score_mean - sample_score_std,
                        sample_score_mean + sample_score_std,
                        alpha=0 if sample == 'test' else 0, color=color)
        ax.plot(X_axis, sample_score_mean, style, color=color,
                alpha=0.25 if sample == 'test' else 0.7,
                label="%s (%s)" % (scorer, sample))

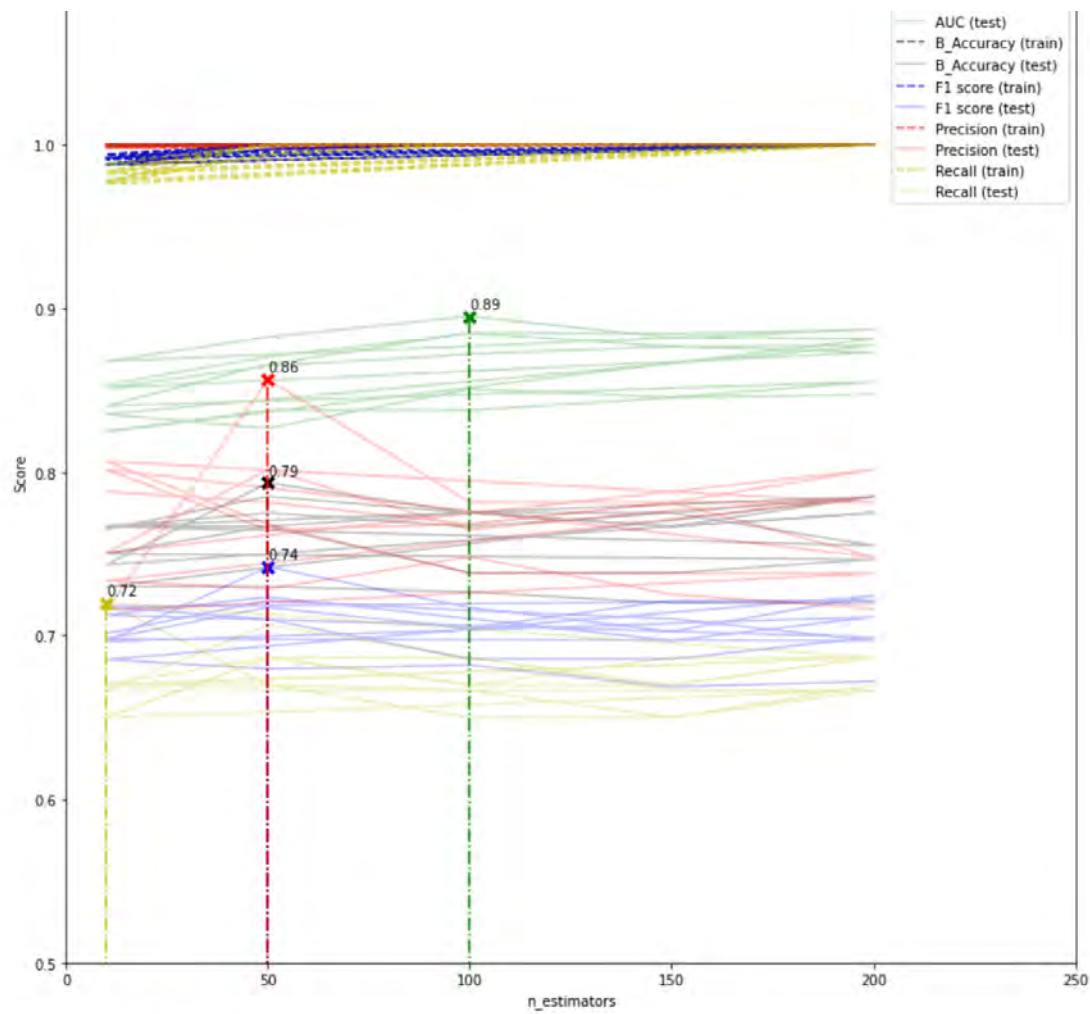
    best_index = np.nonzero(results['rank_test_%s' % scorer] == 1)[0][0]
    best_score = results['mean_test_%s' % scorer][best_index]

    # Plot a dotted vertical line at the best score for that scorer marked by x
    ax.plot([X_axis[best_index], ] * 2, [0, best_score],
            linestyle='-.', color=color, marker='x', markeredgewidth=3, ms=8)

    # Annotate the best score for that scorer
    ax.annotate("%0.2f" % best_score,
                (X_axis[best_index], best_score + 0.005))

plt.legend(loc="best")
plt.grid(False)
plt.show()

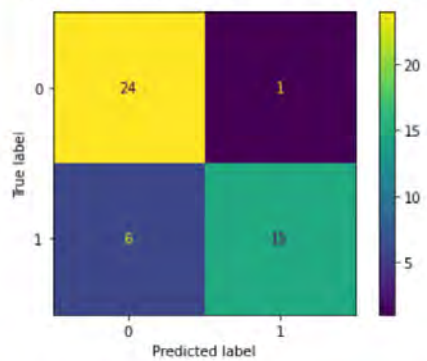
```



MATRIZ DE CONFUSIÓN

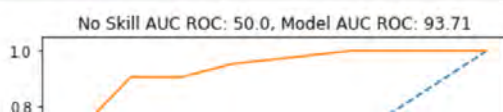
In [16]: `disp.plot()`

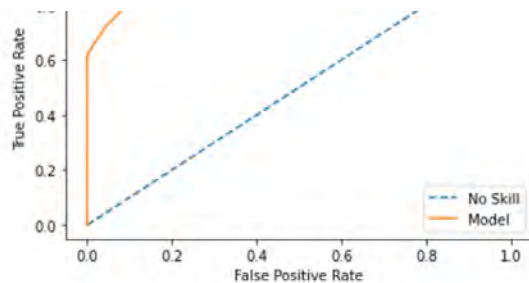
Out[16]: `<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f6bc7d979e8>`



ÁREA BAJO LA CURVA ROC

```
In [17]: plt.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
plt.plot(model_fpr, model_tpr, label='Model')
# Etiquetas de Los ejes
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Leyenda
plt.legend()
# Titulo
plt.title('No Skill AUC ROC: {}, Model AUC ROC: {}'.format(round(ns_auc,2), round(model_auc,2)))
# Mostramos la figura
plt.show()
```

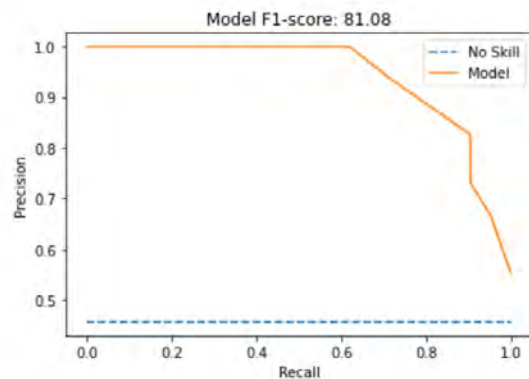




AREA BAJO LA CURVA Precision - Recall (PR)

```
In [18]: # Mostramos visualmente La curva PR

plt.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill')
plt.plot(model_recall, model_precision, label='Model')
# Etiquetas de los ejes
plt.xlabel('Recall')
plt.ylabel('Precision')
# Leyenda
plt.legend()
# Titulo
plt.title('Model F1-score: {}'.format(round(model_f1,2)))
# Mostramos la figura
plt.show()
```



Resultados en formato de texto

```
In [19]: print('Archivo: {} \nMetodo: {} \nMejores Hiperparámetros: {} \nMejor puntuación: {} \nRendimiento: (GM) {} \nROC AUC: {} \nModel AUC PR: {} \n
RF.best_params_,
round(RF.best_score_,2),
round(gmTest*100.0, 2),
round(c1_auc*100.0, 2),
round(model_auc_PR*100,2),
round(lr_f1*100.0, 2),
round(kappa,2),
verindicekappa))
```

```
Archivo: BALANCED_G15.csv
Metodo: RandomForestClassifier(n_jobs=-1, random_state=98)
Mejores Hiperparámetros: {'criterion': 'gini', 'max_features': 'sqrt', 'n_estimators': 10}
Mejor puntuación: 0.72
Rendimiento: (GM) 82.81
ROC AUC: 93.71
Model AUC PR: 94.16
F1 score: 81.08
Índice de Kappa: 0.69
Clasificador: Bueno
```

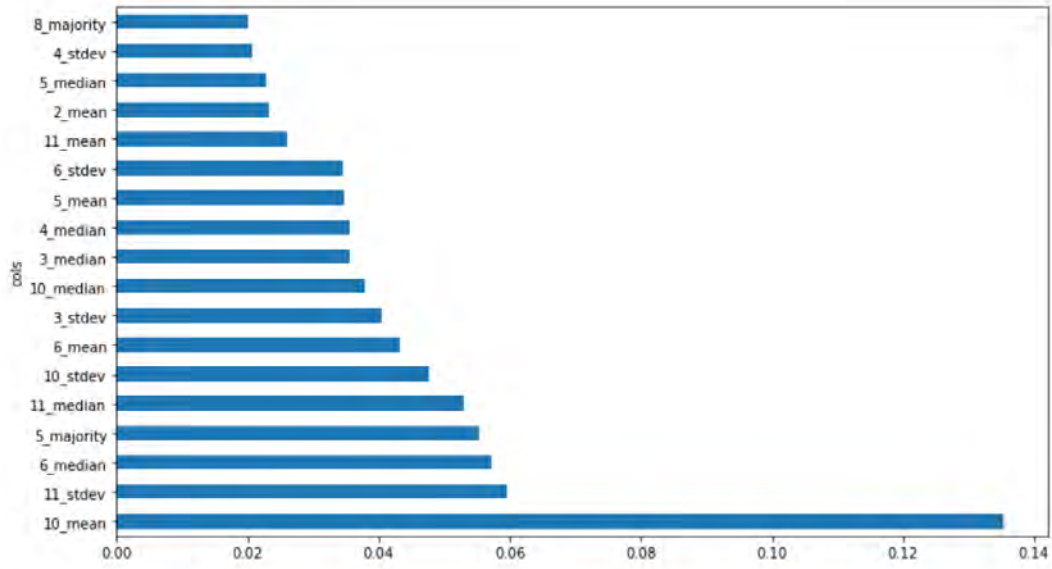
Fragmento de código recopilado del manual de la librería Fastai, a partir del cual se crean dos funciones para analizar la importancia de cada variable continua del clasificador.

```
In [20]: def importancia (m,df):
return pd.DataFrame({'cols':df.columns, 'imp': m.best_estimator_.feature_importances_}).sort_values('imp', ascending = False)

def plot_importancia (imp):
return imp.plot('cols', 'imp', 'barh', figsize = (12,7), legend=False)
```

```
In [21]: z = (variables.loc[:, 'l_median': 'l1_majority'])
imp = importancia (RF,z)
to_feet = imp[:20]
to_keep = imp[imp.imp > 0.02]
plot_importancia(to_keep)
```

Out[21]: <AxesSubplot:ylabel='cols'>



Exportar el modelo

```
In [22]: import joblib
joblib.dump(RF, '/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELOS/modelo_G'+str(f)+'.pk1')
```

Out[22]: ['/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELOS/modelo_G15.pk1']

In []:

NOTEBOOK 6

Este cuaderno esta diseñado para buscar la mejor configuración de un clasificador basado en ensemble de tipo Gradient Boosting.

LIBRERÍAS

```
In [1]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from pandas import read_csv
from timeit import default_timer as timer
import warnings

warnings.filterwarnings('ignore')
```

DATASET

Leer la ubicación de los datos

```
In [2]: data_dir = os.path.dirname('/home/alex/TFM/YACIMIENTOS/CSV/ventanas_balanceadas/')
lista = os.listdir(data_dir)
```

Seleccionar el número de la ventana de observación

```
In [3]: f = 15
```

Eligir el archivo para calcular los mejor configuración

```
In [4]: cont = 0
for i in lista:
    if i == 'BALANCED_G'+str(f)+'.csv':
        n = cont
    else:
        cont += 1

lista[n]
```

```
Out[4]: 'BALANCED_G15.csv'
```

Leer el archivo

```
In [5]: dataset = pd.read_csv(data_dir+'/'+str(lista[n]), sep=',', decimal = '.', na_values= '')
dataset.shape
```

```
Out[5]: (152, 55)
```

Convertir los valores de NaN a valores de media

```
In [6]: dataset = dataset.fillna(dataset.mean())
```

```
In [7]: dataset.columns
```

```
Out[7]: Index(['fid', 'Group', 'AREA', 'CLASE', 'NOMBRE', 'COD_YACI', 'REDACTOR_FICHA',
'MUNICIPIO', '1_median', '1_stdev', '1_majority', '2_mean', '2_median',
'2_stdev', '2_majority', '3_mean', '3_median', '3_stdev', '3_majority',
'4_mean', '4_median', '4_stdev', '4_majority', '5_mean', '5_median',
'5_stdev', '5_majority', '1_mean', 'TIPO', 'TIPOLOGIA', 'CRONOLOGIA',
'6_mean', '6_median', '6_stdev', '6_majority', '7_mean', '7_median',
'7_stdev', '7_majority', '8_mean', '8_median', '8_stdev', '8_majority',
'9_mean', '9_median', '9_stdev', '9_majority', '10_mean', '10_median',
'10_stdev', '10_majority', '11_mean', '11_median', '11_stdev',
'11_majority'],
dtype='object')
```

Leer las variables continuas y categóricas implicadas en la clasificación

```
In [8]: atributos = []
for var in list(dataset.columns):
    if var[0].isdigit():
        atributos.append(var)
atributos.append('CLASE')
print(atributos)
```

```
[ '1_median', '1_stdev', '1_majority', '2_mean', '2_median', '2_stdev', '2_majority', '3_mean', '3_median', '3_stdev', '3_majority', '4_mean', '4_median', '4_stdev', '4_majority', '5_mean', '5_median', '5_stdev', '5_majority', '6_mean', '6_median', '6_stdev', '6_majority', '7_mean', '7_median', '7_stdev', '7_majority', '8_mean', '8_median', '8_stdev', '8_majority', '9_mean', '9_median', '9_stdev', '9_majority', '10_mean', '10_median', '10_stdev', '10_majority', '11_mean', '11_median', '11_stdev', '11_majority', 'CLASE']
```

```
In [9]: variables = dataset.loc[:,atributos]
variables.shape
```

```
Out[9]: (152, 45)
```

Crear los atributos X, Y

```
In [10]: X = np.array(variables.loc[:, '1_median': '11_majority'])
Y = np.array(variables.loc[:, 'CLASE'])
```

```
In [11]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

ENTRENAMIENTO

Separar el dataset en dos grupos. Un grupo para el entrenamiento y el otro para la validación. Así como normalizar los datos de entrenamiento y ajustar los datos de validación a la dimensión de los datos de entrenamiento.

```
In [12]: from sklearn import model_selection
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from imblearn.metrics import geometric_mean_score
from sklearn.metrics import classification_report
from sklearn.metrics import cohen_kappa_score
from sklearn import metrics

np.random.seed(12)

X_train, X_test, y_train, y_test = model_selection.train_test_split(X, Y, train_size=0.7)
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

GRADIENT BOOSTING

Buscar la mejor configuración con GridSearchCV

```
In [13]: from sklearn.ensemble import GradientBoostingClassifier

#Inicio contador tiempo
start = timer()
np.random.seed(12)
#constructor y parametros

scoring = {'AUC': 'roc_auc', 'Accuracy': 'balanced_accuracy', 'Precision': 'precision', 'Recall': 'recall', 'F1 score': 'f1'}

model = GradientBoostingClassifier(learning_rate=1,
                                  max_depth=1,
                                  random_state=0)

param_grid = {'n_estimators': [10, 25, 50, 100, 150, 200],
              'max_features': ['sqrt', 'log2', None]}

GB = model_selection.GridSearchCV(model, param_grid, scoring=scoring, refit='Recall', cv=10, return_train_score=True)
GB = GB.fit(X_train, y_train)

GB_Train = GB.predict(X_train)
gmTrain = geometric_mean_score(y_train, GB_Train)

GB_Test = GB.predict(X_test)
gmTest = geometric_mean_score(y_test, GB_Test)

cl_probs = GB.predict_proba(X_test)
cl_probs = cl_probs[:, 1]
cl_auc = metrics.roc_auc_score(y_test, cl_probs)
lr_f1 = metrics.f1_score(y_test, GB_Test)

cm = confusion_matrix(y_test, GB_Test, labels=GB.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                              display_labels=GB.classes_)

kappa = cohen_kappa_score(y_test, GB_Test)

indicekappa = ['Pobre', 'Débil', 'Moderado', 'Bueno', 'Muy Bueno']
verindicekappa = []
if kappa < 0.2:
    verindicekappa = indicekappa[0]
elif kappa > 0.2 and kappa < 0.4:
```



```

verindicekappa = indicekappa[1]
elif kappa > 0.4 and kappa < 0.6:
    verindicekappa = indicekappa[2]
elif kappa > 0.6 and kappa < 0.8:
    verindicekappa = indicekappa[3]
else:
    verindicekappa = indicekappa[4]

model_probs = GB.predict_proba(X_test)
model_probs = model_probs[:, 1]

ns_probs = [0 for _ in range (len (y_test))]
ns_auc = metrics.roc_auc_score(y_test, ns_probs)*100.0

model_auc = metrics.roc_auc_score(y_test, model_probs)*100.0
ns_fpr, ns_tpr, _ = metrics.roc_curve(y_test, ns_probs)
model_fpr, model_tpr, _ = metrics.roc_curve(y_test, model_probs)

no_skill = len(y_test[y_test==1]) / len(y_test)
model_precision, model_recall, _ = metrics.precision_recall_curve(y_test, model_probs)
model_f1 = metrics.f1_score(y_test, GB_Test)*100.0
model_auc_PR = metrics.auc(model_recall, model_precision)

tiempo_RF_aprender = timer() - start
print ("Calculados los mejores hiperparámetros en %f s" % tiempo_RF_aprender )

```

Calculados los mejores hiperparámetros en 7.271899 s

RESULTADOS

MULTIPLE EVALUATION METRICS

Este fragmento del cuaderno forma parte del manual de *scikit learn*

```
In [14]: results = GB.cv_results_
```

```
In [15]: plt.figure(figsize=(13, 13))
plt.title("GridSearchCV evaluating using multiple scorers simultaneously",
         fontsize=16)

plt.xlabel("n_estimators")
plt.ylabel("Score")

ax = plt.gca()
ax.set_xlim(0, 250)
ax.set_ylim(0.5, 1.10)

# Get the regular numpy array from the MaskedArray
X_axis = np.array(results['param_n_estimators'].data, dtype=float)

for scorer, color in zip(sorted(scoring), ['g', 'k', 'b', 'r', 'y']):
    for sample, style in (('train', '--'), ('test', '-')):
        sample_score_mean = results['mean_%s_%s' % (sample, scorer)]
        sample_score_std = results['std_%s_%s' % (sample, scorer)]
        ax.fill_between(X_axis, sample_score_mean - sample_score_std,
                       sample_score_mean + sample_score_std,
                       alpha=0 if sample == 'test' else 0, color=color)
        ax.plot(X_axis, sample_score_mean, style, color=color,
                alpha=0.25 if sample == 'test' else 0.7,
                label="%s (%s)" % (scorer, sample))

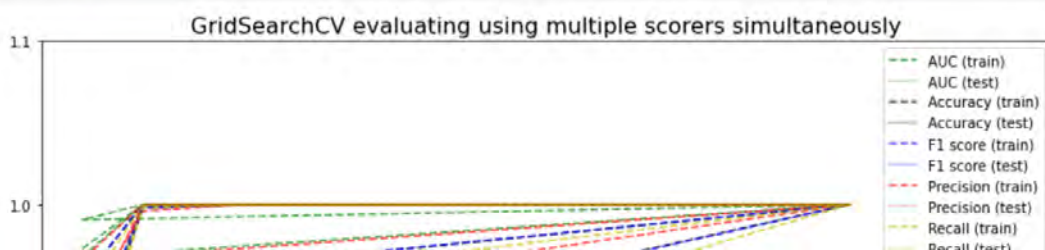
    best_index = np.nonzero(results['rank_test_%s' % scorer] == 1)[0][0]
    best_score = results['mean_test_%s' % scorer][best_index]

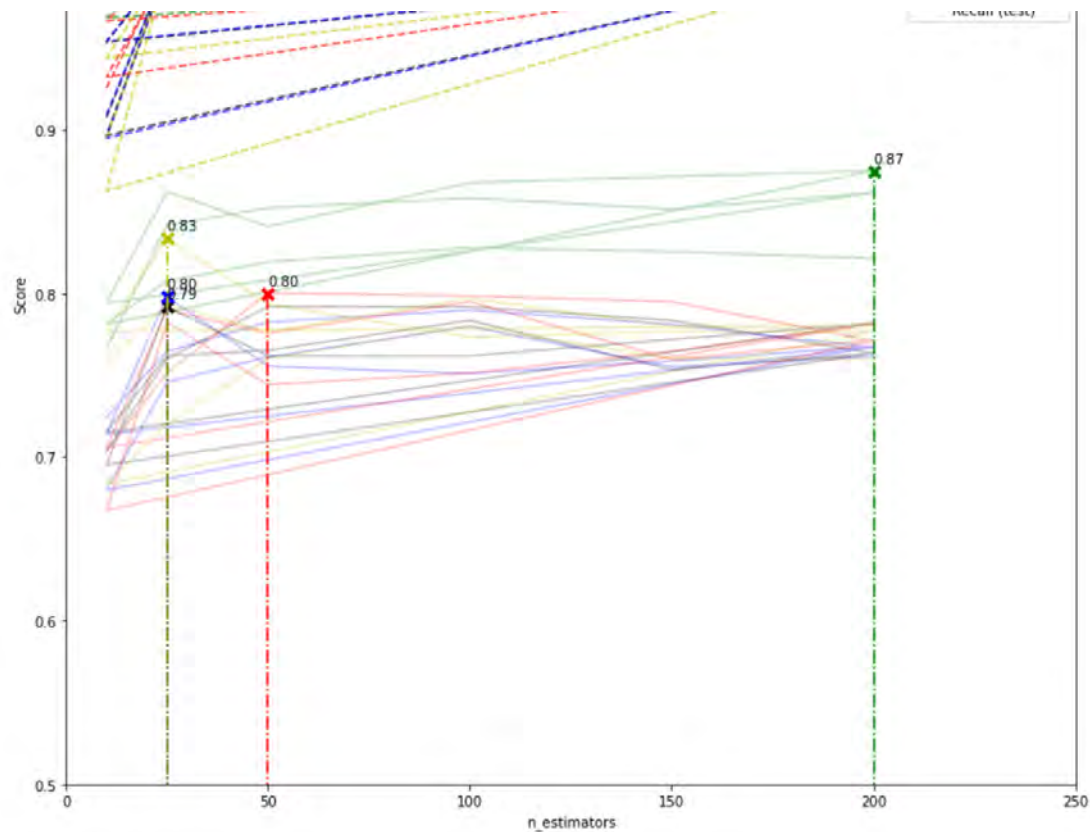
    # Plot a dotted vertical line at the best score for that scorer marked by x
    ax.plot([X_axis[best_index], ] * 2, [0, best_score],
            linestyle='-.', color=color, marker='x', markeredgewidth=3, ms=8)

    # Annotate the best score for that scorer
    ax.annotate("%0.2f" % best_score,
                (X_axis[best_index], best_score + 0.005))

plt.legend(loc="best")
plt.grid(False)
plt.show()

```

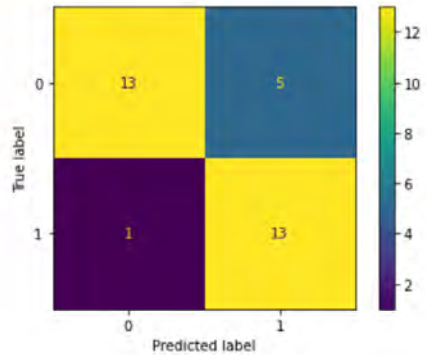




MATRIZ DE CONFUSIÓN

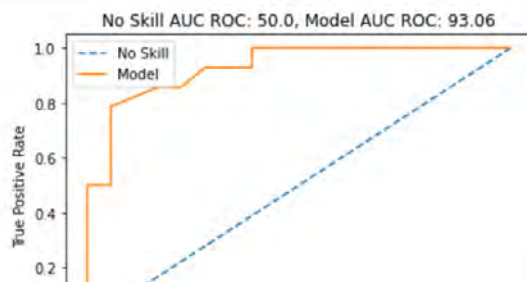
In [211]_ `disp.plot()`

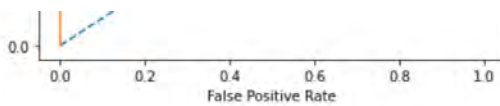
Out[211]_ `<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f548bcc4a8>`



ÁREA BAJO LA CURVA ROC

```
In [212]_ plt.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
plt.plot(model_fpr, model_tpr, label='Model')
# Etiquetas de los ejes
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Leyenda
plt.legend()
# Título
plt.title('No Skill AUC ROC: {}, Model AUC ROC: {}'.format(round(ns_auc,2), round(model_auc,2)))
# Mostramos la figura
plt.show()
```

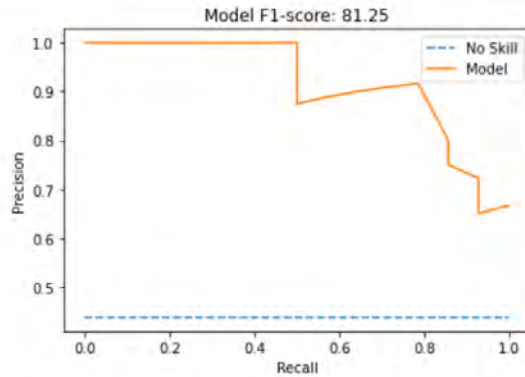




AREA BAJO LA CURVA Precision - Recall (PR)

```
In [213]: # Mostramos visualmente La curva PR

plt.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill')
plt.plot(model_recall, model_precision, label='Model')
# Etiquetas de los ejes
plt.xlabel('Recall')
plt.ylabel('Precision')
# Leyenda
plt.legend()
# Título
plt.title('Model F1-score: {}'.format(round(model_f1,2)))
# Mostramos la figura
plt.show()
```



Resultados en formato de texto

```
In [16]: print('Archivo: {} \nMetodo: {} \nMejores Hiperparámetros: {} \nMejor puntuación: {} \nRendimiento: (GM) {} \nROC AUC: {} \nModel AUC PR: {} \n
          GB.best_params_,
          round(GB.best_score,2),
          round(gmTest*100.0, 2),
          round(c1_auc*100.0, 2),
          round(model_auc_PR*100,2),
          round(lr_f1*100.0, 2),
          round(kappa,2),
          verindicekappa))
```

```
Archivo: BALANCED_G15.csv
Metodo: GradientBoostingClassifier(learning_rate=1, max_depth=1, random_state=0)
Mejores Hiperparámetros: {'max_features': 'log2', 'n_estimators': 25}
Mejor puntuación: 0.83
Rendimiento: (GM) 88.8
ROC AUC: 90.1
Model AUC PR: 92.14
F1 score: 87.8
Índice de Kappa: 0.78
Clasificador: Bueno
```

Fragmento de código recopilado del manual de la librería Fastai, a partir del cual se crean dos funciones para analizar la importancia de cada variable continua del clasificador.

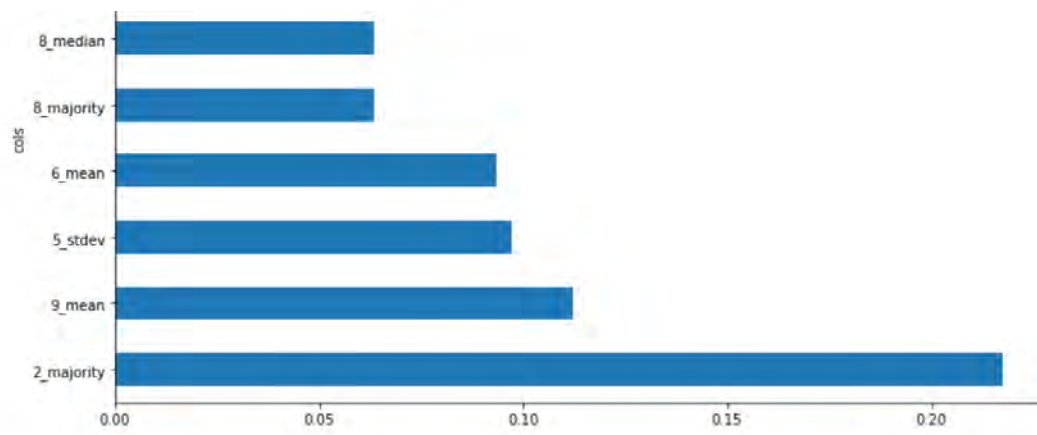
```
In [17]: def importancia (m,df):
          return pd.DataFrame({'cols':df.columns, 'imp': m.best_estimator_.feature_importances_}).sort_values('imp', ascending = False)

          def plot_importancia (imp):
              return imp.plot('cols', 'imp', 'barh', figsize = (12,7), legend=False)
```

```
In [19]: z = (variables.loc[:, '1_median': '11_majority'])
          imp = importancia (GB,z)
          to_feet = imp[:20]
          to_keep = imp[imp.imp > 0.05]
          plot_importancia(to_keep)
```

Out[19]: <AxesSubplot:ylabel='cols'>





Exportar el modelo

```
In [20]: import joblib
         joblib.dump(GB, '/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELLOS/modelo_G'+str(f)+'.pk1')
```

```
Out[20]: ['/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELLOS/modelo_G15.pk1']
```

NOTEBOOK 7

Este cuaderno está diseñado para buscar la mejor configuración de un clasificador basado en una Regresión Logística.

LIBRERÍAS

```
In [1]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from pandas import read_csv
from timeit import default_timer as timer
import warnings

warnings.filterwarnings('ignore')
```

DATASET

Leer la ubicación de los datos

```
In [2]: data_dir = os.path.dirname('/home/alex/TFM/YACIMIENTOS/CSV/ventanas_balanceadas/')
lista = os.listdir(data_dir)
```

Seleccionar el número de la ventana de observación

```
In [3]: f = 15
```

Eligir el archivo para calcular los mejor configuración

```
In [4]: cont = 0
for i in lista:
    if i == 'BALANCED_G'+str(f)+'.csv':
        n = cont
    else:
        cont += 1

lista[n]
```

```
Out[4]: 'BALANCED_G15.csv'
```

Leer el archivo

```
In [5]: dataset = pd.read_csv(data_dir+'/'+str(lista[n]), sep=',', decimal='.', na_values='')
dataset.shape
```

```
Out[5]: (152, 55)
```

Convertir los valores de NaN a valores de media

```
In [6]: dataset = dataset.fillna(dataset.mean())
```

```
In [7]: dataset.columns
```

```
Out[7]: Index(['fid', 'Group', 'AREA', 'CLASE', 'NOMBRE', 'COD_YACI', 'REDACTOR_FICHA',
'MUNICIPIO', '1_median', '1_stdev', '1_majority', '2_mean', '2_median',
'2_stdev', '2_majority', '3_mean', '3_median', '3_stdev', '3_majority',
'4_mean', '4_median', '4_stdev', '4_majority', '5_mean', '5_median',
'5_stdev', '5_majority', '1_mean', 'TIPO', 'TIPOLOGIA', 'CRONOLOGIA',
'6_mean', '6_median', '6_stdev', '6_majority', '7_mean', '7_median',
'7_stdev', '7_majority', '8_mean', '8_median', '8_stdev', '8_majority',
'9_mean', '9_median', '9_stdev', '9_majority', '10_mean', '10_median',
'10_stdev', '10_majority', '11_mean', '11_median', '11_stdev',
'11_majority'],
dtype='object')
```

Leer las variables continuas y categóricas implicadas en la clasificación

```
In [8]: atributos = []
for var in list(dataset.columns):
    if var[0].isdigit():
        atributos.append(var)

atributos.append('CLASE')
print(atributos)
```

```
11_stdev', '11_majority', '12_mean', '12_median', '12_stdev', '12_majority', '13_mean', '13_median', '13_stdev', '13_majority', '14_mean', '14_median', '14_stdev', '14_majority', '15_mean', '15_median', '15_stdev', '15_majority', '16_mean', '16_median', '16_stdev', '16_majority', '17_mean', '17_median', '17_stdev', '17_majority', '18_mean', '18_median', '18_stdev', '18_majority', '19_mean', '19_median', '19_stdev', '19_majority', '20_mean', '20_median', '20_stdev', '20_majority', '21_mean', '21_median', '21_stdev', '21_majority', '22_mean', '22_median', '22_stdev', '22_majority', '23_mean', '23_median', '23_stdev', '23_majority', '24_mean', '24_median', '24_stdev', '24_majority', '25_mean', '25_median', '25_stdev', '25_majority', '26_mean', '26_median', '26_stdev', '26_majority', '27_mean', '27_median', '27_stdev', '27_majority', '28_mean', '28_median', '28_stdev', '28_majority', '29_mean', '29_median', '29_stdev', '29_majority', '30_mean', '30_median', '30_stdev', '30_majority', '31_mean', '31_median', '31_stdev', '31_majority', '32_mean', '32_median', '32_stdev', '32_majority', '33_mean', '33_median', '33_stdev', '33_majority', '34_mean', '34_median', '34_stdev', '34_majority', '35_mean', '35_median', '35_stdev', '35_majority', '36_mean', '36_median', '36_stdev', '36_majority', '37_mean', '37_median', '37_stdev', '37_majority', '38_mean', '38_median', '38_stdev', '38_majority', '39_mean', '39_median', '39_stdev', '39_majority', '40_mean', '40_median', '40_stdev', '40_majority', '41_mean', '41_median', '41_stdev', '41_majority', '42_mean', '42_median', '42_stdev', '42_majority', '43_mean', '43_median', '43_stdev', '43_majority', '44_mean', '44_median', '44_stdev', '44_majority', '45_mean', '45_median', '45_stdev', '45_majority', '46_mean', '46_median', '46_stdev', '46_majority', '47_mean', '47_median', '47_stdev', '47_majority', '48_mean', '48_median', '48_stdev', '48_majority', '49_mean', '49_median', '49_stdev', '49_majority', '50_mean', '50_median', '50_stdev', '50_majority', '51_mean', '51_median', '51_stdev', '51_majority', '52_mean', '52_median', '52_stdev', '52_majority', '53_mean', '53_median', '53_stdev', '53_majority', '54_mean', '54_median', '54_stdev', '54_majority', '55_mean', '55_median', '55_stdev', '55_majority', '56_mean', '56_median', '56_stdev', '56_majority', '57_mean', '57_median', '57_stdev', '57_majority', '58_mean', '58_median', '58_stdev', '58_majority', '59_mean', '59_median', '59_stdev', '59_majority', '60_mean', '60_median', '60_stdev', '60_majority', '61_mean', '61_median', '61_stdev', '61_majority', '62_mean', '62_median', '62_stdev', '62_majority', '63_mean', '63_median', '63_stdev', '63_majority', '64_mean', '64_median', '64_stdev', '64_majority', '65_mean', '65_median', '65_stdev', '65_majority', '66_mean', '66_median', '66_stdev', '66_majority', '67_mean', '67_median', '67_stdev', '67_majority', '68_mean', '68_median', '68_stdev', '68_majority', '69_mean', '69_median', '69_stdev', '69_majority', '70_mean', '70_median', '70_stdev', '70_majority', '71_mean', '71_median', '71_stdev', '71_majority', '72_mean', '72_median', '72_stdev', '72_majority', '73_mean', '73_median', '73_stdev', '73_majority', '74_mean', '74_median', '74_stdev', '74_majority', '75_mean', '75_median', '75_stdev', '75_majority', '76_mean', '76_median', '76_stdev', '76_majority', '77_mean', '77_median', '77_stdev', '77_majority', '78_mean', '78_median', '78_stdev', '78_majority', '79_mean', '79_median', '79_stdev', '79_majority', '80_mean', '80_median', '80_stdev', '80_majority', '81_mean', '81_median', '81_stdev', '81_majority', '82_mean', '82_median', '82_stdev', '82_majority', '83_mean', '83_median', '83_stdev', '83_majority', '84_mean', '84_median', '84_stdev', '84_majority', '85_mean', '85_median', '85_stdev', '85_majority', '86_mean', '86_median', '86_stdev', '86_majority', '87_mean', '87_median', '87_stdev', '87_majority', '88_mean', '88_median', '88_stdev', '88_majority', '89_mean', '89_median', '89_stdev', '89_majority', '90_mean', '90_median', '90_stdev', '90_majority', '91_mean', '91_median', '91_stdev', '91_majority', '92_mean', '92_median', '92_stdev', '92_majority', '93_mean', '93_median', '93_stdev', '93_majority', '94_mean', '94_median', '94_stdev', '94_majority', '95_mean', '95_median', '95_stdev', '95_majority', '96_mean', '96_median', '96_stdev', '96_majority', '97_mean', '97_median', '97_stdev', '97_majority', '98_mean', '98_median', '98_stdev', '98_majority', '99_mean', '99_median', '99_stdev', '99_majority', '100_mean', '100_median', '100_stdev', '100_majority', '101_mean', '101_median', '101_stdev', '101_majority', '102_mean', '102_median', '102_stdev', '102_majority', '103_mean', '103_median', '103_stdev', '103_majority', '104_mean', '104_median', '104_stdev', '104_majority', '105_mean', '105_median', '105_stdev', '105_majority', '106_mean', '106_median', '106_stdev', '106_majority', '107_mean', '107_median', '107_stdev', '107_majority', '108_mean', '108_median', '108_stdev', '108_majority', '109_mean', '109_median', '109_stdev', '109_majority', '110_mean', '110_median', '110_stdev', '110_majority', '111_mean', '111_median', '111_stdev', '111_majority', '112_mean', '112_median', '112_stdev', '112_majority', '113_mean', '113_median', '113_stdev', '113_majority', '114_mean', '114_median', '114_stdev', '114_majority', '115_mean', '115_median', '115_stdev', '115_majority', '116_mean', '116_median', '116_stdev', '116_majority', '117_mean', '117_median', '117_stdev', '117_majority', '118_mean', '118_median', '118_stdev', '118_majority', '119_mean', '119_median', '119_stdev', '119_majority', '120_mean', '120_median', '120_stdev', '120_majority', '121_mean', '121_median', '121_stdev', '121_majority', '122_mean', '122_median', '122_stdev', '122_majority', '123_mean', '123_median', '123_stdev', '123_majority', '124_mean', '124_median', '124_stdev', '124_majority', '125_mean', '125_median', '125_stdev', '125_majority', '126_mean', '126_median', '126_stdev', '126_majority', '127_mean', '127_median', '127_stdev', '127_majority', '128_mean', '128_median', '128_stdev', '128_majority', '129_mean', '129_median', '129_stdev', '129_majority', '130_mean', '130_median', '130_stdev', '130_majority', '131_mean', '131_median', '131_stdev', '131_majority', '132_mean', '132_median', '132_stdev', '132_majority', '133_mean', '133_median', '133_stdev', '133_majority', '134_mean', '134_median', '134_stdev', '134_majority', '135_mean', '135_median', '135_stdev', '135_majority', '136_mean', '136_median', '136_stdev', '136_majority', '137_mean', '137_median', '137_stdev', '137_majority', '138_mean', '138_median', '138_stdev', '138_majority', '139_mean', '139_median', '139_stdev', '139_majority', '140_mean', '140_median', '140_stdev', '140_majority', '141_mean', '141_median', '141_stdev', '141_majority', '142_mean', '142_median', '142_stdev', '142_majority', '143_mean', '143_median', '143_stdev', '143_majority', '144_mean', '144_median', '144_stdev', '144_majority', '145_mean', '145_median', '145_stdev', '145_majority', '146_mean', '146_median', '146_stdev', '146_majority', '147_mean', '147_median', '147_stdev', '147_majority', '148_mean', '148_median', '148_stdev', '148_majority', '149_mean', '149_median', '149_stdev', '149_majority', '150_mean', '150_median', '150_stdev', '150_majority', '151_mean', '151_median', '151_stdev', '151_majority', '152_mean', '152_median', '152_stdev', '152_majority', 'CLASE'
```

```
[ '1_median', '1_stdev', '1_majority', '2_mean', '2_median', '2_stdev', '2_majority', '3_mean', '3_median', '3_stdev', '3_majority', '4_mean', '4_median', '4_stdev', '4_majority', '5_mean', '5_median', '5_stdev', '5_majority', '6_mean', '6_median', '6_stdev', '6_majority', '7_mean', '7_median', '7_stdev', '7_majority', '8_mean', '8_median', '8_stdev', '8_majority', '9_mean', '9_median', '9_stdev', '9_majority', '10_mean', '10_median', '10_stdev', '10_majority', '11_mean', '11_median', '11_stdev', '11_majority', 'CLASE']
```

```
In [9]: variables = dataset.loc[:,atributos]
variables.shape
```

```
Out[9]: (152, 45)
```

Crear los atributos X, Y

```
In [10]: X = np.array(variables.loc[:, '1_median': '11_majority'])
Y = np.array(variables.loc[:, 'CLASE'])
```

```
In [11]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

ENTRENAMIENTO

Separar el dataset en dos grupos. Un grupo para el entrenamiento y el otro para la validación. Así como normalizar los datos de entrenamiento y ajustar los datos de validación a la dimensión de los datos de entrenamiento.

```
In [12]: from sklearn import model_selection
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from imblearn.metrics import geometric_mean_score
from sklearn.metrics import classification_report
from sklearn.metrics import cohen_kappa_score
from sklearn import metrics

np.random.seed(12)

X_train, X_test, y_train, y_test = model_selection.train_test_split(X, Y, train_size=0.7)
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

LOGISTIC REGRESION

Buscar la mejor configuración con GridSearchCV

```
In [13]: from sklearn.linear_model import LogisticRegression

#Inicio contador tiempo
start = timer()
np.random.seed(12)
#constructor y parametros

scoring = {'AUC': 'roc_auc', 'F1 score': 'f1', 'Precision': 'precision', 'Recall': 'recall', 'B_Accuracy': 'balanced_accuracy'}

model = LogisticRegression(multi_class= 'ovr', n_jobs = -1, random_state = 48)

param_grid = {'penalty': ['l1', 'l2', 'elasticnet', 'none'], 'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'], 'max_iter': [100, 500]}

LR = model_selection.GridSearchCV(model, param_grid, scoring=scoring, refit= 'Recall' , cv=10, return_train_score=True)
LR = LR.fit(X_train, y_train)

LR_Train = LR.predict(X_train)
gmTrain = geometric_mean_score(y_train, LR_Train)

LR_Test = LR.predict(X_test)
gmTest = geometric_mean_score(y_test, LR_Test)

cl_probs = LR.predict_proba(X_test)
cl_probs = cl_probs[:, 1]
cl_auc = metrics.roc_auc_score(y_test, cl_probs)
lr_f1 = metrics.f1_score(y_test, LR_Test)

cm = confusion_matrix(y_test, LR_Test, labels=LR.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=LR.classes_)

kappa = cohen_kappa_score(y_test, LR_Test)

indicekappa = ['Pobre', 'Débil', 'Moderado', 'Bueno', 'Muy Bueno']
verindicekappa = []
if kappa < 0.2:
    verindicekappa = indicekappa[0]
elif kappa > 0.2 and kappa < 0.4:
    verindicekappa = indicekappa[1]
elif kappa > 0.4 and kappa < 0.6:
    verindicekappa = indicekappa[2]
```

```

elif kappa > 0.6 and kappa < 0.8:
    verindicekappa = indicekappa[3]
else:
    verindicekappa = indicekappa[4]

model_probs = LR.predict_proba(X_test)
model_probs = model_probs[:, 1]

ns_probs = [0 for _ in range(len(y_test))]
ns_auc = metrics.roc_auc_score(y_test, ns_probs)*100.0

model_auc = metrics.roc_auc_score(y_test, model_probs)*100.0
ns_fpr, ns_tpr, _ = metrics.roc_curve(y_test, ns_probs)
model_fpr, model_tpr, _ = metrics.roc_curve(y_test, model_probs)

no_skill = len(y_test[y_test==1]) / len(y_test)
model_precision, model_recall, _ = metrics.precision_recall_curve(y_test, model_probs)
model_f1 = metrics.f1_score(y_test, LR_Test)*100.0
model_auc_PR = metrics.auc(model_recall, model_precision)

tiempo_RF_aprender = timer() - start

print ("Calculados los mejores hiperparámetros en %f segundos" % tiempo_RF_aprender )

```

Calculados los mejores hiperparámetros en 20.006560 segundos

RESULTADOS

MULTIPLE EVALUATION METRICS

Este fragmento del cuaderno forma parte del manual de *scikit learn*

```
In [14]: results = LR.cv_results_
```

```
In [15]: plt.figure(figsize=(13, 13))
plt.title("GridSearchCV evaluating using multiple scorers simultaneously",
          fontsize=16)

plt.xlabel("max_iter")
plt.ylabel("Score")

ax = plt.gca()
ax.set_xlim(0, 250)
ax.set_ylim(0.5, 1.10)

# Get the regular numpy array from the MaskedArray
X_axis = np.array(results['param_max_iter'].data, dtype=float)

for scorer, color in zip(sorted(scoring), ['g', 'k', 'b', 'r', 'y']):
    for sample, style in (('train', '-.-'), ('test', '-')):
        sample_score_mean = results['mean_%s_%s' % (sample, scorer)]
        sample_score_std = results['std_%s_%s' % (sample, scorer)]
        ax.fill_between(X_axis, sample_score_mean - sample_score_std,
                        sample_score_mean + sample_score_std,
                        alpha=0 if sample == 'test' else 0, color=color)
        ax.plot(X_axis, sample_score_mean, style, color=color,
                alpha=0.25 if sample == 'test' else 0.7,
                label="%s (%s)" % (scorer, sample))

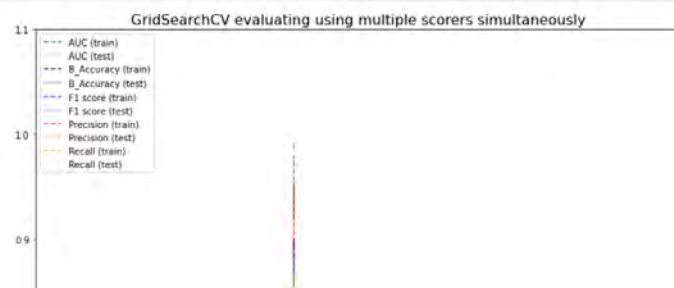
    best_index = np.nonzero(results['rank_test_%s' % scorer] == 1)[0][0]
    best_score = results['mean_test_%s' % scorer][best_index]

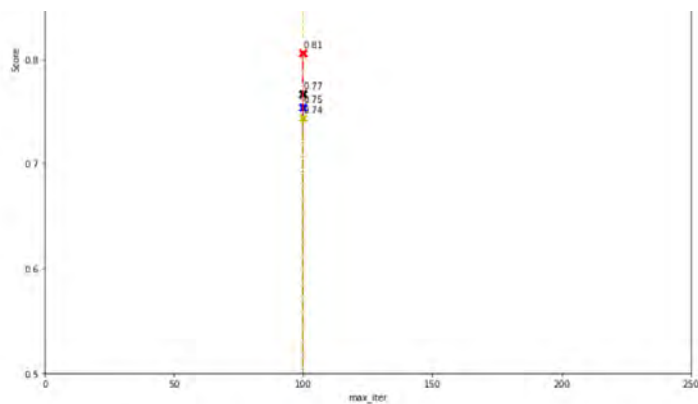
    # Plot a dotted vertical line at the best score for that scorer marked by x
    ax.plot([X_axis[best_index], ] * 2, [0, best_score],
            linestyle='-.', color=color, marker='x', markeredgewidth=3, ms=8)

    # Annotate the best score for that scorer
    ax.annotate("%0.2f" % best_score,
                (X_axis[best_index], best_score + 0.005))

plt.legend(loc="best")
plt.grid(False)
plt.show()

```

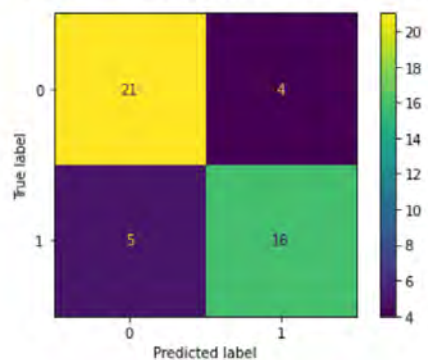




MATRIZ DE CONFUSIÓN

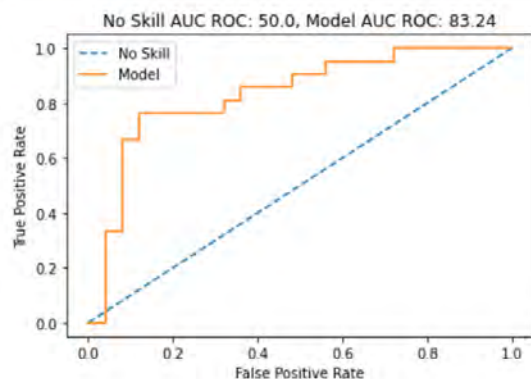
```
In [16]: disp.plot()
```

```
Out[16]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f78a01efa58>
```



ÁREA BAJO LA CURVA ROC

```
In [17]: plt.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
plt.plot(model_fpr, model_tpr, label='Model')
# Etiquetas de Los ejes
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Leyenda
plt.legend()
# Titulo
plt.title('No Skill AUC ROC: {}, Model AUC ROC: {}'.format(round(ns_auc,2), round(model_auc,2)))
# Mostramos la figura
plt.show()
```



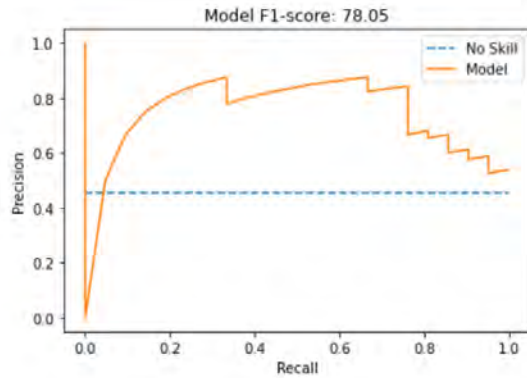
AREA BAJO LA CURVA Precision - Recall (PR)

```
In [18]: # Mostramos visualmente La curva PR

plt.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill')
plt.plot(model_recall, model_precision, label='Model')
# Etiquetas de Los ejes
plt.xlabel('Recall')
plt.ylabel('Precision')
# Leyenda
plt.legend()
# Titulo
plt.title('Model F1-score: {}'.format(round(model_f1,2)))
```



```
# Mostramos la figura
plt.show()
```



Resultados en formato de texto

```
In [19]: print('Archivo: {} \nMetodo: {} \nMejores Hiperparámetros: {} \nMejor puntuación: {} \nRendimiento: (GM) {} \nROC AUC: {} \nModel AUC PR: {} \nLR.best_params_,
round(LR.best_score_,2),
round(gmTest*100.0, 2),
round(c1_auc*100.0, 2),
round(model_auc_PR*100,2),
round(lr_f1*100.0, 2),
round(kappa,2),
verindicekappa))
```

```
Archivo: BALANCED_G15.csv
Metodo: LogisticRegression(multi_class='ovr', n_jobs=-1, random_state=48)
Mejores Hiperparámetros: {'max_iter': 100, 'penalty': 'l1', 'solver': 'saga'}
Mejor puntuación: 0.74
Rendimiento: (GM) 80.0
ROC AUC: 83.24
Model AUC PR: 73.39
F1 score: 78.05
Índice de Kappa: 0.6
Clasificador: Bueno
```

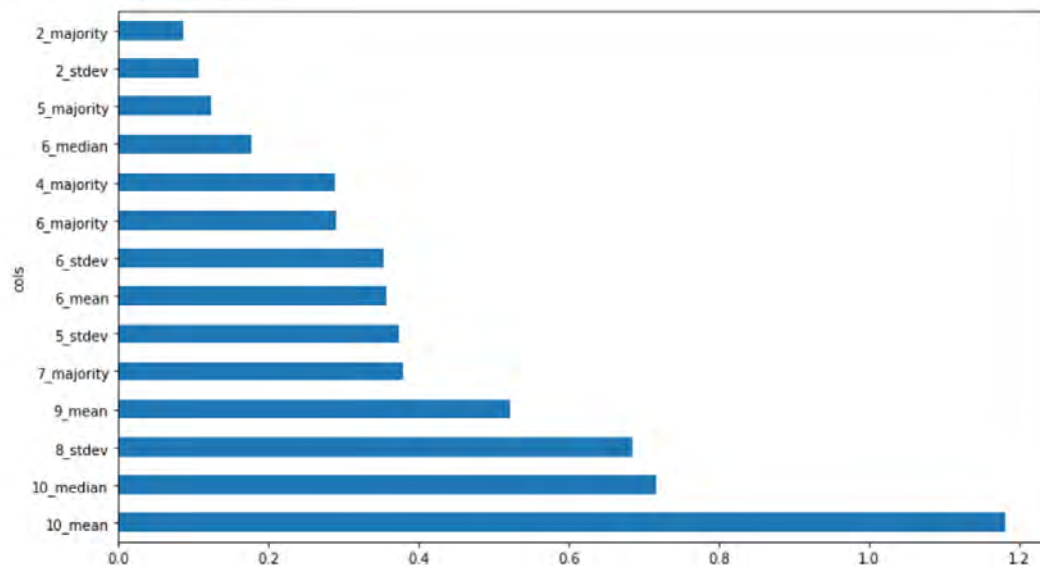
Fragmento de código recopilado del manual de la librería Fastai, a partir del cual se crean dos funciones para analizar la importancia de cada variable continua del clasificador.

```
In [20]: def importancia (m,df):
return pd.DataFrame({'cols':df.columns, 'imp': m.best_estimator_.coef_[0]}).sort_values('imp', ascending = False)

def plot_importancia (imp):
return imp.plot('cols', 'imp', 'barh', figsize = (12,7), legend=False)
```

```
In [21]: z = (variables.loc[:, '1_median': '11_majority'])
imp = importancia (LR,z)
to_feet = imp[:20]
to_keep = imp[imp.imp > 0.05]
plot_importancia(to_keep)
```

Out[21]: <AxesSubplot:ylabel='cols'>



Exportar el modelo

```
In [22]: import joblib
         joblib.dump(LR, '/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELOS/modelo_G'+str(f)+'.pk1')
```

```
Out[22]: ['/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELOS/modelo_G15.pk1']
```

NOTEBOOK 8

Este cuaderno esta diseñado para aplicar la mejor configuración de un clasificador sobre datos nunca vistos.

LIBRERÍAS

```
In [1]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from pandas import read_csv
from timeit import default_timer as timer
import warnings

warnings.filterwarnings('ignore')
```

DATASET

Leer la ubicación de los datos

```
In [2]: data_dir = os.path.dirname('/home/alex/TFM/INFERENCE/CSV/')
lista = os.listdir(data_dir)
```

Seleccionar el número de la ventana de observación

```
In [3]: f = 6
```

Eligir el archivo para calcular los mejor configuración

```
In [4]: cont = 0
for i in lista:
    if i == 'G'+str(f)+'.csv':
        n = cont
    else:
        cont += 1

lista[n]
```

```
Out[4]: '66.csv'
```

Leer el archivo

```
In [5]: dataset = pd.read_csv(data_dir+'/'+str(lista[n]),sep=',', decimal = '.',na_values= '')
dataset.shape
```

```
Out[5]: (11583, 50)
```

Convertir los valores de NaN a valores de media

```
In [6]: dataset = dataset.fillna(dataset.mean())
```

```
In [7]: dataset.columns
```

```
Out[7]: Index(['fid', 'id', 'left', 'top', 'right', 'bottom', '1_mean', '1_median',
'1_stdev', '1_majority', '2_mean', '2_median', '2_stdev', '2_majority',
'3_mean', '3_median', '3_stdev', '3_majority', '4_mean', '4_median',
'4_stdev', '4_majority', '5_mean', '5_median', '5_stdev', '5_majority',
'6_mean', '6_median', '6_stdev', '6_majority', '7_mean', '7_median',
'7_stdev', '7_majority', '8_mean', '8_median', '8_stdev', '8_majority',
'9_mean', '9_median', '9_stdev', '9_majority', '10_mean', '10_median',
'10_stdev', '10_majority', '11_mean', '11_median', '11_stdev',
'11_majority'],
dtype='object')
```

Leer las variables continuas y categóricas implicadas en la clasificación

```
In [8]: atributos = []
for var in list(dataset.columns):
    if var[0].isdigit():
        atributos.append(var)
atributos.append('fid')
print(atributos)
```

```
['1_mean', '1_median', '1_stdev', '1_majority', '2_mean', '2_median', '2_stdev', '2_majority', '3_mean', '3_median', '3_stdev', '3_majo
rity', '4_mean', '4_median', '4_stdev', '4_majority', '5_mean', '5_median', '5_stdev', '5_majority', '6_mean', '6_median', '6_stdev',
'6_majority', '7_mean', '7_median', '7_stdev', '7_majority', '8_mean', '8_median', '8_stdev', '8_majority', '9_mean', '9_median', '9_st
dev', '9_majority', '10_mean', '10_median', '10_stdev', '10_majority', '11_mean', '11_median', '11_stdev', '11_majority']
```

```
0_majority', '10_mean', '10_median', '10_stdev', '10_majority', '11_mean', '11_median', '11_stdev', '11_majority', 'fid']
```

```
In [9]: variables = dataset.loc[:,atributos]
variables.shape
```

```
Out[9]: (11583, 45)
```

Crear los atributos X, Y

```
In [10]: X_ = np.array(variables.loc[:, '1_mean': '11_majority'])
Y_ = np.array(variables.loc[:, 'fid'])
```

En este paso se normalizan los valores en un rango de -3 a 3 para que coincidan con el mismo procedimiento de normalización de datos del entrenamiento.

```
In [11]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_ = scaler.fit_transform(X_)
```

Importar el modelo entrenado con la misma dimensión de ventana

```
In [12]: import joblib
m = joblib.load('/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELLOS/modelo_G'+str(f)+''.pk1')
```

Predecir probabilidades y seleccionar sólo los valores de probabilidad de pertenecer a la clase positiva.

```
In [13]: preds = m.predict_proba(X_)
preds = preds[:, 1]
```

Recorrer los valores de Y donde se localizan los indicadores de posición de cada una de las ventanas usadas para inferir.

```
In [14]: for i in range(0, len(Y_)):
    DF = pd.DataFrame({'fid': Y_,
                      'Probabilidad': preds,
                      'index': list(range(0, len(Y_)))})
```

```
In [15]: DF
```

```
Out[15]:
```

	fid	Probabilidad
0	1	0.780000
1	2	0.733333
2	3	0.480000
3	4	0.460000
4	5	0.480000
...
11578	11579	0.480000
11579	11580	0.386667
11580	11581	0.600000
11581	11582	0.660000
11582	11583	0.700000

11583 rows × 2 columns

GUARDAR

```
In [16]: DF.to_csv('/home/alex/TFM/INFERENCE/CSV/predict_G'+str(f)+''.csv')
```

NOTEBOOK 9

En este cuaderno se realiza la comparación entre los mejores modelos entrenados y todo el conjunto de datos separados por ventanas de observación. El objetivo es valorar si existe algún tipo de aprendizaje.

LIBRERÍAS

```
In [1]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from pandas import read_csv
from timeit import default_timer as timer

from sklearn.preprocessing import StandardScaler
from sklearn import model_selection
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from imblearn.metrics import geometric_mean_score
from sklearn.metrics import classification_report
from sklearn.metrics import cohen_kappa_score
from sklearn import metrics

from sklearn.metrics import plot_roc_curve
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
from sklearn.metrics import average_precision_score

from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn import neighbors
from sklearn.linear_model import LogisticRegression
from sklearn import tree
import joblib

import warnings

warnings.filterwarnings('ignore')
```

DATASET

Declarar la localización de los datos

```
In [2]: data_dir = os.path.dirname('/home/alex/TFM/YACIMIENTOS/CSV/ventanas_balanceadas/')
lista = os.listdir(data_dir)
```

```
In [3]: model_dir = os.path.dirname('/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELOS/')
model = os.listdir(model_dir)
```

```
In [4]: a = 40
```

```
In [5]: cont = 0
for i in model:
    if i == 'modelo_G'+str(a)+'.pkl':
        n = cont
    else:
        cont += 1

model[n]
```

```
Out[5]: 'modelo_G40.pkl'
```

Lista para almacenar resultados

```
In [6]: resultados = []
```

```
In [7]: for j in range(0, len(lista)):
dataset = pd.read_csv(data_dir+'/'+str(lista[j]), sep=',', decimal='.', na_values='')
dataset = dataset.fillna(dataset.mean())
atributos = []
for var in list(dataset.columns):
    if var[0].isdigit():
        atributos.append(var)
atributos.append('CLASE')
variables = dataset.loc[:, atributos]

X = np.array(variables.loc[:, '1_median': '11_majority'])
```

```

Y_ = np.array(variables.loc[:, 'CLASE'])

scaler = StandardScaler()
X_ = scaler.fit_transform(X)
m = joblib.load('/home/alex/TFM/DOCUMENTO/RESULTADOS/MODELOS/modelo_G'+str(a)+'.pkl')
start = timer()
m_Test = m.predict(X_)
gmTest = geometric_mean_score(Y_, m_Test)
cl_probs = m.predict_proba(X_)
cl_probs = cl_probs[:, 1]
cl_auc = metrics.roc_auc_score(Y_, cl_probs)
lr_f1 = metrics.f1_score(Y_, m_Test)
cm = confusion_matrix(Y_, m_Test, labels=m.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=m.classes_)
kappa = cohen_kappa_score(Y_, m_Test)
indicekappa = ['Pobre', 'Débil', 'Moderado', 'Bueno', 'Muy Bueno']
verindicekappa = []
if kappa < 0.2:
    verindicekappa = indicekappa[0]
elif kappa >= 0.2001 and kappa <= 0.4:
    verindicekappa = indicekappa[1]
elif kappa >= 0.4001 and kappa <= 0.6:
    verindicekappa = indicekappa[2]
elif kappa >= 0.6001 and kappa <= 0.8:
    verindicekappa = indicekappa[3]
else:
    verindicekappa = indicekappa[4]
model_probs = m.predict_proba(X_)
model_probs = model_probs[:, 1]
ns_probs = [0 for _ in range(len(Y_))]
ns_auc = metrics.roc_auc_score(Y_, ns_probs)*100.0
model_auc = metrics.roc_auc_score(Y_, model_probs)*100.0
ns_fpr, ns_tpr, _ = metrics.roc_curve(Y_, ns_probs)
model_fpr, model_tpr, _ = metrics.roc_curve(Y_, model_probs)
no_skill = len(Y_[Y_==1]) / len(Y_)
model_precision, model_recall, _ = metrics.precision_recall_curve(Y_, model_probs)
model_f1 = metrics.f1_score(Y_, m_Test)*100.0
model_auc_PR = metrics.auc(model_recall, model_precision)
tiempo = timer() - start
#print ("Hecho en " % tiempo)
resultados.append((model[n],
                   lista[j],
                   round(gmTest*100.0, 2),
                   round(cl_auc*100.0, 2),
                   round(model_auc_PR*100.0, 2),
                   round(lr_f1*100.0, 2),
                   round(kappa, 2),
                   verindicekappa))

```

Declarar tantas listas como posiciones tenga la lista resultados donde se han almacenado los valores resultantes del entrenamiento.

```
In [8]: Modelo, Ventana, GM, ROC, AUCPR, F1, K, tipo = [], [], [], [], [], [], [], []
```

Asignar valores

```
In [9]: for i in range(0, len(resultados)):
Modelo.append(resultados[i][0])
Ventana.append(resultados[i][1])
GM.append(resultados[i][2])
ROC.append(resultados[i][3])
AUCPR.append(resultados[i][4])
F1.append(resultados[i][5])
K.append(resultados[i][6])
tipo.append(resultados[i][7])

```

Crear el DataFrame

```
In [10]: for i in range(0, len(resultados)):
DF = pd.DataFrame({'Modelo': Modelo,
                  'Ventana': Ventana,
                  'Media Geométrica': GM,
                  'ROC AUC': ROC,
                  'AUC PR': AUCPR,
                  'F1 score': F1,
                  'Índice Kappa': K,
                  'Tipo de clasificador': tipo},
                 index=list(range(0, len(resultados))))

```

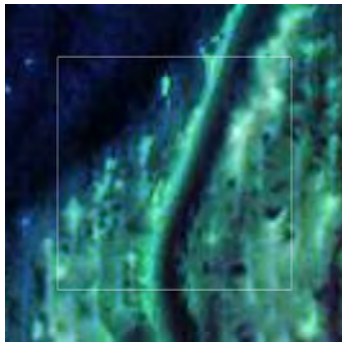
Guardar como .csv

```
In [11]: DF.to_csv('/home/alex/TFM/DOCUMENTO/RESULTADOS/COMPARATIVA/modelo_G'+str(a)+'_comparado.csv')
```

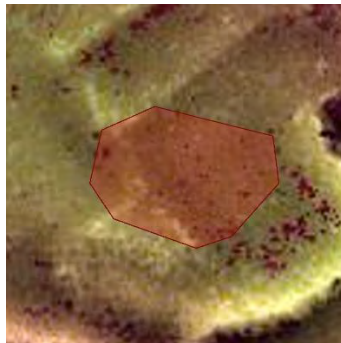
Anexo II. Fichas de resultados del entrenamiento de los modelos

Ventana	G1	Tipo de clasificador	Bueno		
Método	GradientBoostingClassifier(learning_rate=1, max_depth=1, random_state=0)				
Mejor configuración	{ 'max_features': 'sqrt', 'n_estimators': 50 }				
Rango (m ²)	1047 - 1586	Rendimiento (GM)	81.65	ROC AUC	88.89
Model AUC PR	90.04	F1 score	80.0	Índice de Kappa	0.69
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

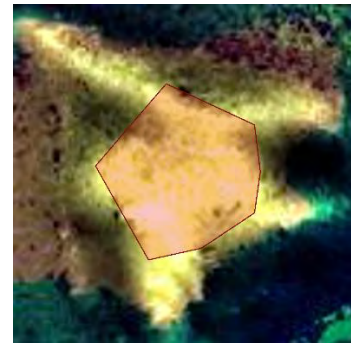
33



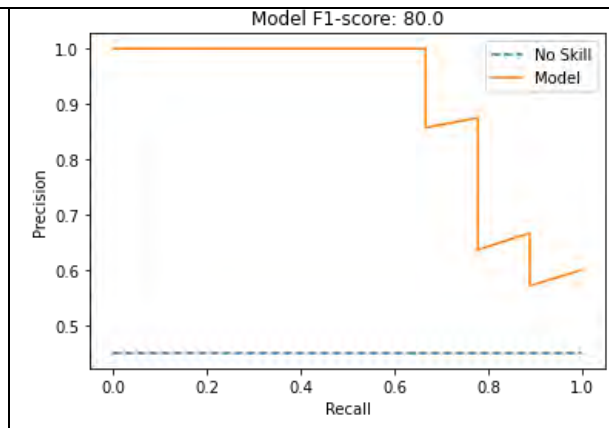
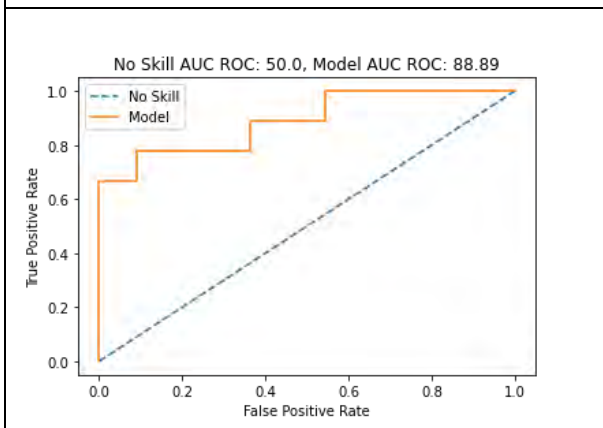
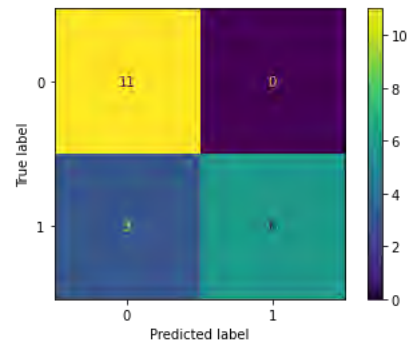
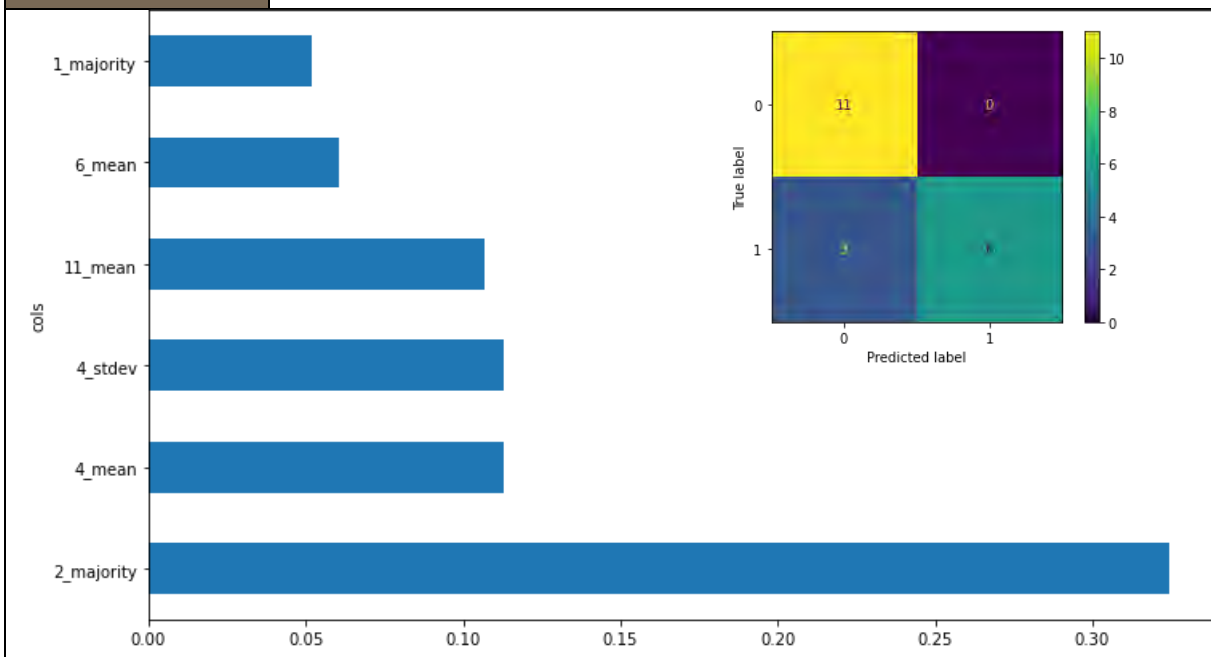
8



25



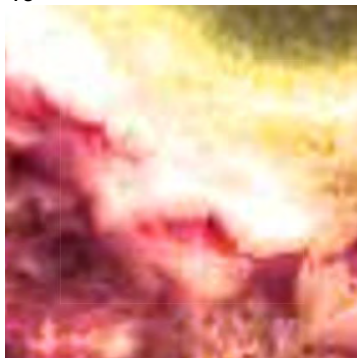
Leyenda imágenes de ejemplo		Ventana clasificador		Yacimiento IAN
-----------------------------	--	----------------------	--	----------------



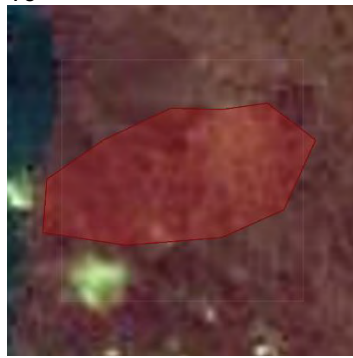
Ventana	G2	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 50 }				
Rango (m²)	547 - 569	Rendimiento (GM)	79.33	ROC AUC	88.11
Model AUC PR	83.21	F1 score	78.26	Índice de Kappa	0.58

Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)
-----------------------	----------------------------	-------------------------

40



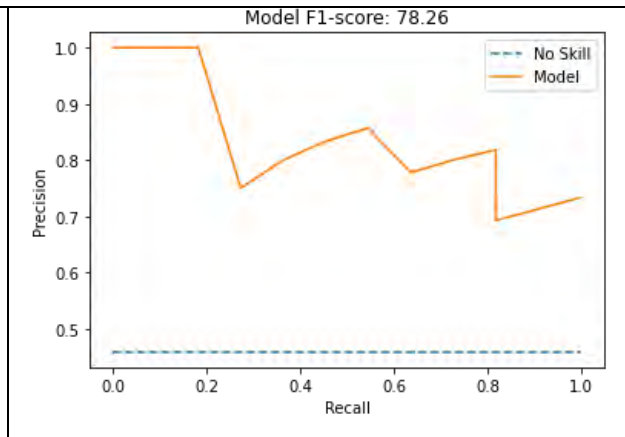
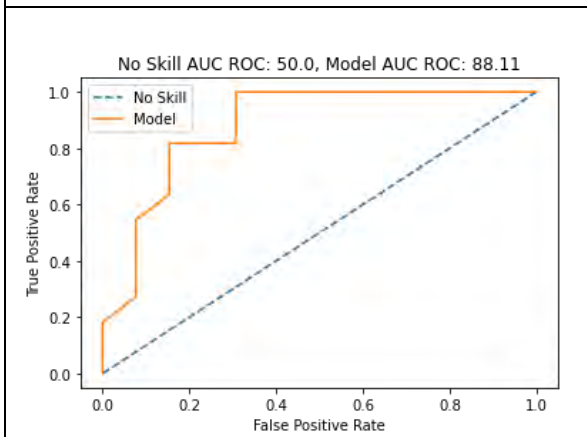
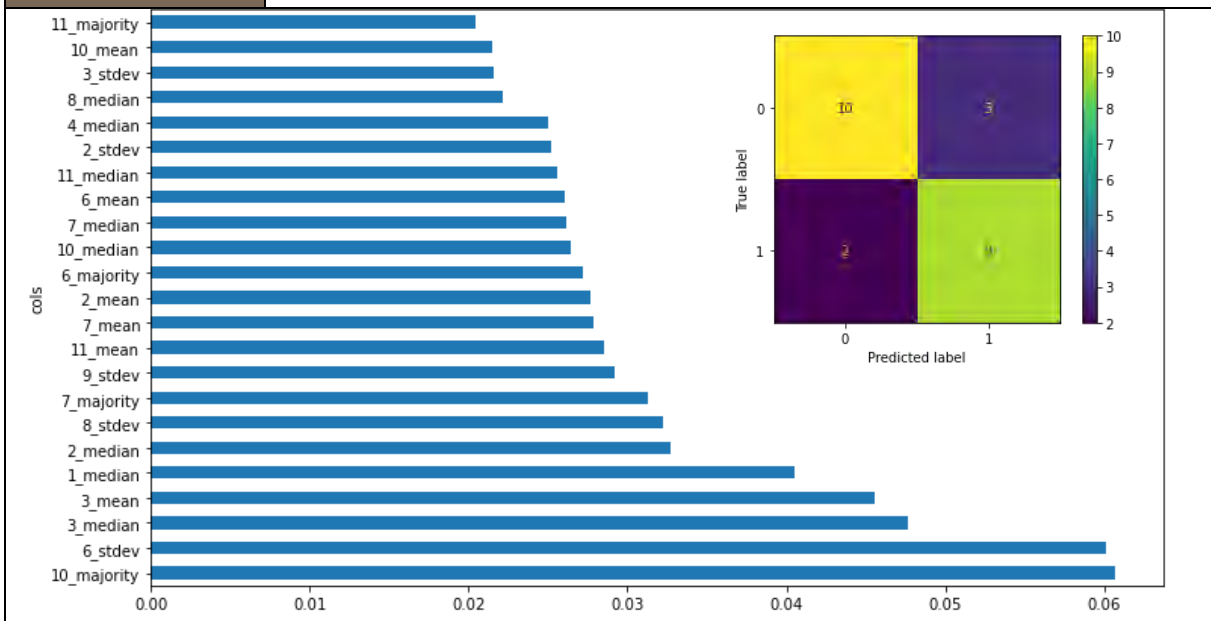
10



30

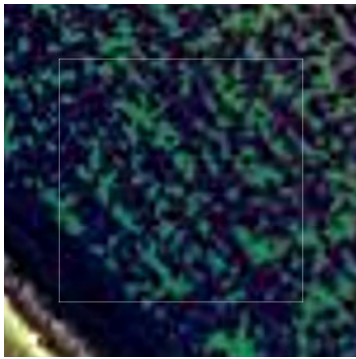


Leyenda imágenes de ejemplo	Ventana clasificador	Yacimiento IAN
------------------------------------	----------------------	----------------



Ventana	G3	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 200 }				
Rango (m ²)	590 - 667	Rendimiento (GM)	79.21	ROC AUC	84.64
Model AUC PR	86.15	F1 score	79.01	Índice de Kappa	0.57
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

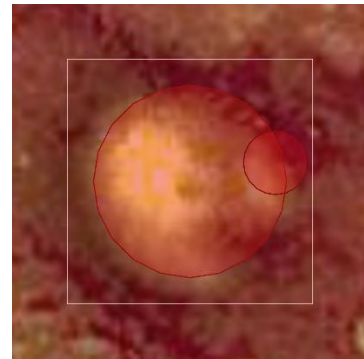
131



33



98



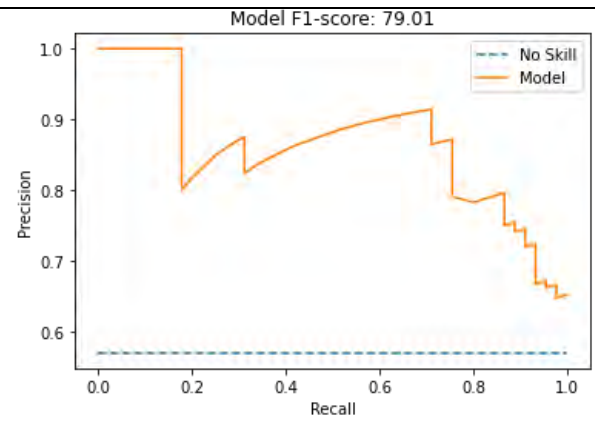
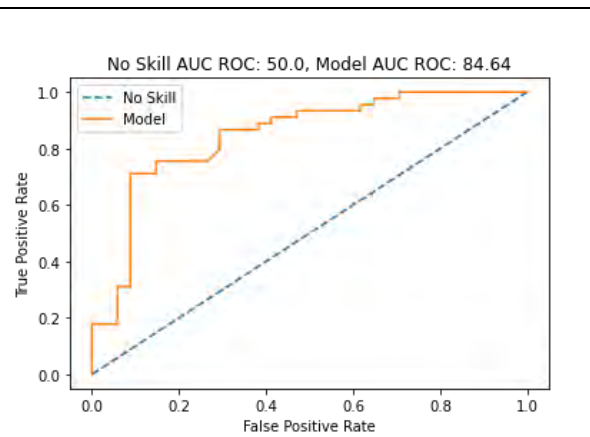
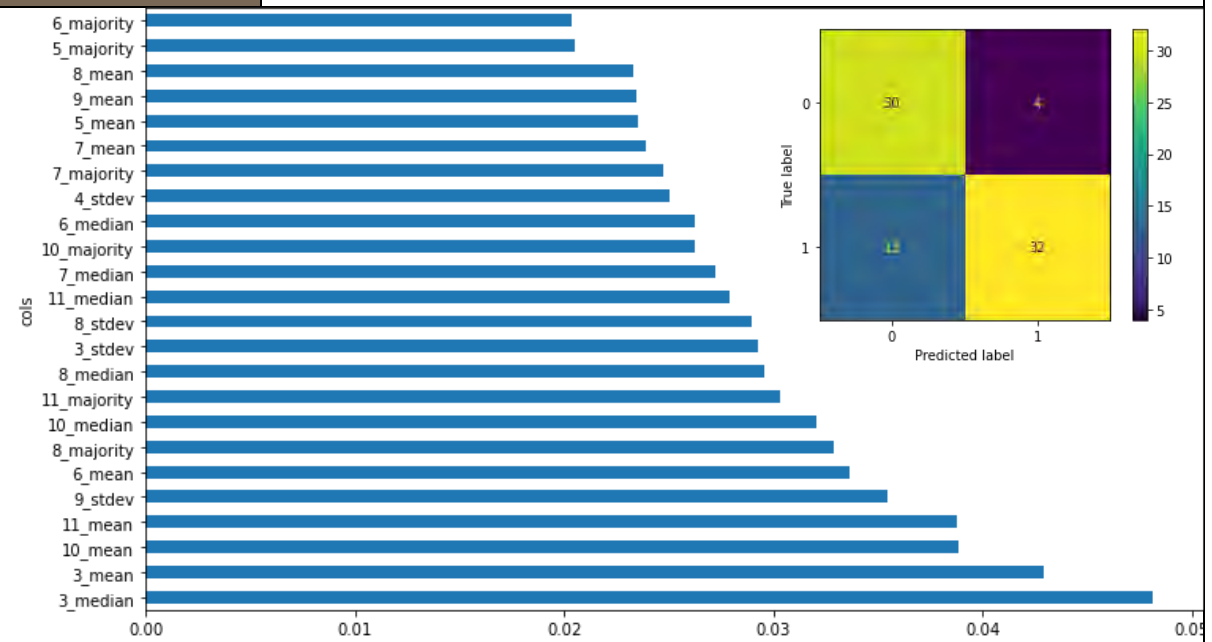
Leyenda imágenes de ejemplo



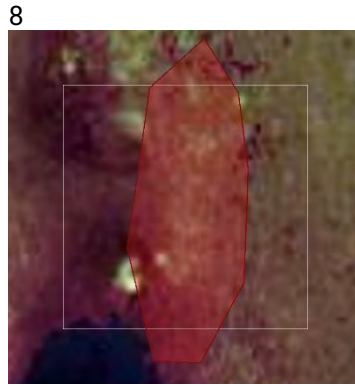
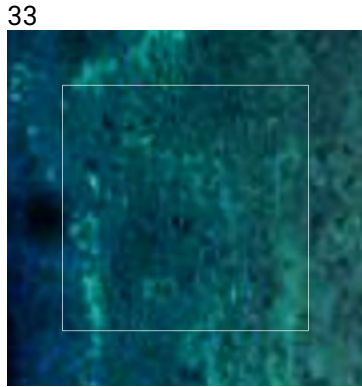
Ventana clasificador



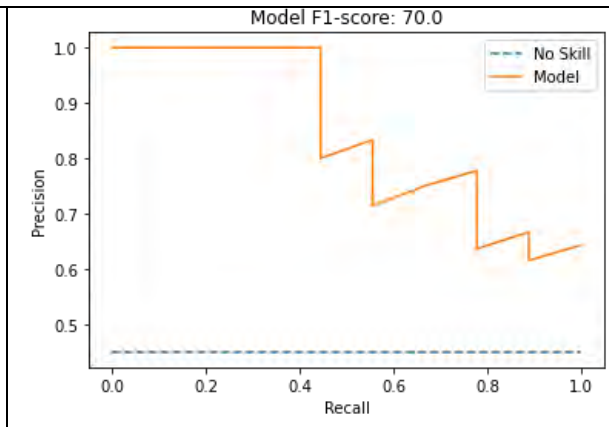
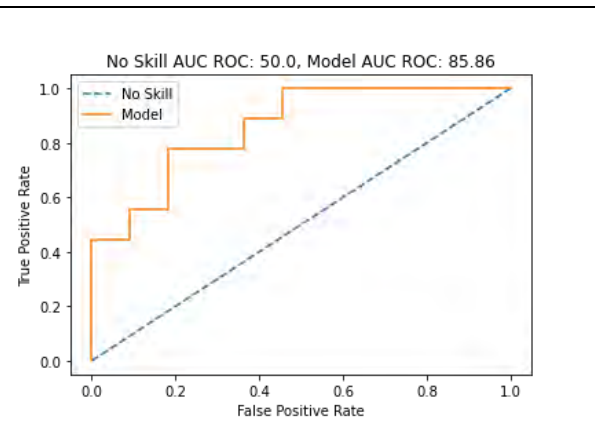
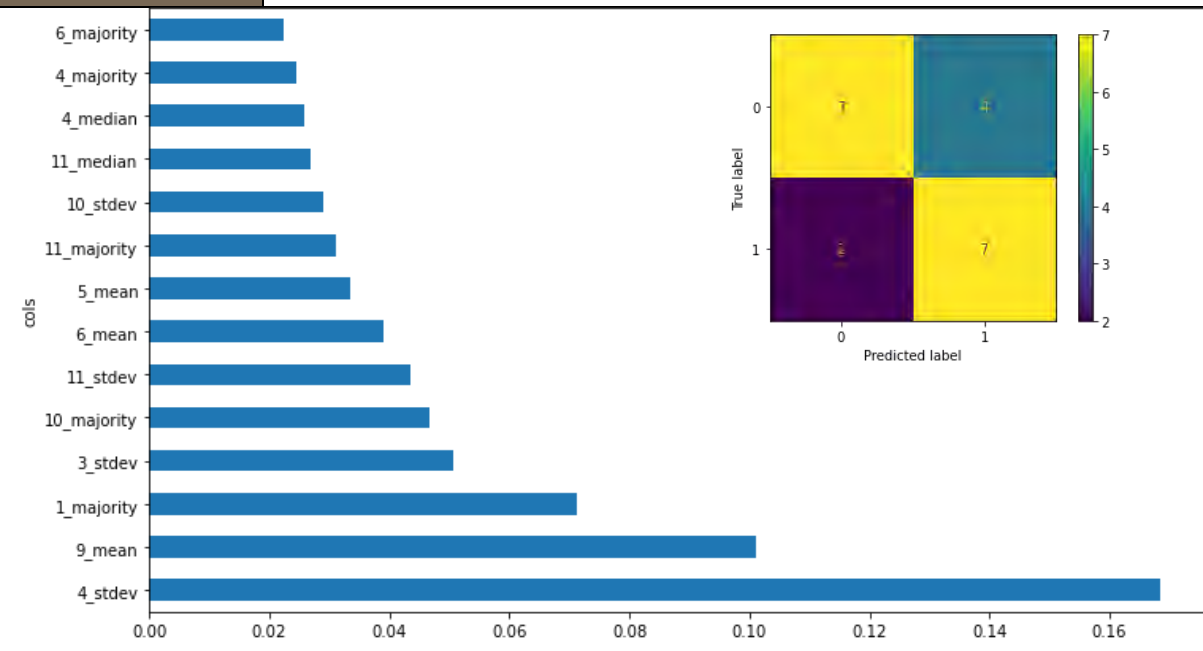
Yacimiento IAN



Ventana	G4	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': None, 'n_estimators': 50}				
Rango (m ²)	802 - 921	Rendimiento (GM)	70.35	ROC AUC	85.06
Model AUC PR	84.37	F1 score	70.0	Índice de Kappa	0.41
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

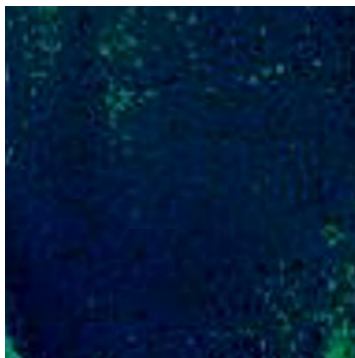


Leyenda imágenes de ejemplo Ventana clasificador Yacimiento IAN

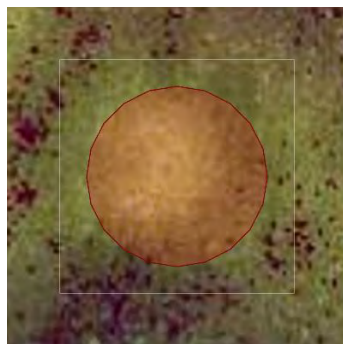


Ventana	G5	Tipo de clasificador	Moderado		
Método	GradientBoostingClassifier(learning_rate=1, max_depth=1, random_state=0)				
Mejor configuración	{ 'max_features': None, 'n_estimators': 10 }				
Rango (m ²)	939 - 1036	Rendimiento (GM)	73.79	ROC AUC	87.22
Model AUC PR	88.57	F1 score	73.68	Índice de Kappa	0.48
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

31



8



23



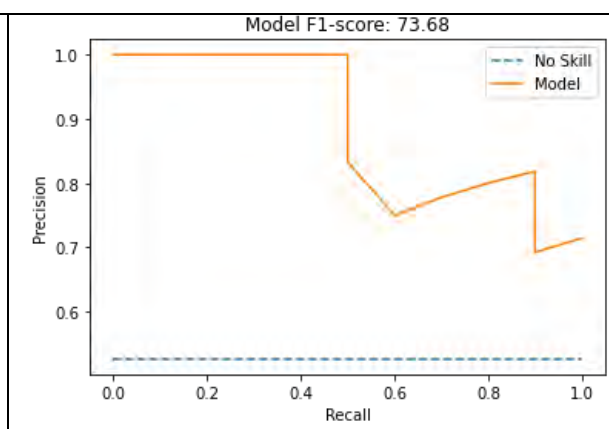
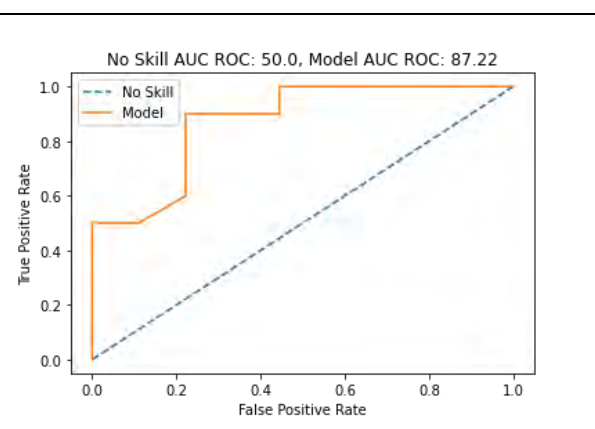
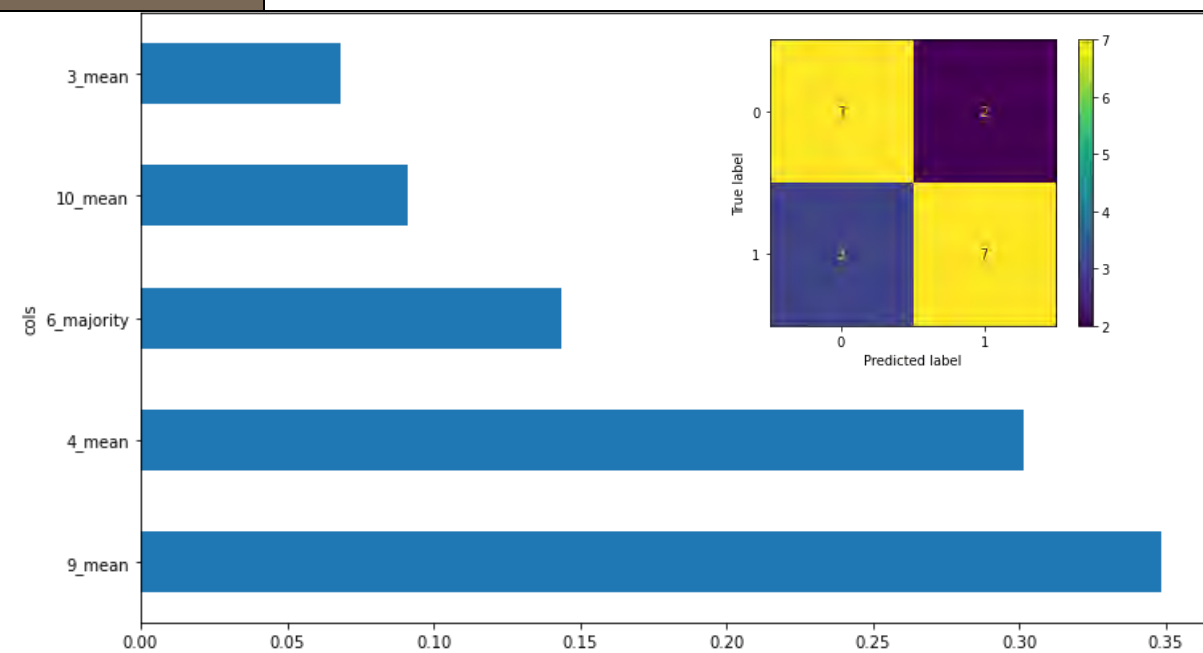
Leyenda imágenes de ejemplo



Ventana clasificador



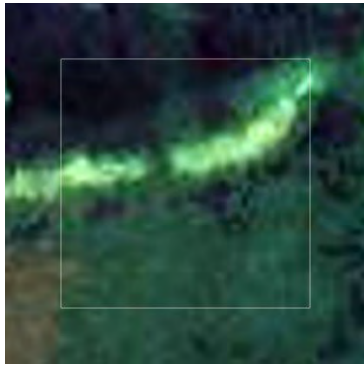
Yacimiento IAN



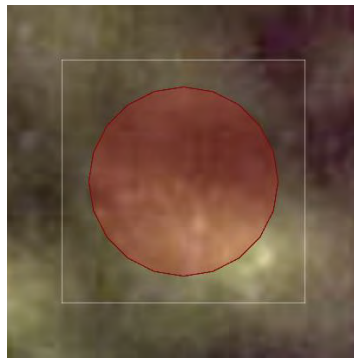
Ventana	G6	Tipo de clasificador	Muy Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 150}				
Rango (m²)	457 - 520	Rendimiento (GM)	97.33	ROC AUC	98.85
Model AUC PR	98.56	F1 score	96.97	Índice de Kappa	0.94

Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)
-----------------------	----------------------------	-------------------------

57



14



43



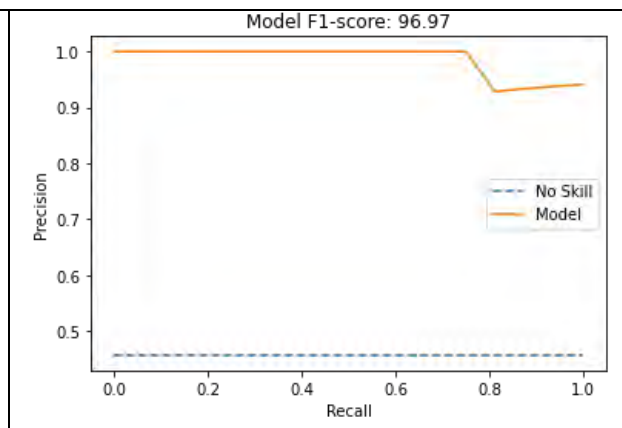
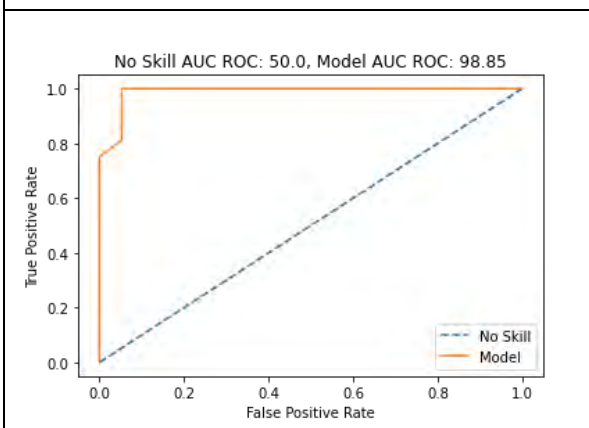
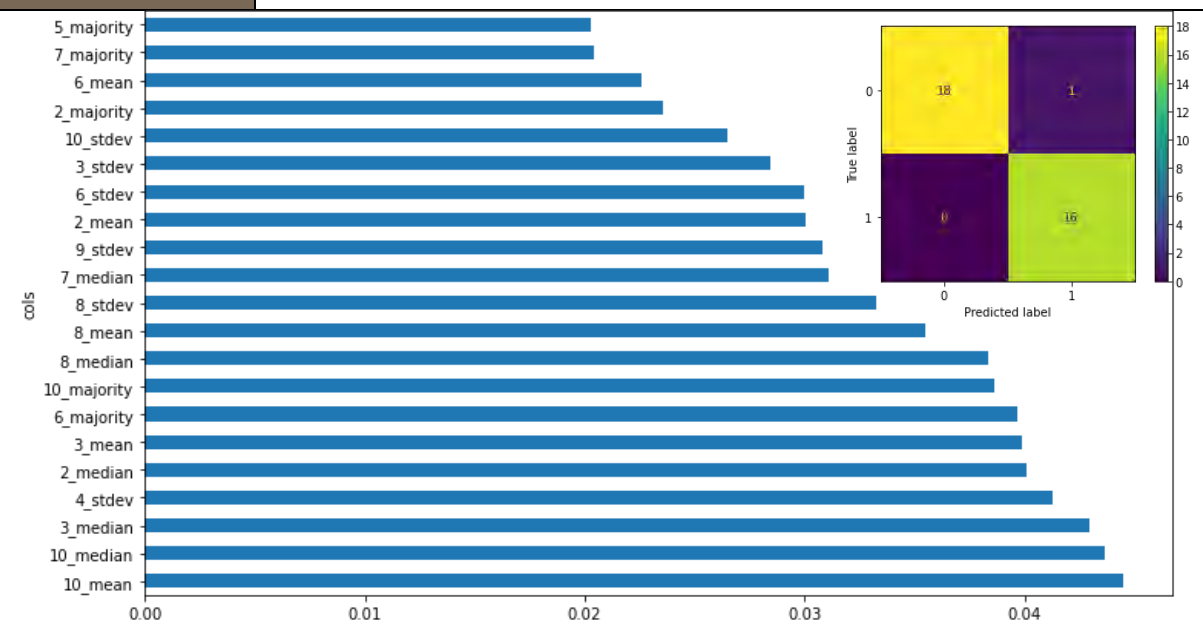
Legenda imágenes de ejemplo



Ventana clasificador

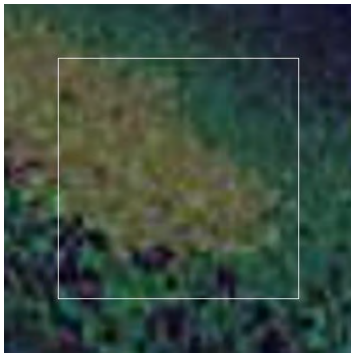


Yacimiento IAN

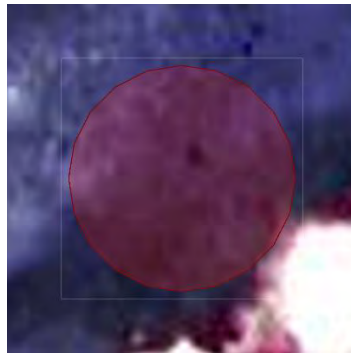


Ventana	G7	Tipo de clasificador	Débil		
Método	GradientBoostingClassifier(learning_rate=1, max_depth=1, random_state=0)				
Mejor configuración	{ 'max_features': 'sqrt', 'n_estimators': 150 }				
Rango (m ²)	397 - 454	Rendimiento (GM)	69.29	ROC AUC	74.68
Model AUC PR	65.54	F1 score	68.75	Índice de Kappa	0.38
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

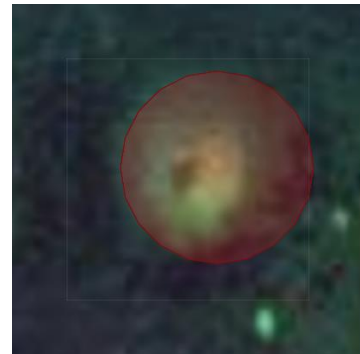
52



13



39



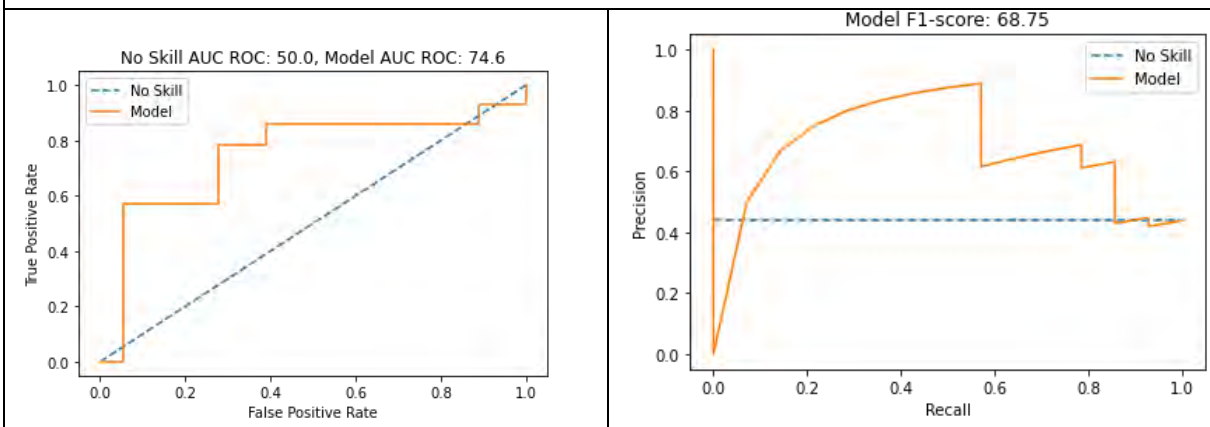
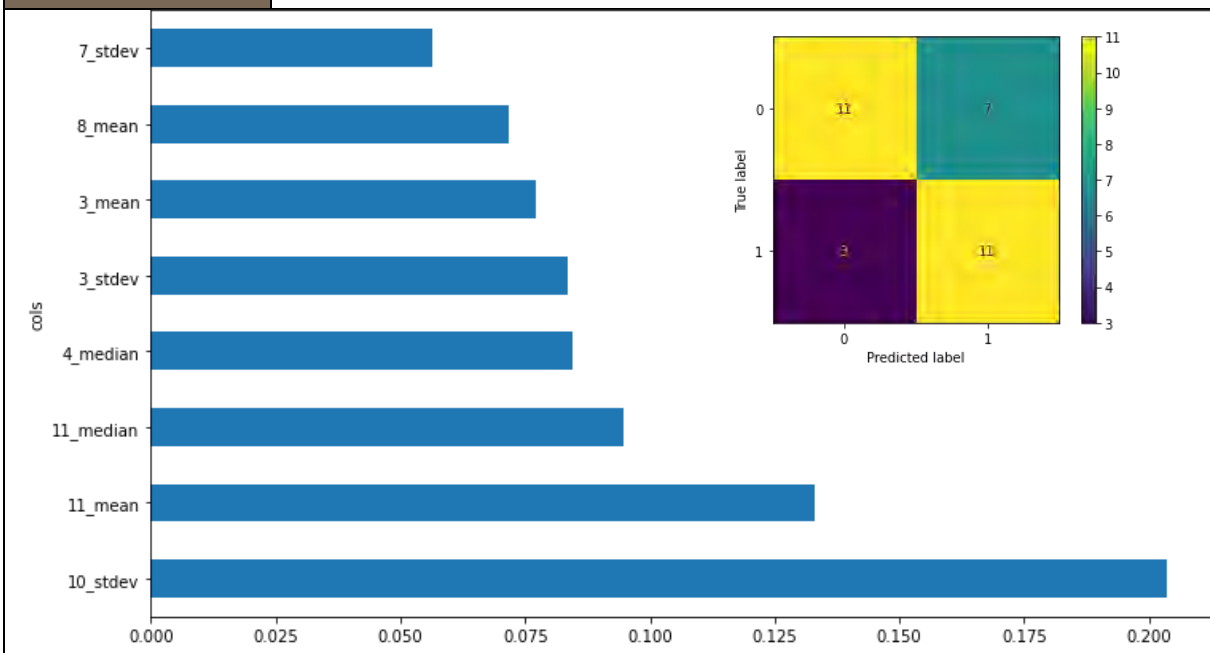
Leyenda imágenes de ejemplo



Ventana clasificador



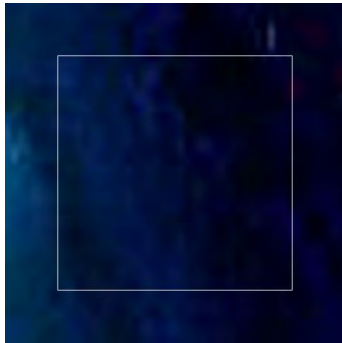
Yacimiento IAN



Ventana	G8	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 200 }				
Rango (m ²)	237 - 318	Rendimiento (GM)	83.0	ROC AUC	94.98
Model AUC PR	96.42	F1 score	81.58	Índice de Kappa	0.67

Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)
----------------	---------------------	------------------

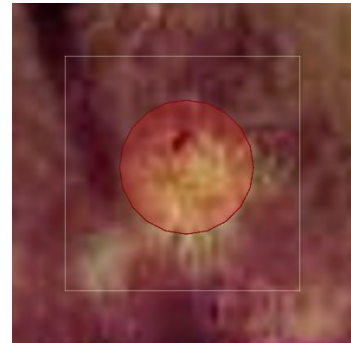
136



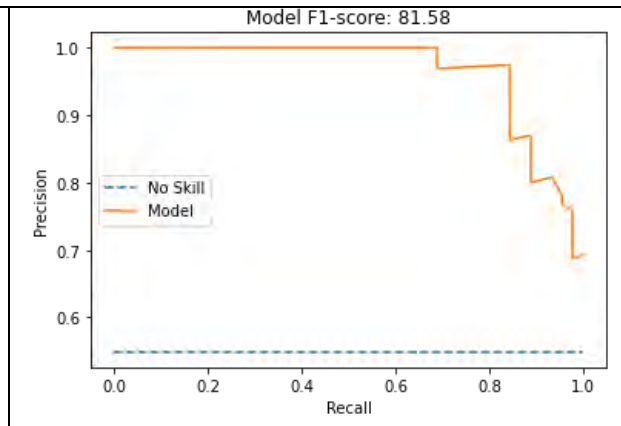
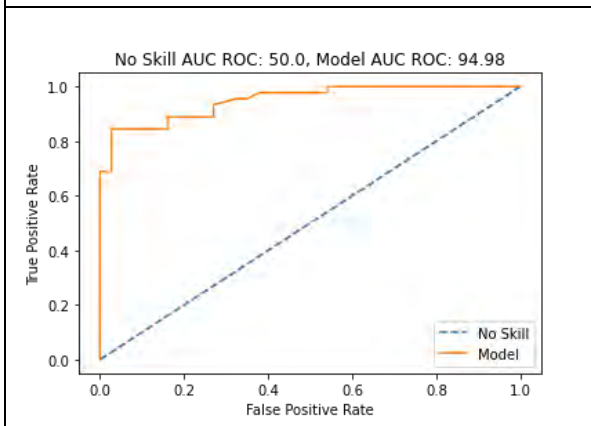
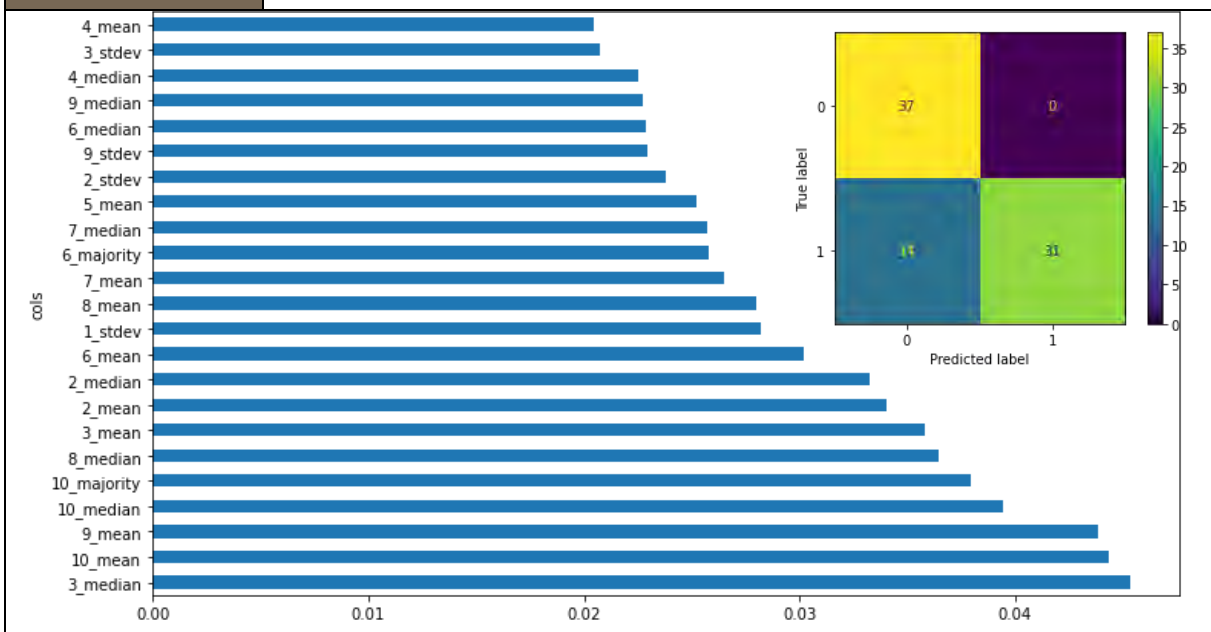
34



102

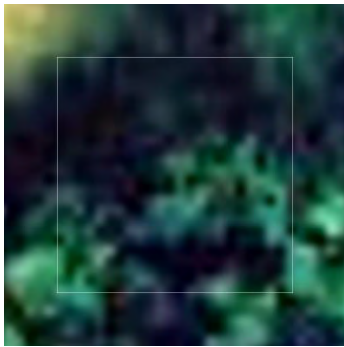


Leyenda imágenes de ejemplo		Ventana clasificador		Yacimiento IAN
-----------------------------	--	----------------------	--	----------------

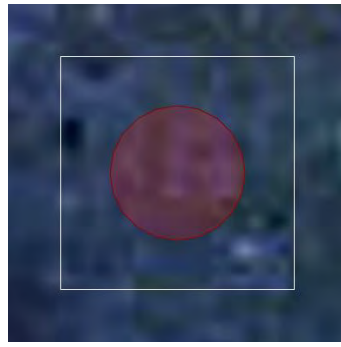


Ventana	G9	Tipo de clasificador	Moderado		
Método	GradientBoostingClassifier(learning_rate=1, max_depth=1, random_state=0)				
Mejor configuración	{ 'max_features': None, 'n_estimators': 200 }				
Rango (m ²)	31 - 138	Rendimiento (GM)	76.03	ROC AUC	84.62
Model AUC PR	91.09	F1 score	81.58	Índice de Kappa	0.51
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

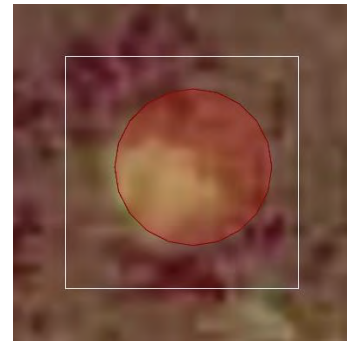
101



15



76



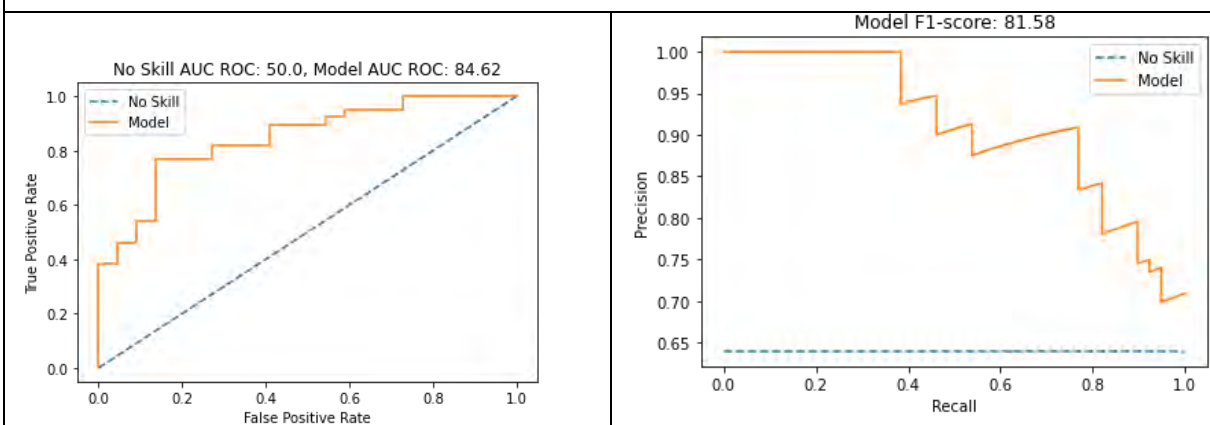
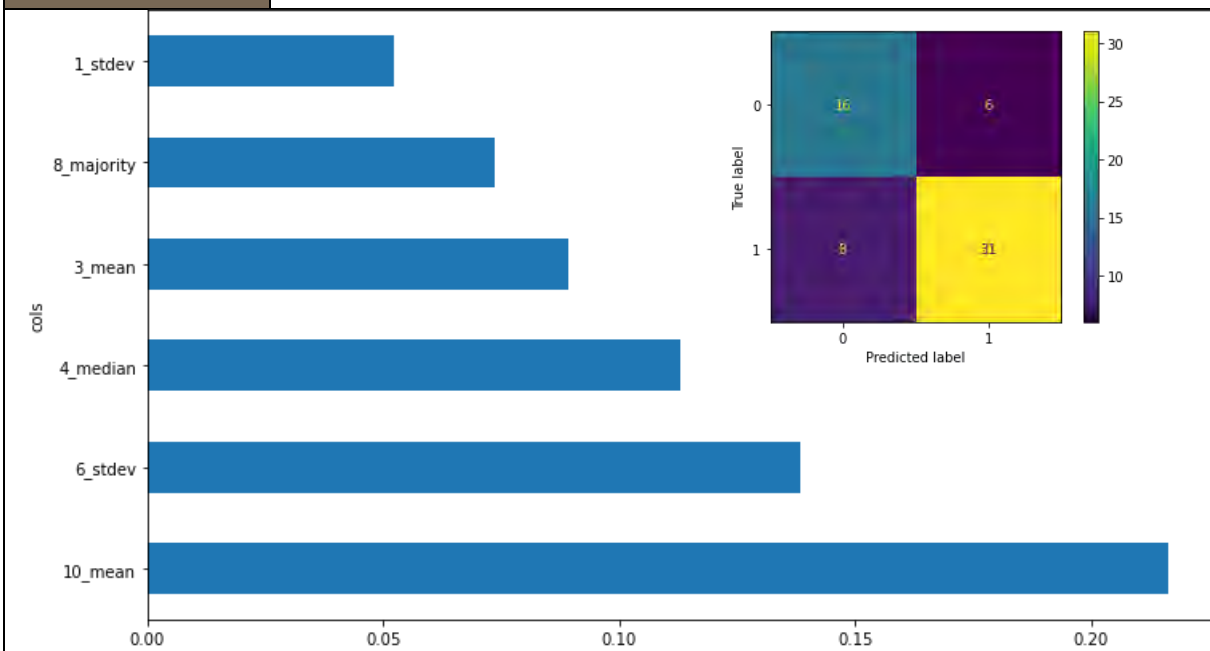
Leyenda imágenes de ejemplo



Ventana clasificador



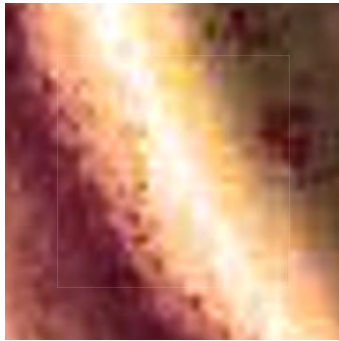
Yacimiento IAN



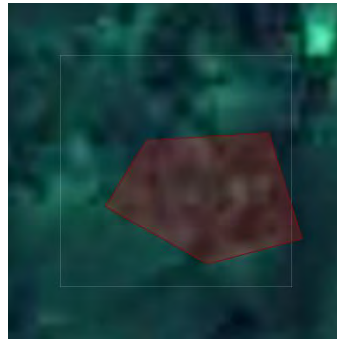
Ventana	G10	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'gini', 'max_features': None, 'n_estimators': 100 }				
Rango (m ²)	142 - 234	Rendimiento (GM)	78.83	ROC AUC	88.04
Model AUC PR	91.12	F1 score	79.45	Índice de Kappa	0.56

Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)
----------------	---------------------	------------------

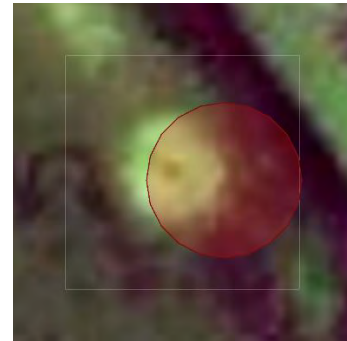
113



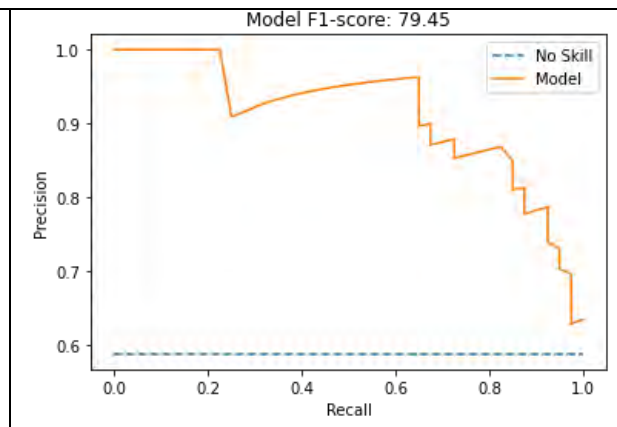
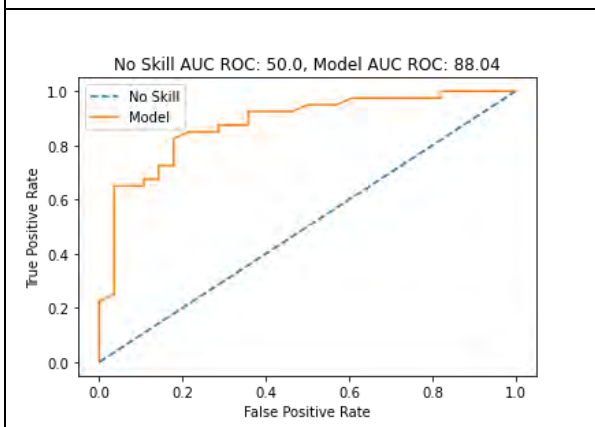
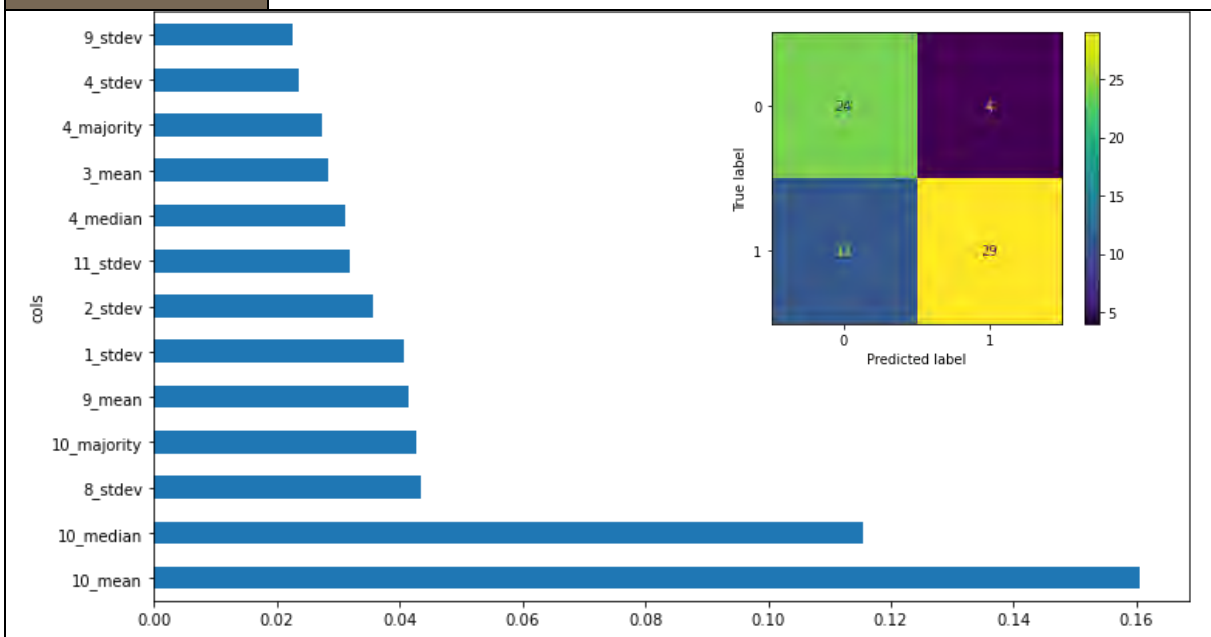
28



85



Leyenda imágenes de ejemplo		Ventana clasificador		Yacimiento IAN
-----------------------------	--	----------------------	--	----------------



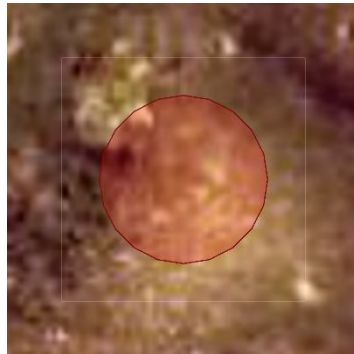
Ventana	G11	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': None, 'n_estimators': 200 }				
Rango (m²)	319 - 451	Rendimiento (GM)	80.81	ROC AUC	88.61
Model AUC PR	88.9	F1 score	78.05	Índice de Kappa	0.62

Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)
----------------	---------------------	------------------

81



20



61



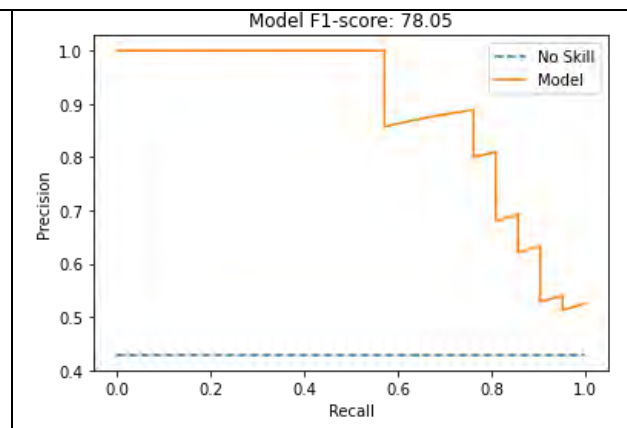
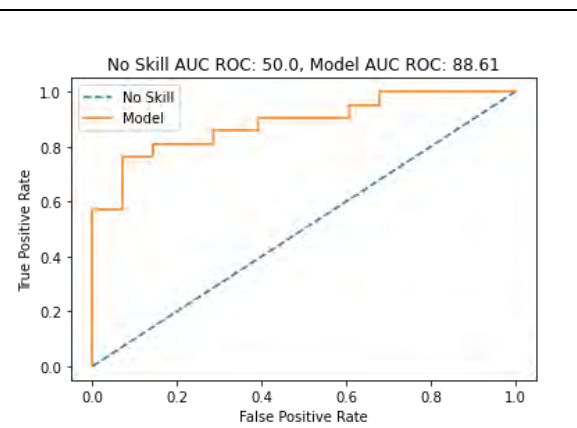
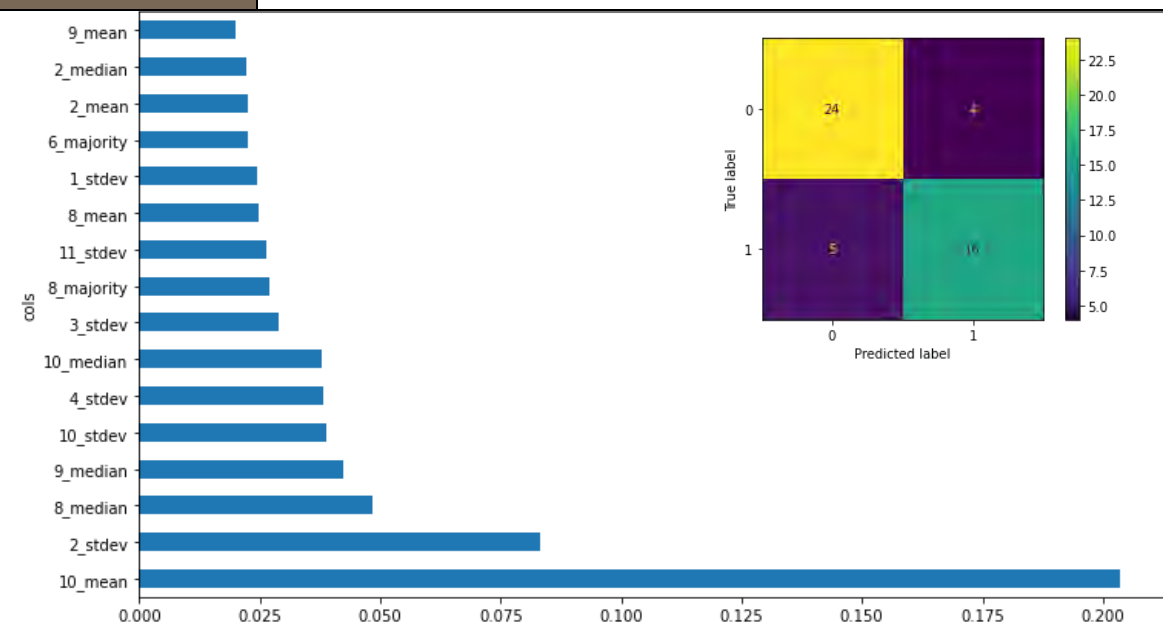
Leyenda imágenes de ejemplo



Ventana clasificador



Yacimiento IAN



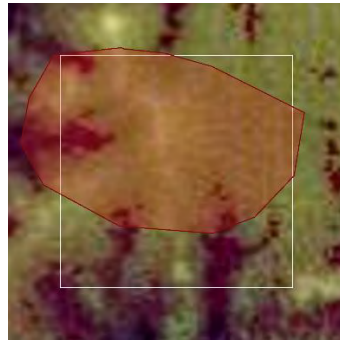
Ventana	G12	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 50 }				
Rango (m ²)	675 - 794	Rendimiento (GM)	85.84	ROC AUC	90.95
Model AUC PR	84.4	F1 score	84.85	Índice de Kappa	0.71

Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)
----------------	---------------------	------------------

57



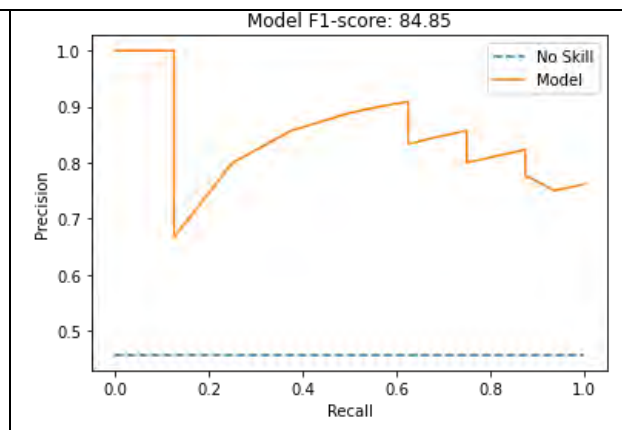
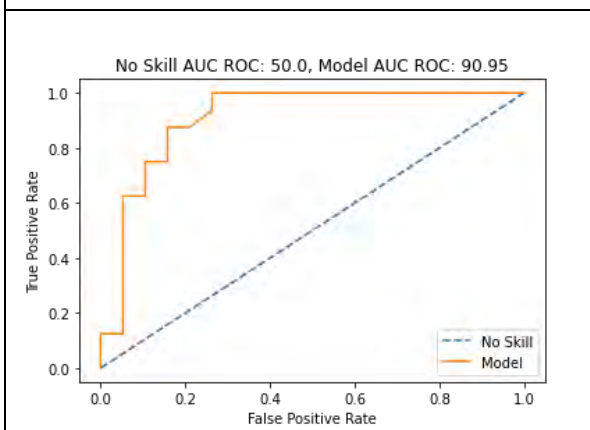
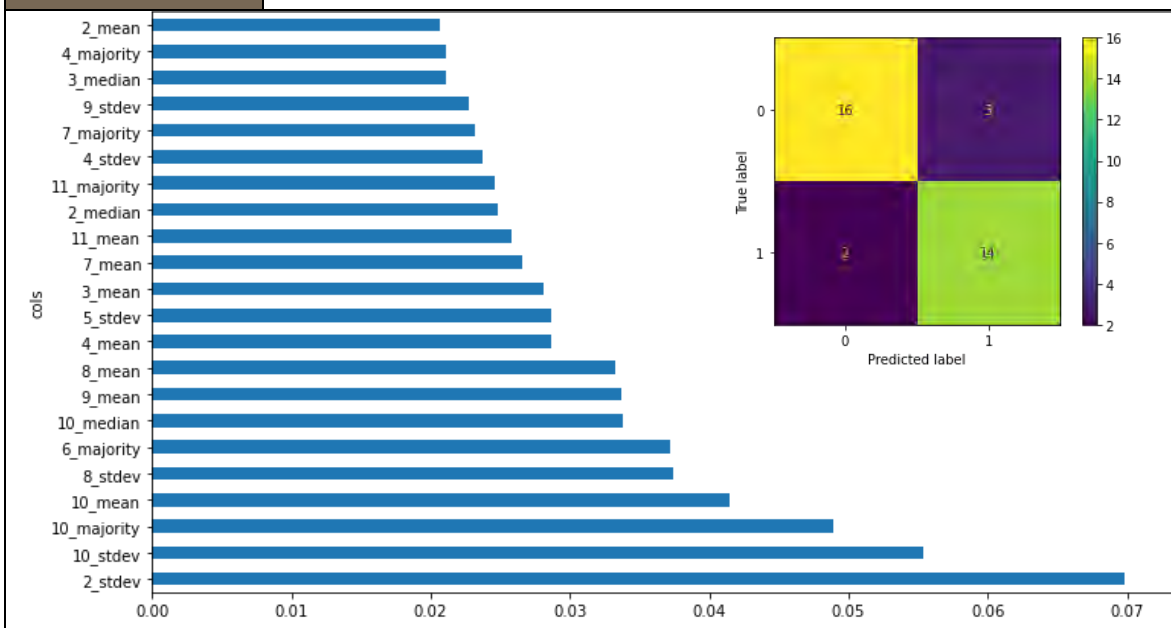
14



43

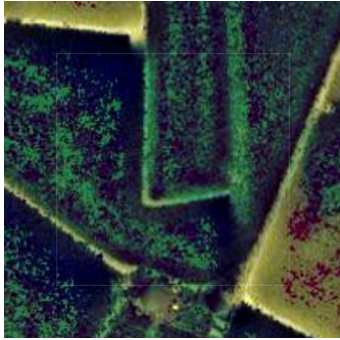


Leyenda imágenes de ejemplo Ventana clasificador Yacimiento IAN

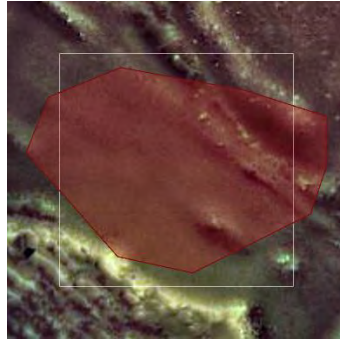


Ventana	G13	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 100}				
Rango (m ²)	4724 - 5118	Rendimiento (GM)	85.28	ROC AUC	94.44
Model AUC PR	94.63	F1 score	84.21	Índice de Kappa	0.7
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

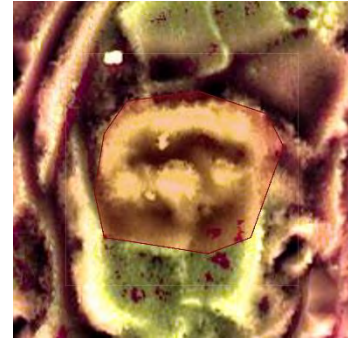
33



8



25



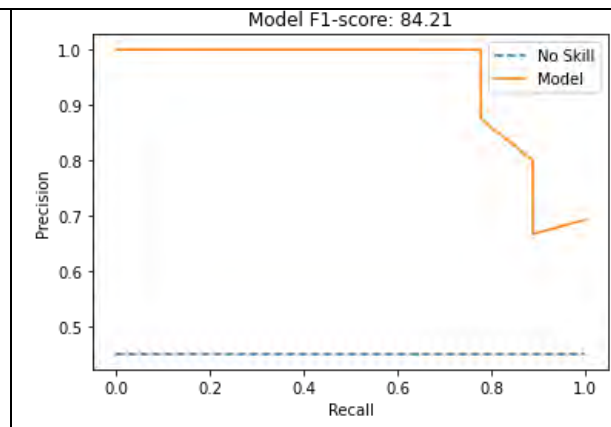
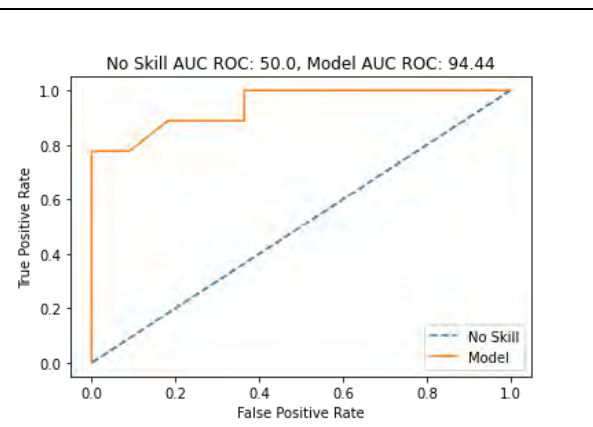
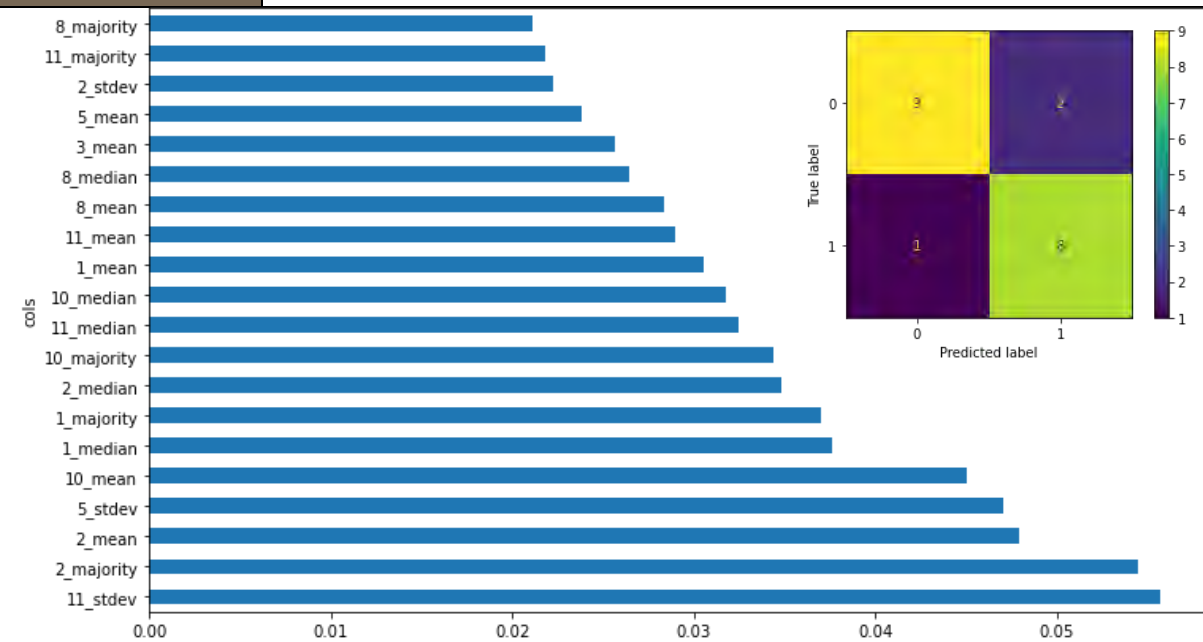
Leyenda imágenes de ejemplo



Ventana clasificador

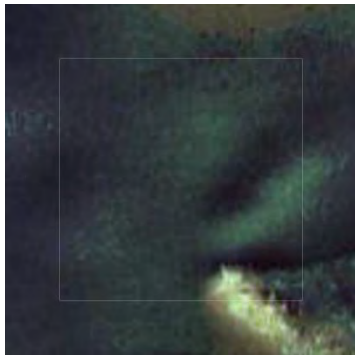


Yacimiento IAN

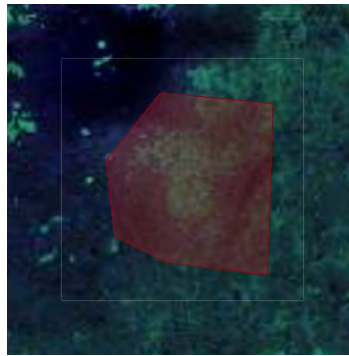


Ventana	G14	Tipo de clasificador	Pobre		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 50 }				
Rango (m ²)	1677 - 2059	Rendimiento (GM)	49.72	ROC AUC	68.96
Model AUC PR	71.93	F1 score	58.06	Índice de Kappa	0.05
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

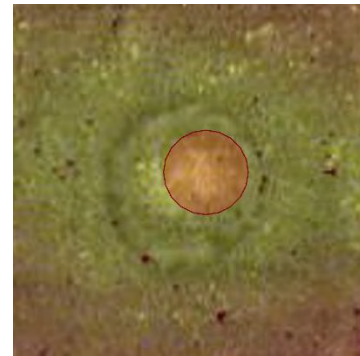
44



11



33



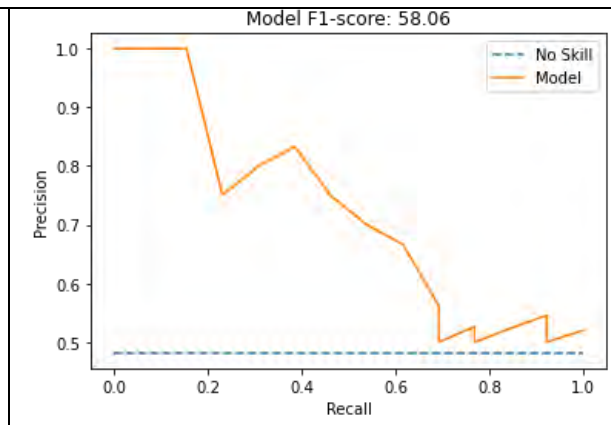
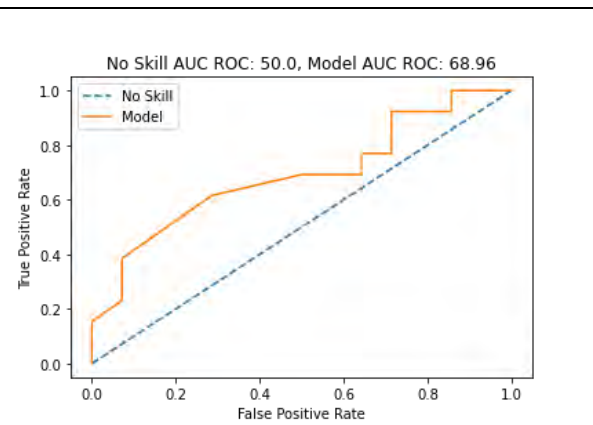
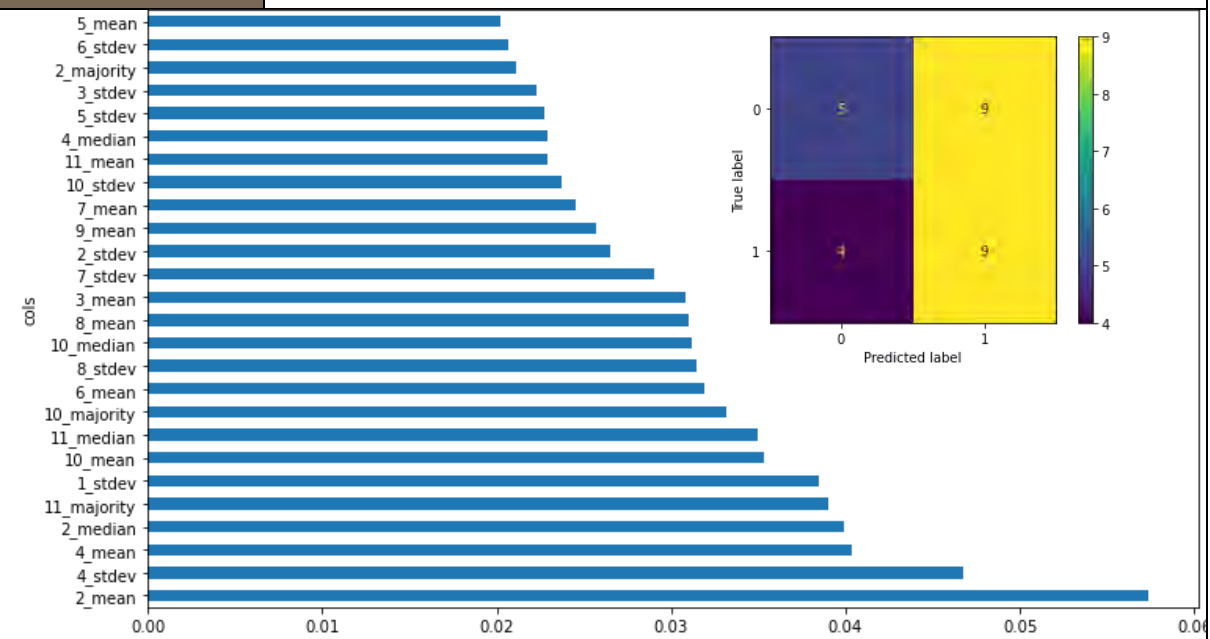
Leyenda imágenes de ejemplo



Ventana clasificador

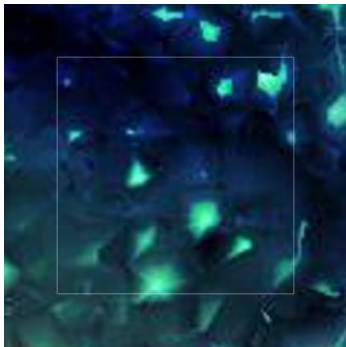


Yacimiento IAN



Ventana	G15	Tipo de clasificador	Bueno		
Método	GradientBoostingClassifier(learning_rate=1, max_depth=1, random_state=0)				
Mejor configuración	{ 'max_features': 'log2', 'n_estimators': 25 }				
Rango (m ²)	1143 - 1648	Rendimiento (GM)	88.8	ROC AUC	90.10
Model AUC PR	92.14	F1 score	87.8	Índice de Kappa	0.78
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

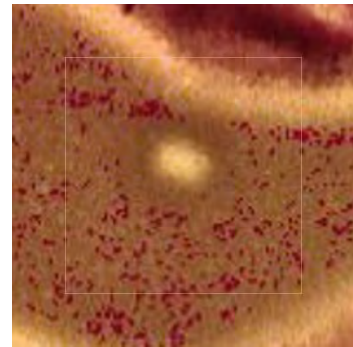
76



19



57



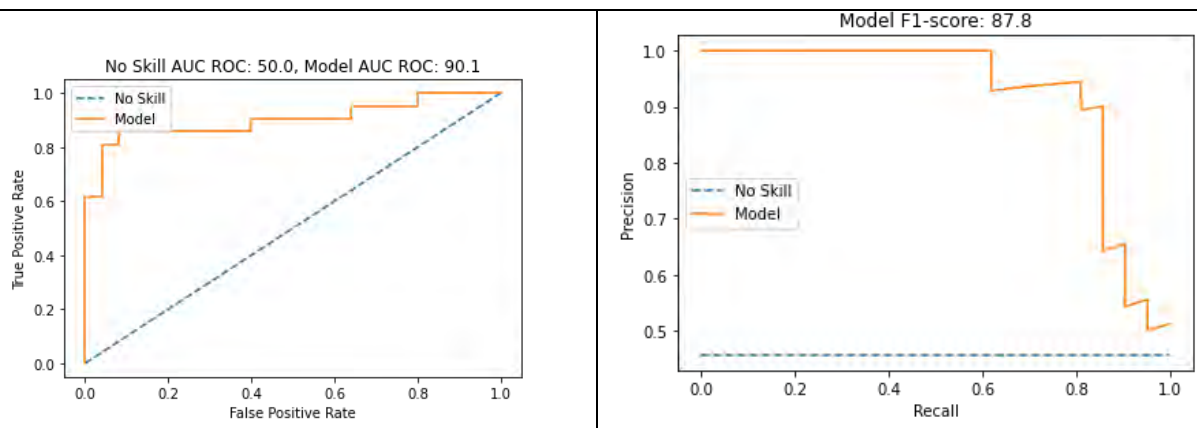
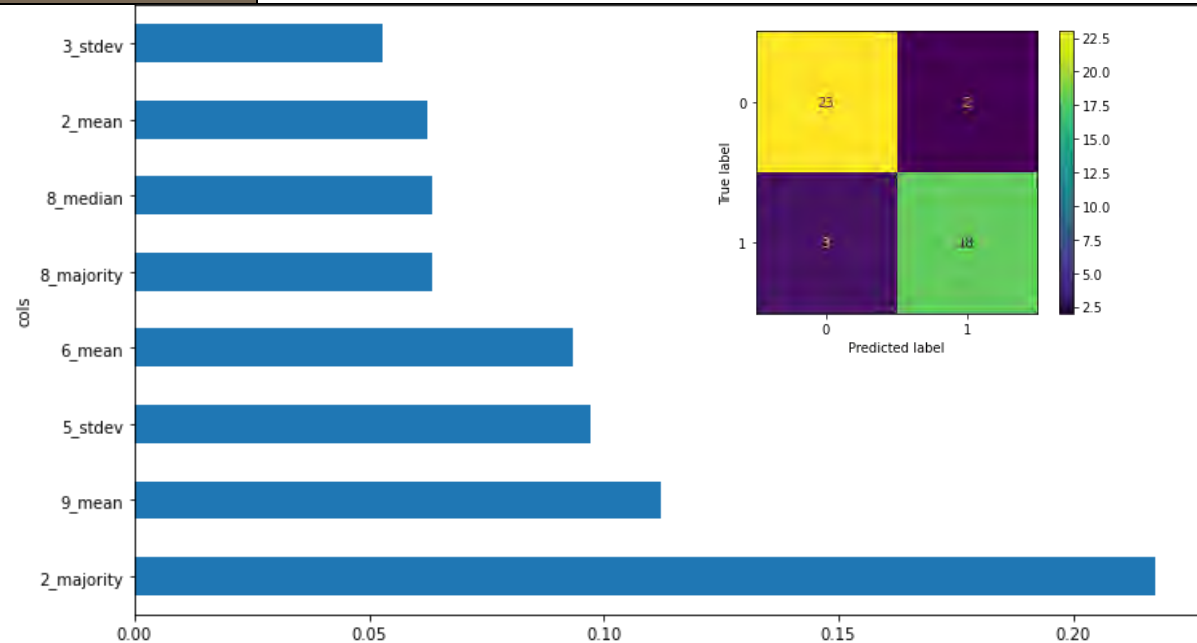
Leyenda imágenes de ejemplo



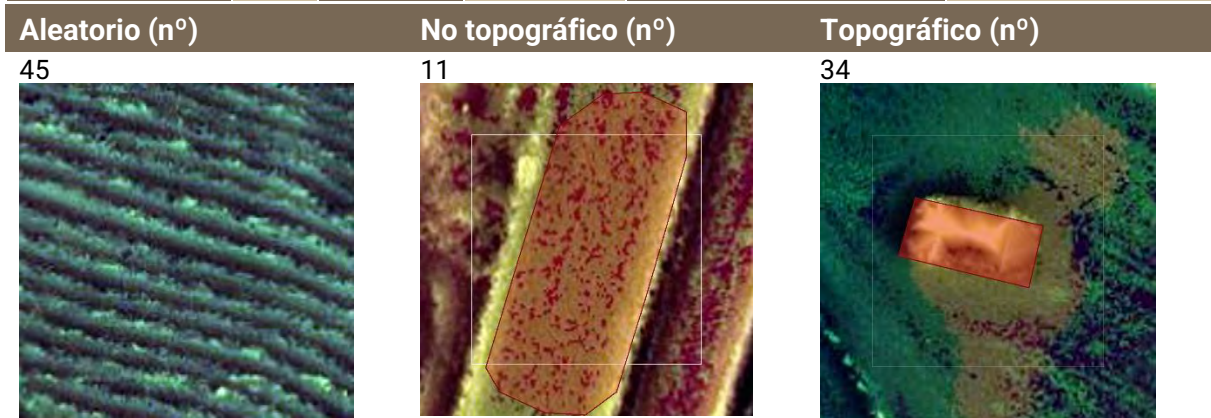
Ventana clasificador



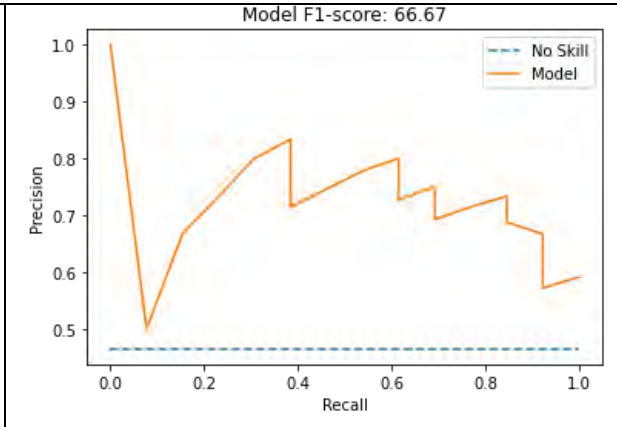
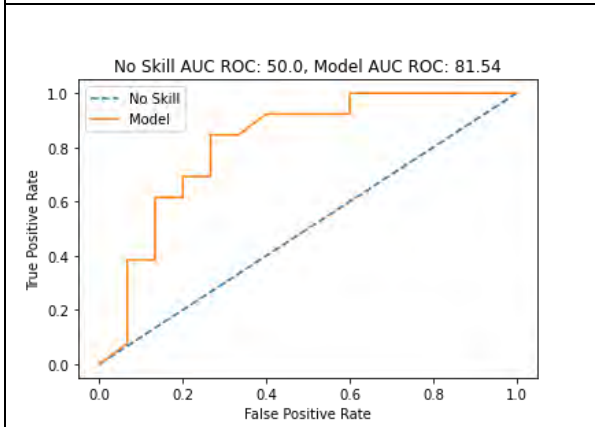
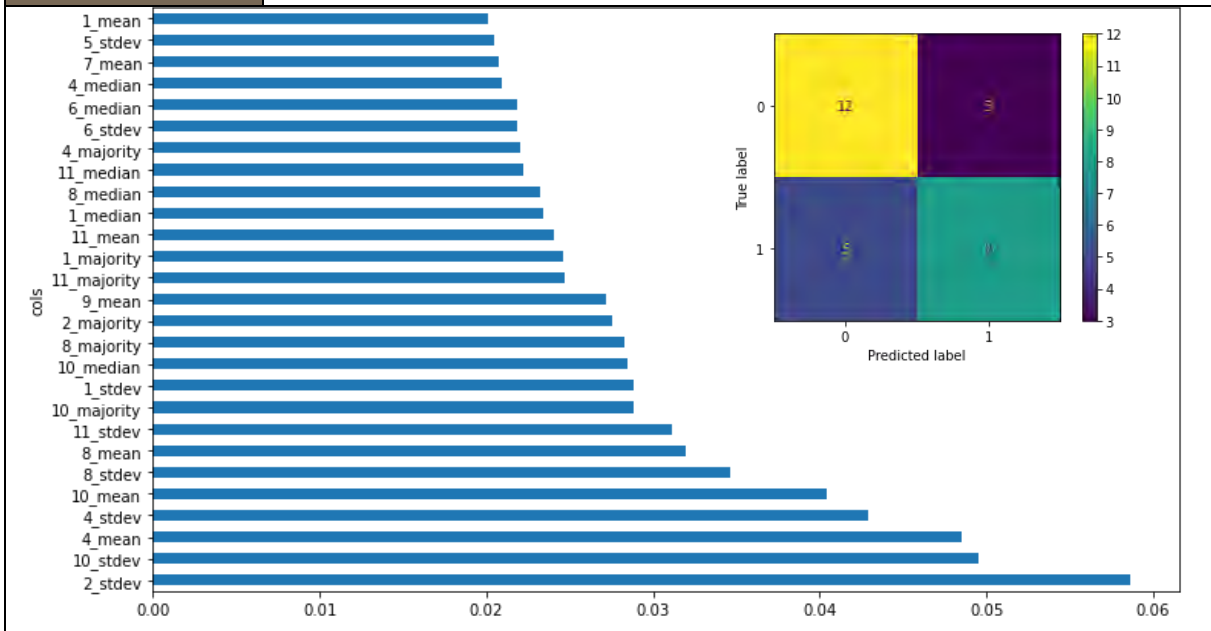
Yacimiento IAN



Ventana	G16	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 50}				
Rango (m²)	2088 - 2458	Rendimiento (GM)	70.16	ROC AUC	81.54
Model AUC PR	71.7	F1 score	66.67	Índice de Kappa	0.42

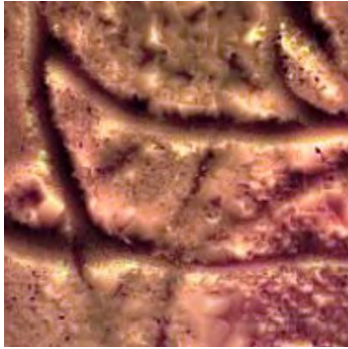


Leyenda imágenes de ejemplo Ventana clasificador Yacimiento IAN

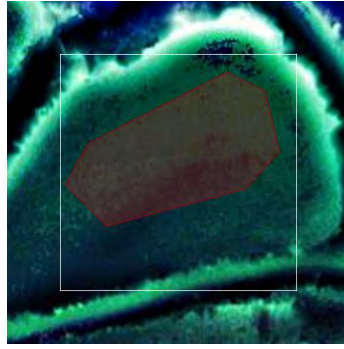


Ventana	G17	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 50}				
Rango (m ²)	3226 - 3950	Rendimiento (GM)	72.06	ROC AUC	84.86
Model AUC PR	81.75	F1 score	69.23	Índice de Kappa	0.44
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

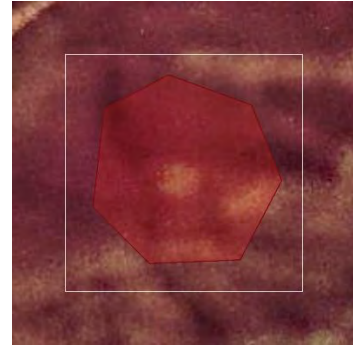
48



12



36



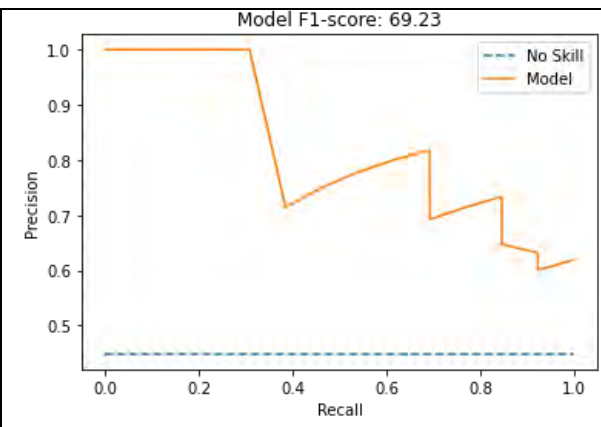
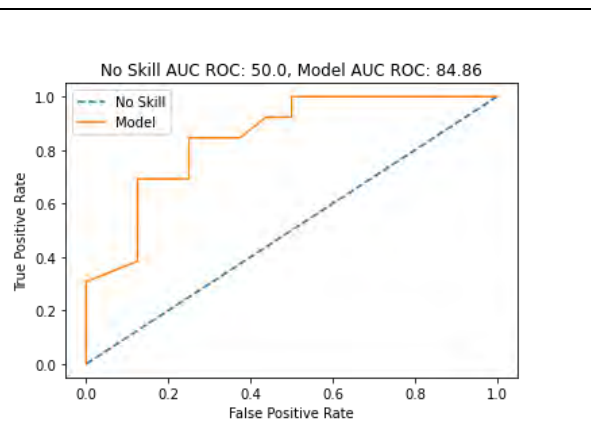
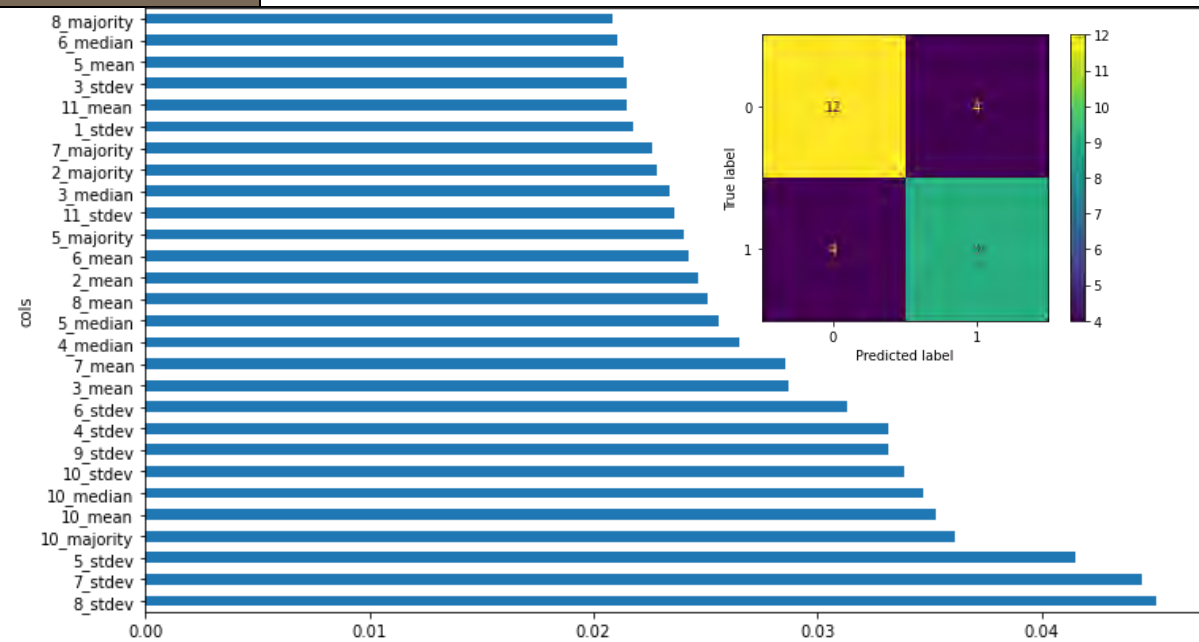
Leyenda imágenes de ejemplo



Ventana clasificador

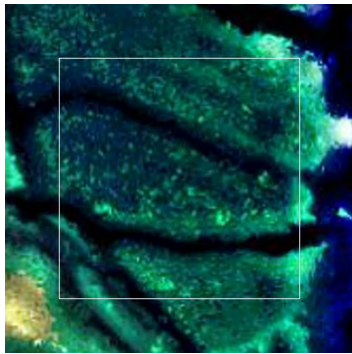


Yacimiento IAN

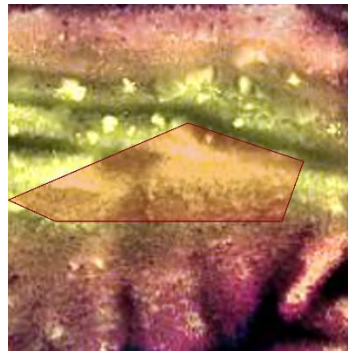


Ventana	G18	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 150}				
Rango (m ²)	2326 - 3129	Rendimiento (GM)	72.06	ROC AUC	84.38
Model AUC PR	81.74	F1 score	69.23	Índice de Kappa	0.44
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

48



12



36



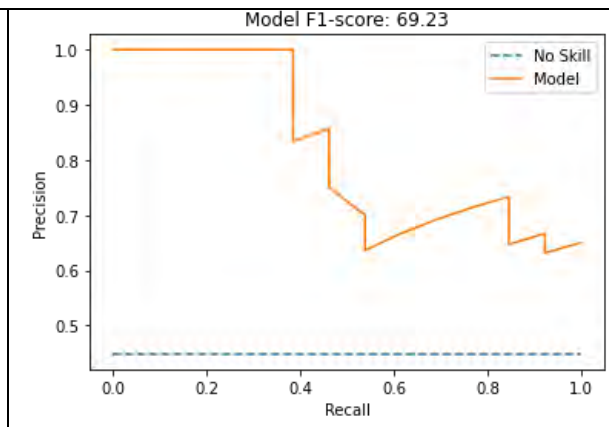
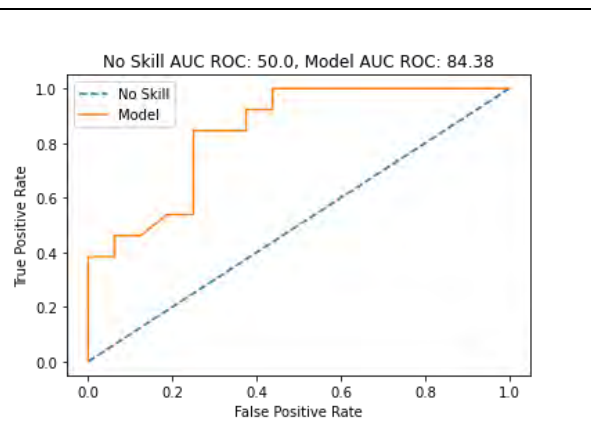
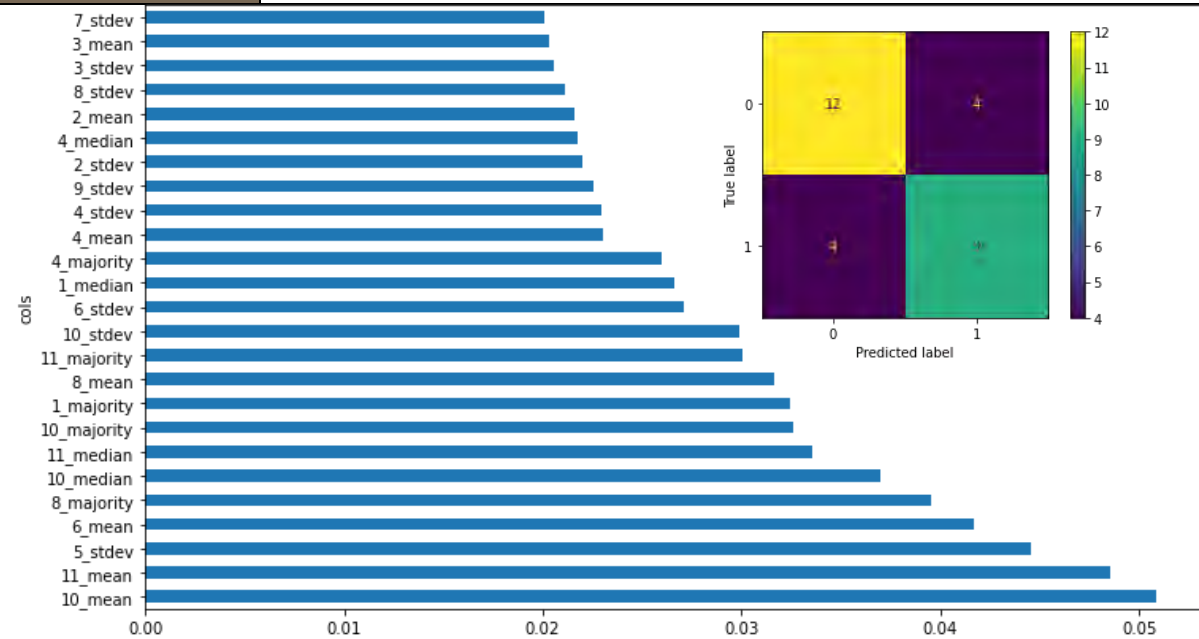
Leyenda imágenes de ejemplo



Ventana clasificador

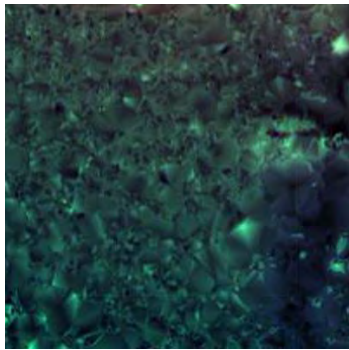


Yacimiento IAN



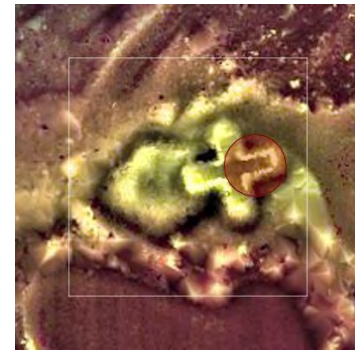
Ventana	G19	Tipo de clasificador	Débil		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': None, 'n_estimators': 150 }				
Rango (m ²)	5189 - 5899	Rendimiento (GM)	60.86	ROC AUC	79.63
Model AUC PR	86.33	F1 score	63.16	Índice de Kappa	0.22
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

29



0

29



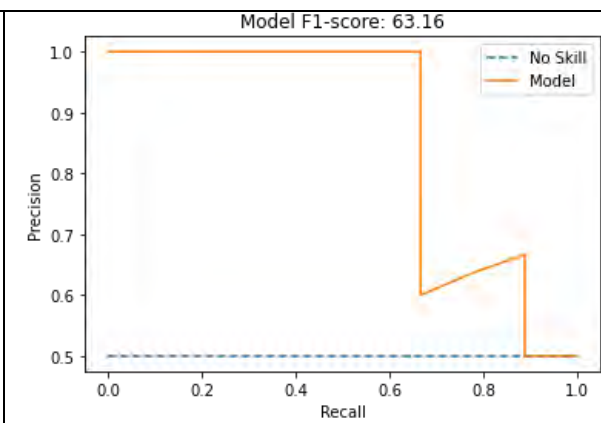
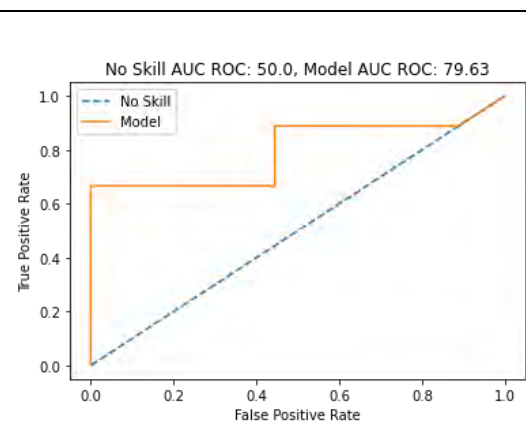
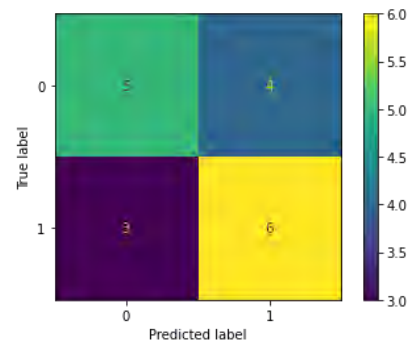
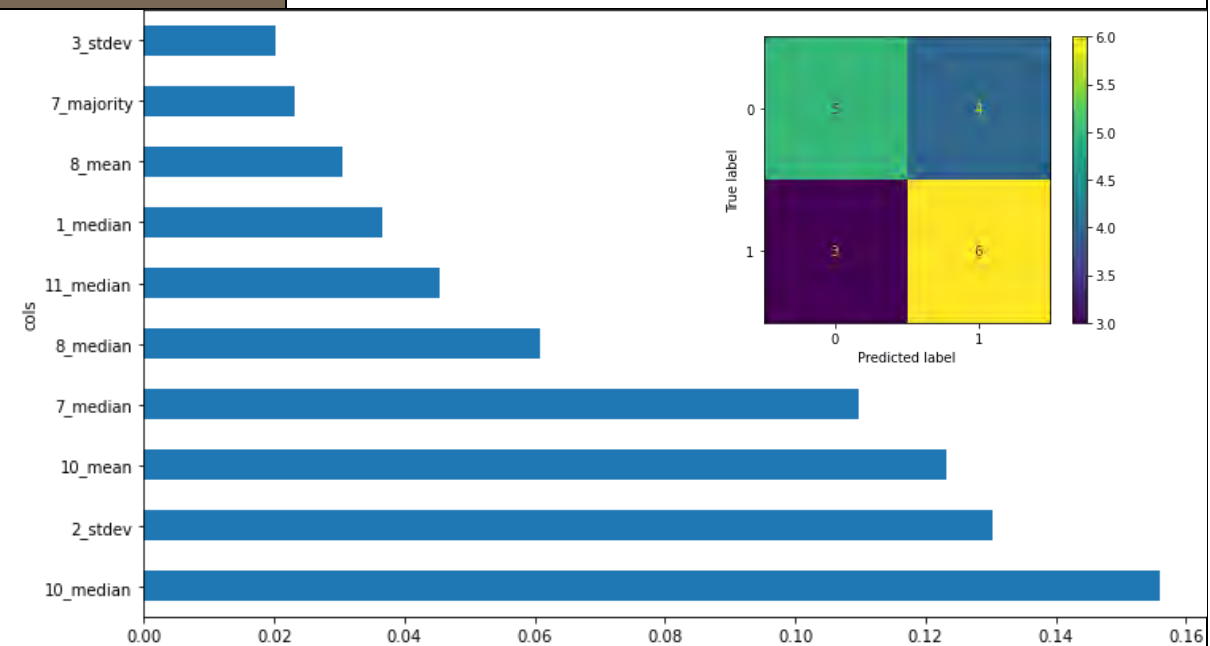
Leyenda imágenes de ejemplo



Ventana clasificador

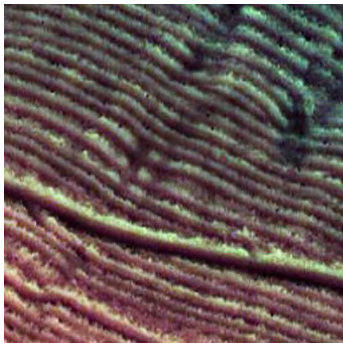


Yacimiento IAN

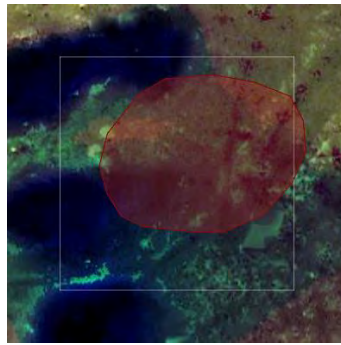


Ventana	G20	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 50 }				
Rango (m ²)	6603 - 7550	Rendimiento (GM)	83.21	ROC AUC	91.61
Model AUC PR	90.28	F1 score	84.62	Índice de Kappa	0.67
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

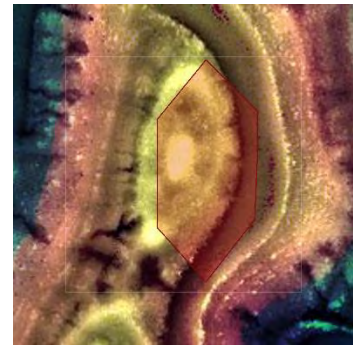
40



10



30



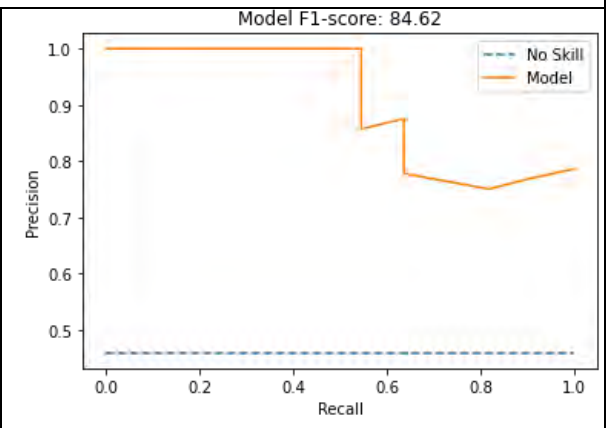
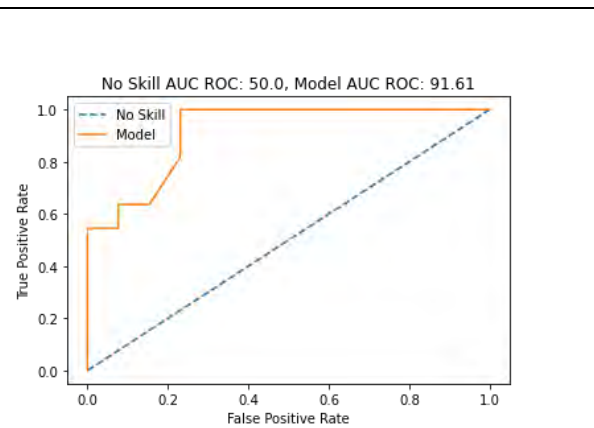
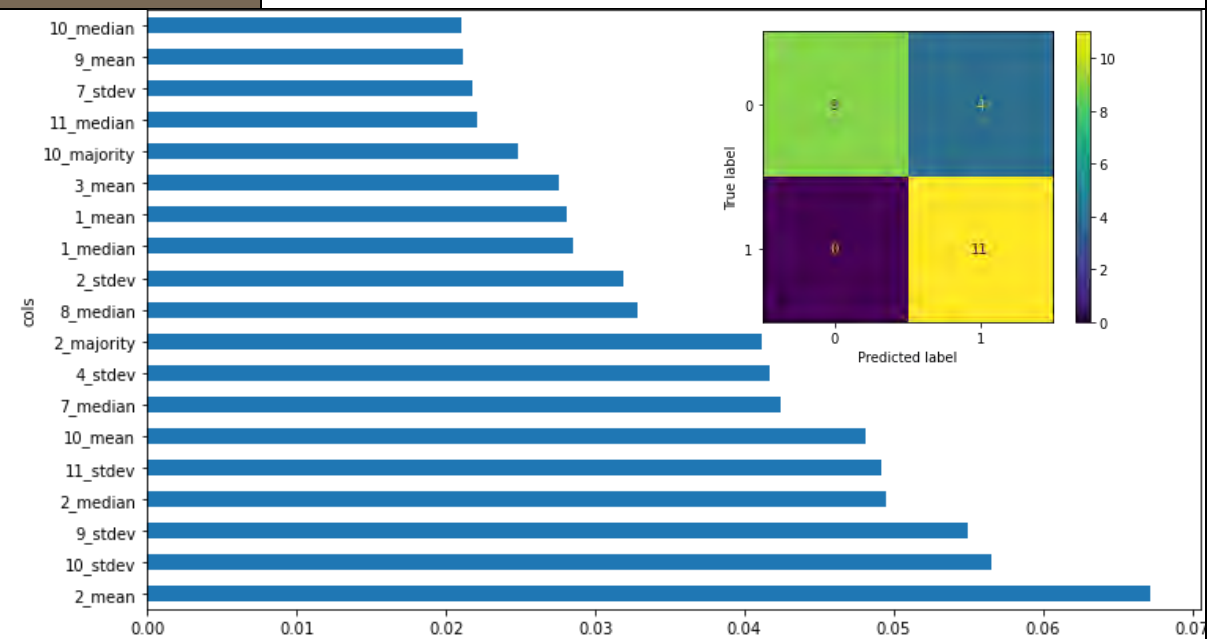
Leyenda imágenes de ejemplo



Ventana clasificador

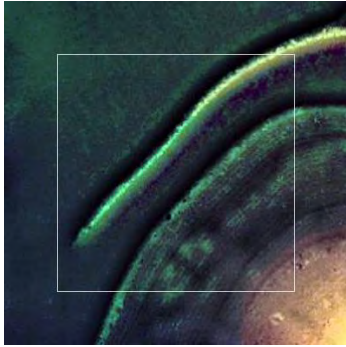


Yacimiento IAN

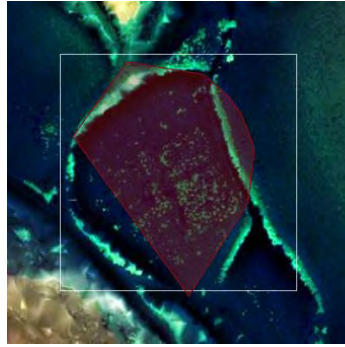


Ventana	G21	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 150}				
Rango (m ²)	7564 - 8433	Rendimiento (GM)	79.33	ROC AUC	91.96
Model AUC PR	91.44	F1 score	80.0	Índice de Kappa	0.59
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

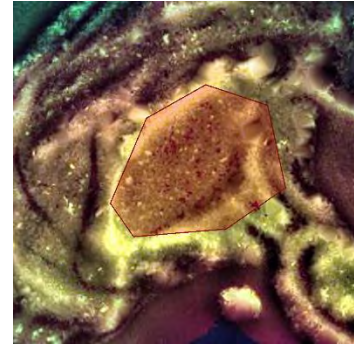
40



10



30



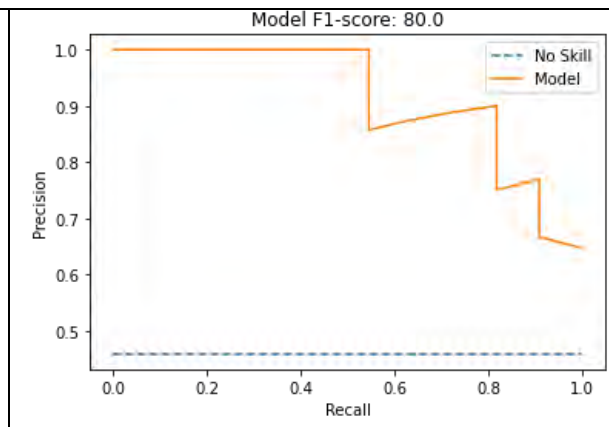
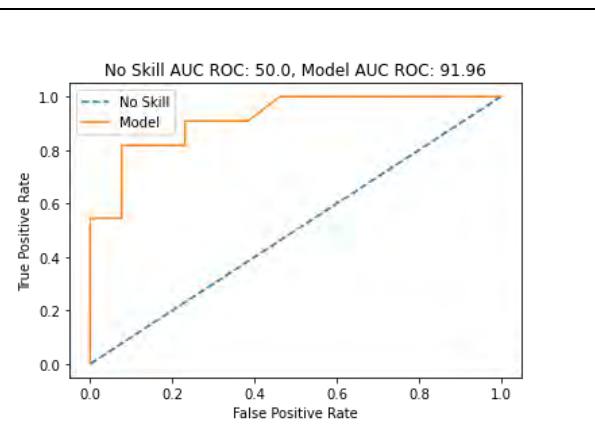
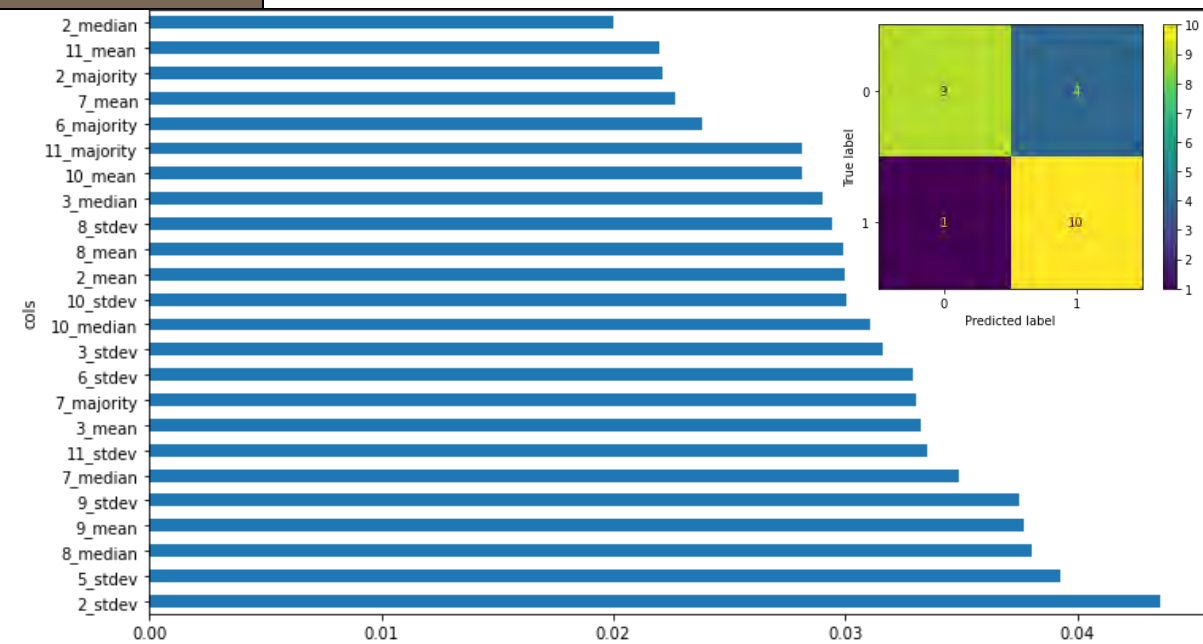
Leyenda imágenes de ejemplo



Ventana clasificador

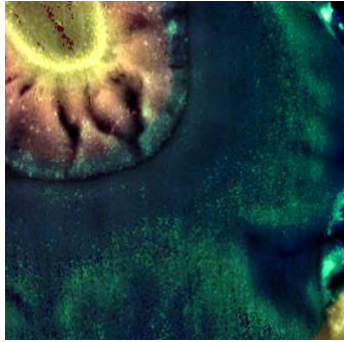


Yacimiento IAN

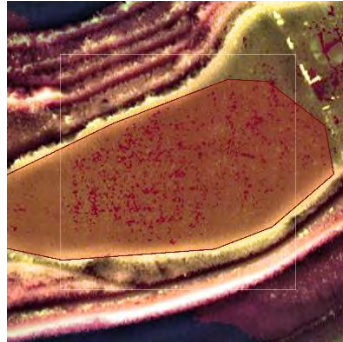


Ventana	G22	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 150 }				
Rango (m ²)	10612 - 11429	Rendimiento (GM)	70.35	ROC AUC	84.34
Model AUC PR	78.53	F1 score	70.0	Índice de Kappa	0.41
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

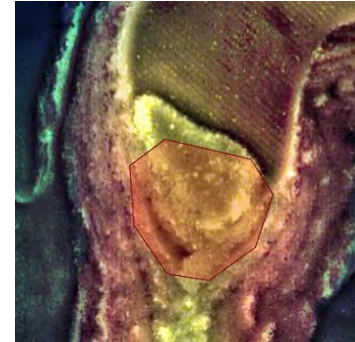
33



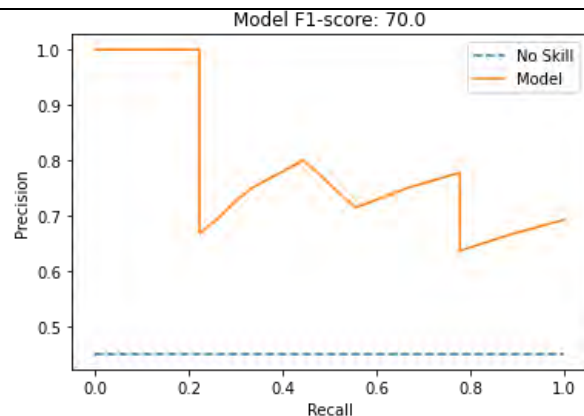
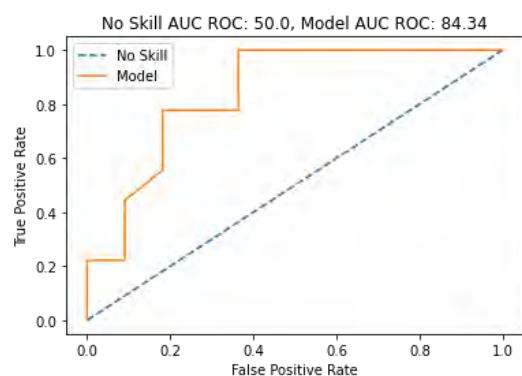
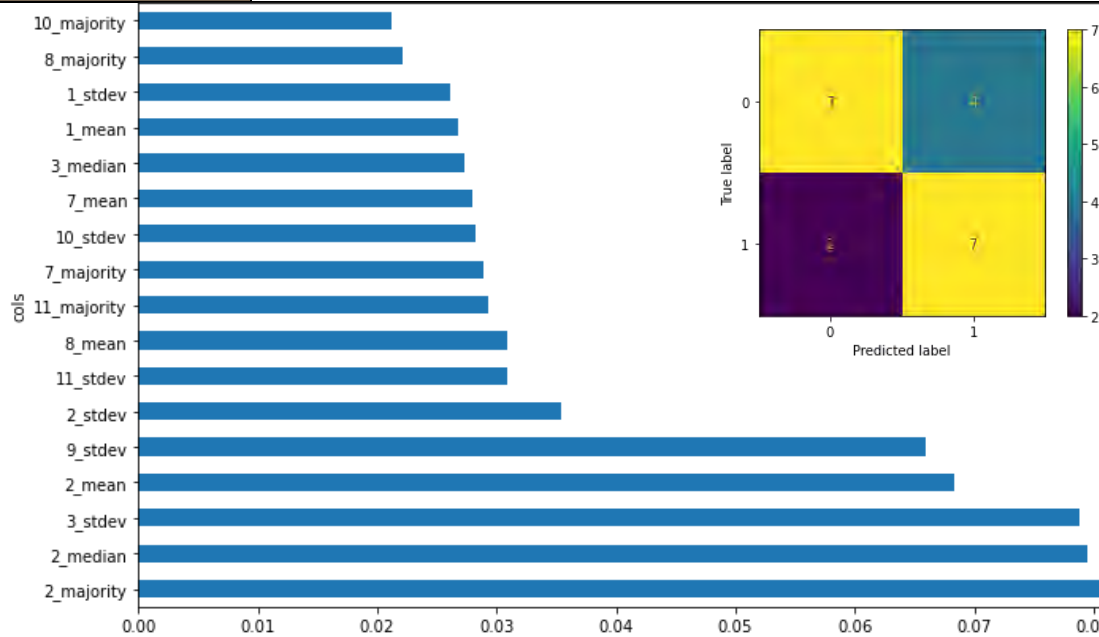
8



25

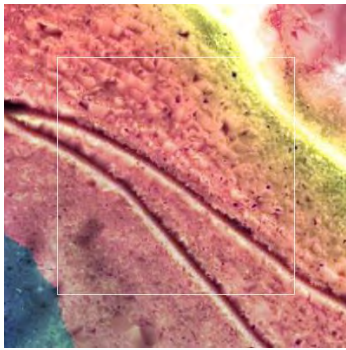


Leyenda imágenes de ejemplo Ventana clasificador Yacimiento IAN

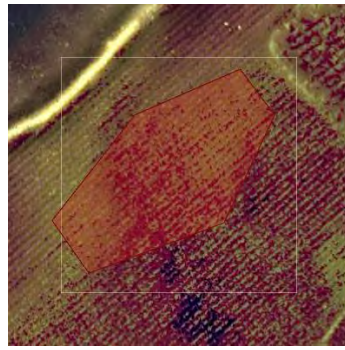


Ventana	G23	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 50 }				
Rango (m ²)	9909 -10577	Rendimiento (GM)	70.16	ROC AUC	84.62
Model AUC PR	85.36	F1 score	69.57	Índice de Kappa	0.4
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

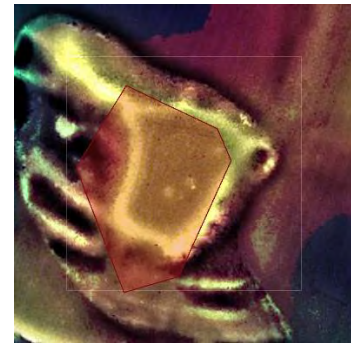
37



9



28



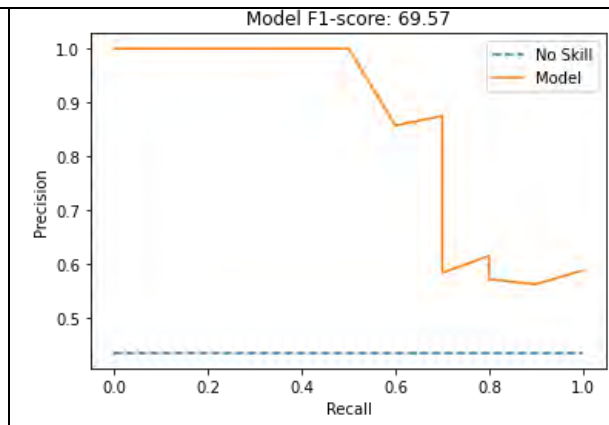
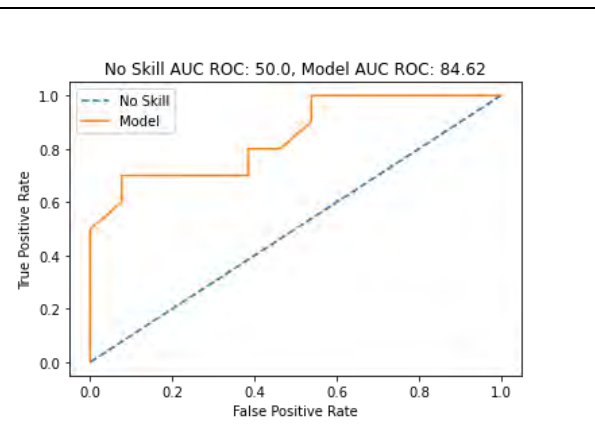
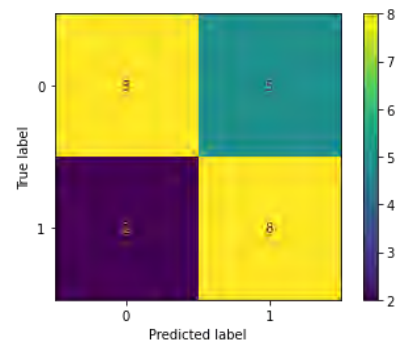
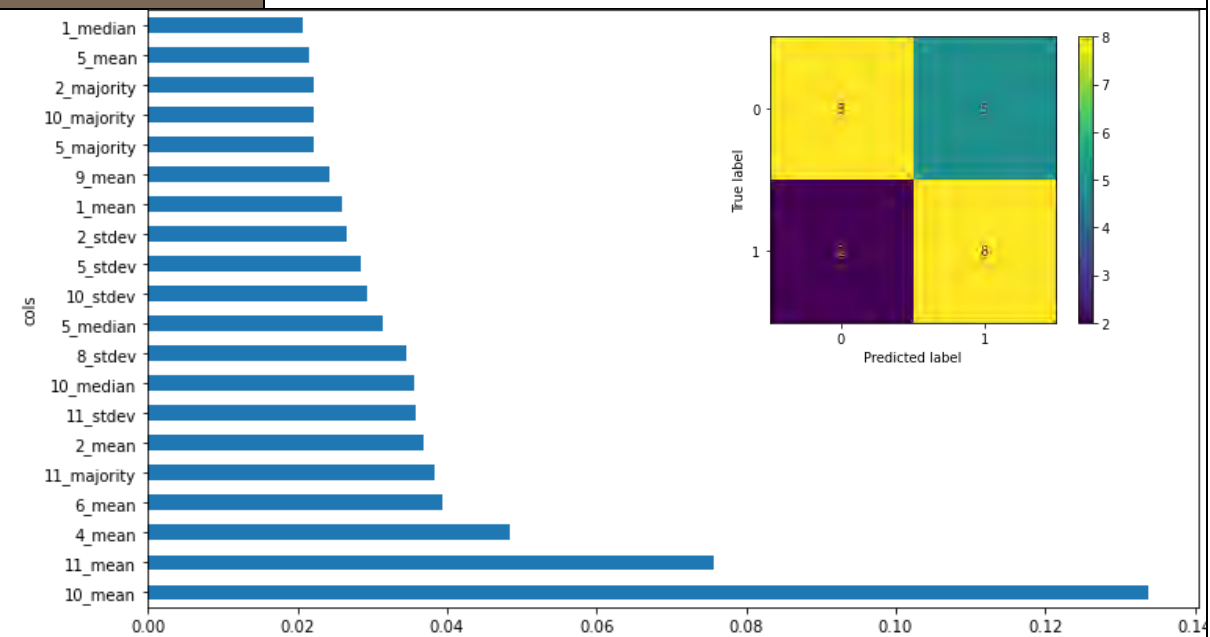
Leyenda imágenes de ejemplo



Ventana clasificador

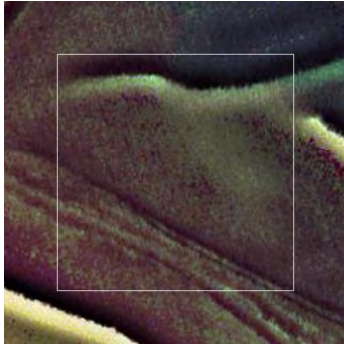


Yacimiento IAN

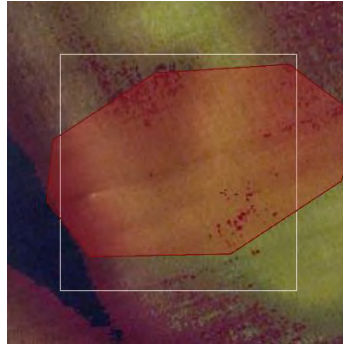


Ventana	G24	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 200}				
Rango (m ²)	4018 - 6577	Rendimiento (GM)	75.58	ROC AUC	84.55
Model AUC PR	83.16	F1 score	75.0	Índice de Kappa	0.51
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

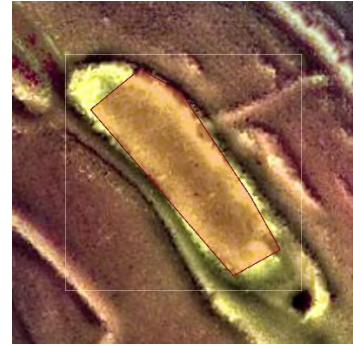
135



34



101



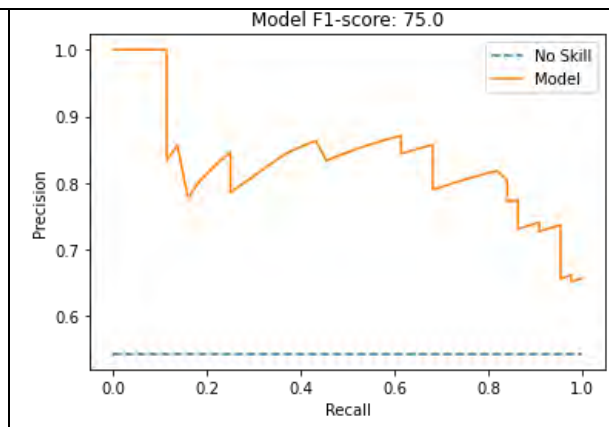
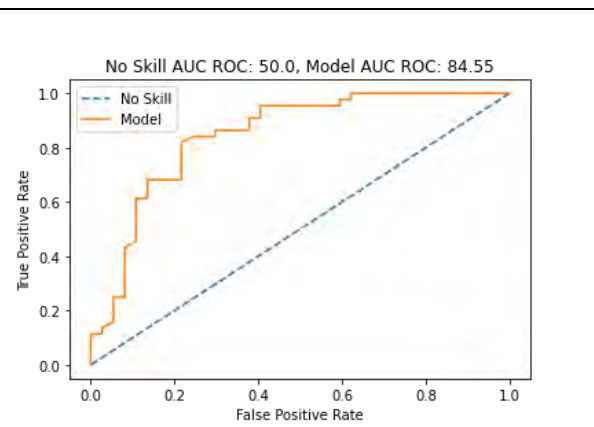
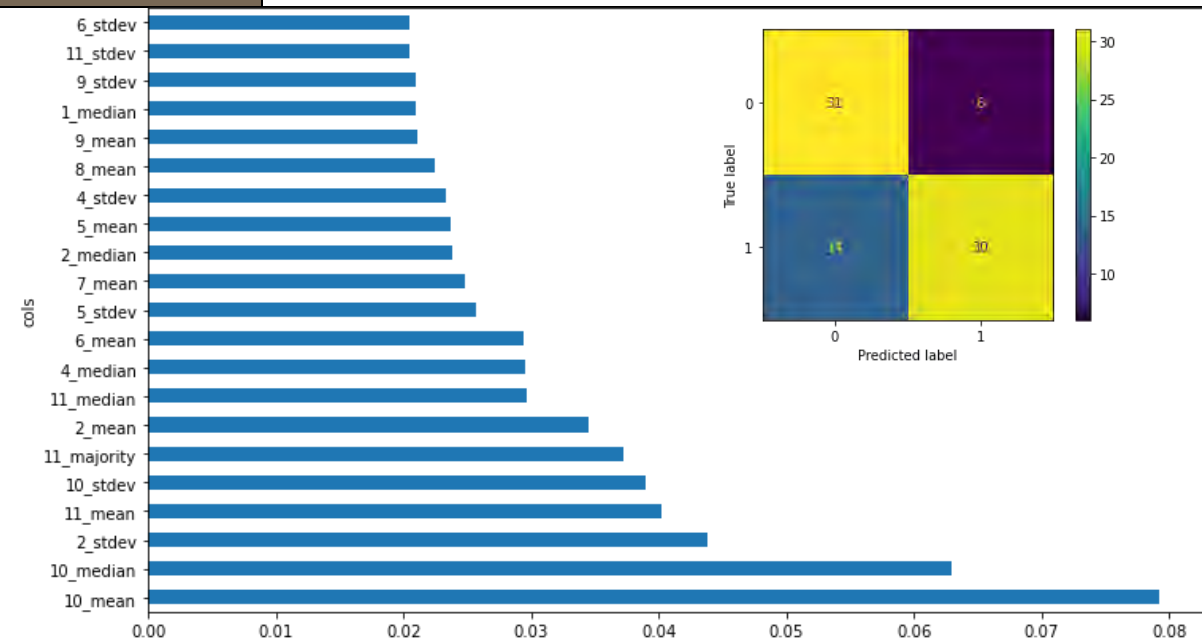
Leyenda imágenes de ejemplo



Ventana clasificador

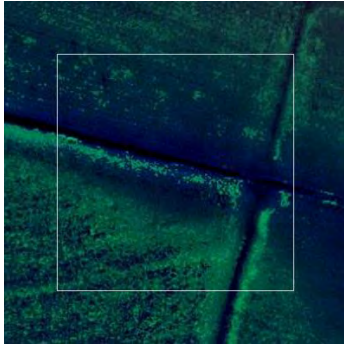


Yacimiento IAN

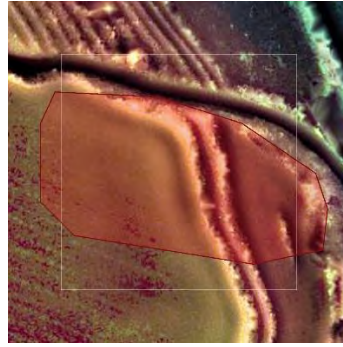


Ventana	G25	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	'{criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 200}				
Rango (m ²)	8537 - 9725	Rendimiento (GM)	87.71	ROC AUC	93.36
Model AUC PR	90.42	F1 score	88.0	Índice de Kappa	0.75
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

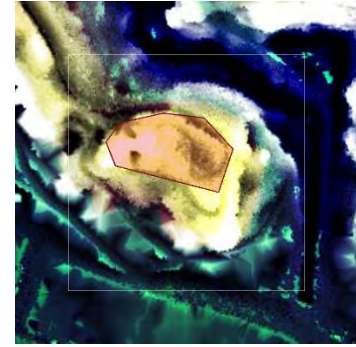
40



10



30



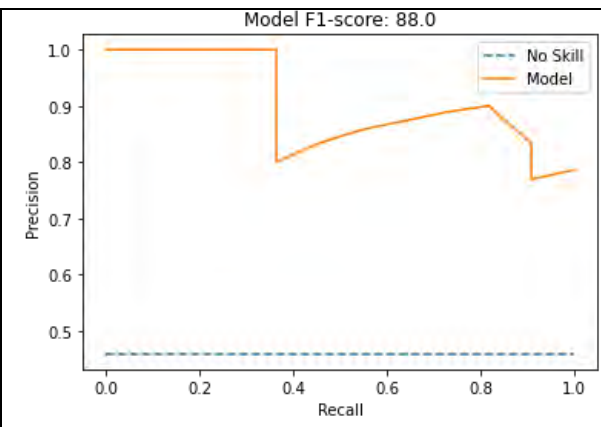
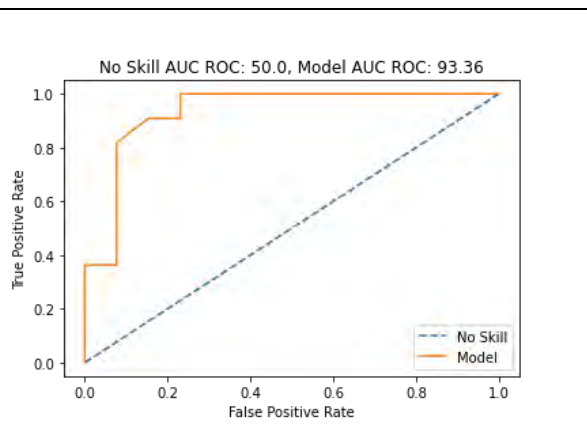
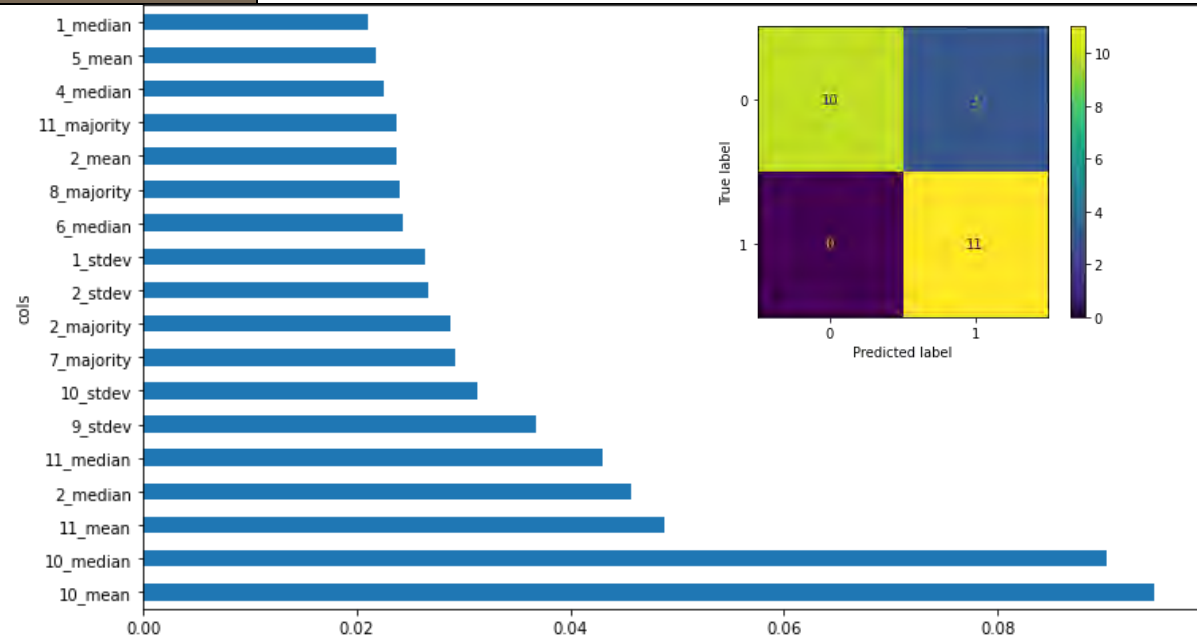
Leyenda imágenes de ejemplo



Ventana clasificador

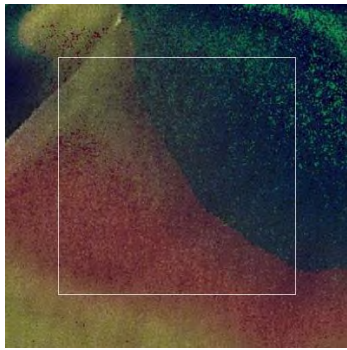


Yacimiento IAN

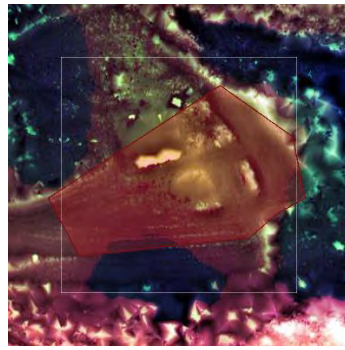


Ventana	G26	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': None, 'n_estimators': 50 }				
Rango (m ²)	15983 - 16875	Rendimiento (GM)	75.21	ROC AUC	88.38
Model AUC PR	88.14	F1 score	73.68	Índice de Kappa	0.5
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

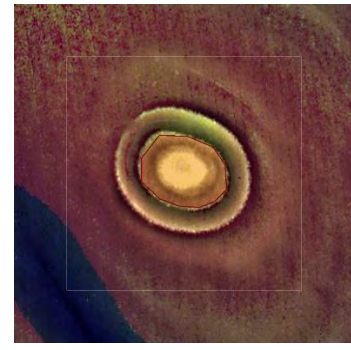
33



8



25



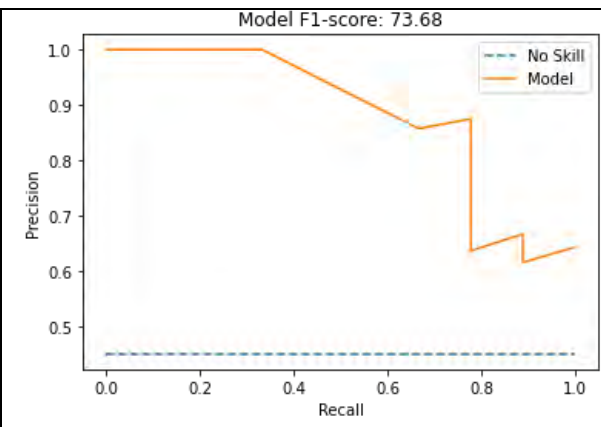
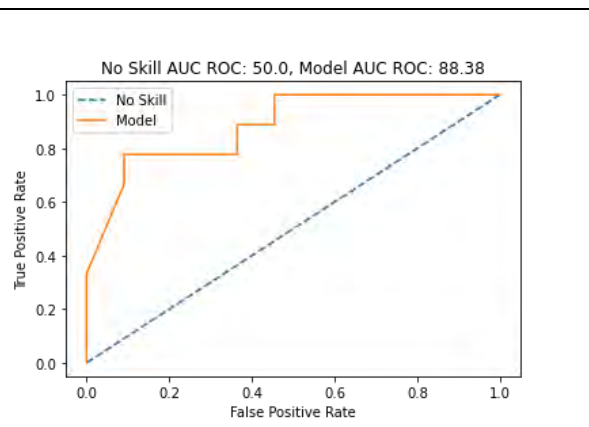
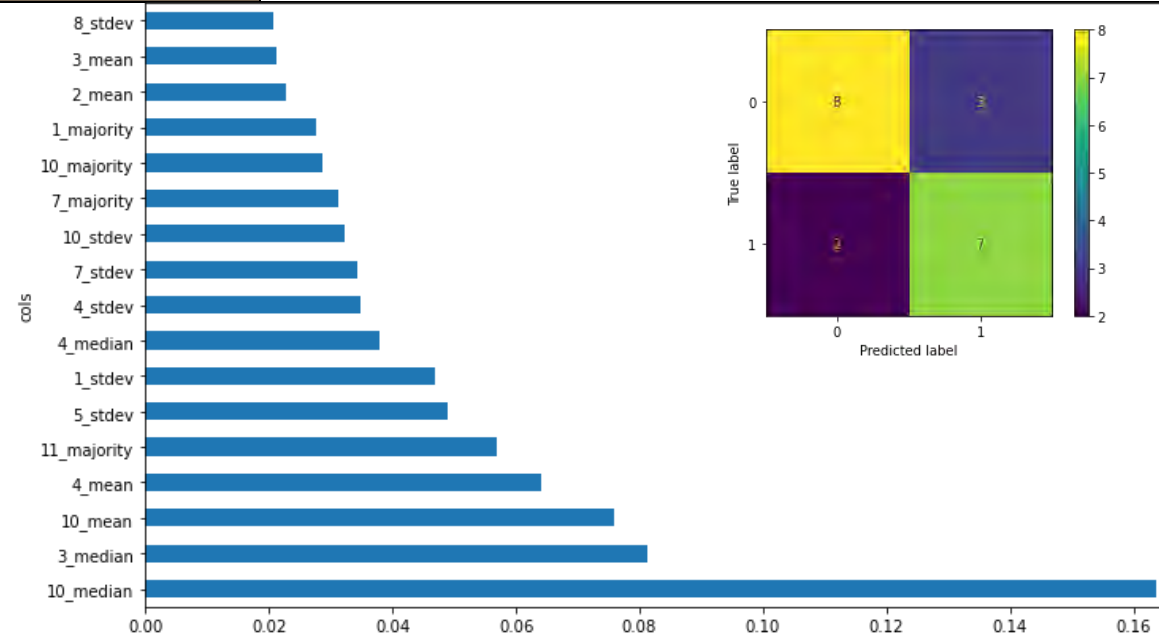
Leyenda imágenes de ejemplo



Ventana clasificador

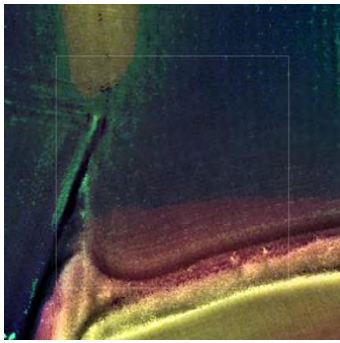


Yacimiento IAN

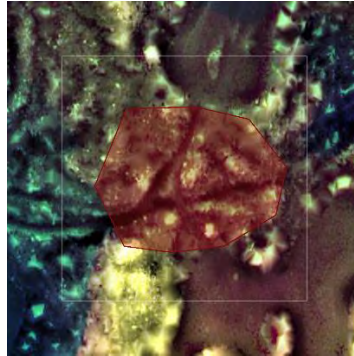


Ventana	G27	Tipo de clasificador	Débil		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 50 }				
Rango (m ²)	12971 - 13837	Rendimiento (GM)	68.03	ROC AUC	78.10
Model AUC PR	82.58	F1 score	69.57	Índice de Kappa	36
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

36



9



27



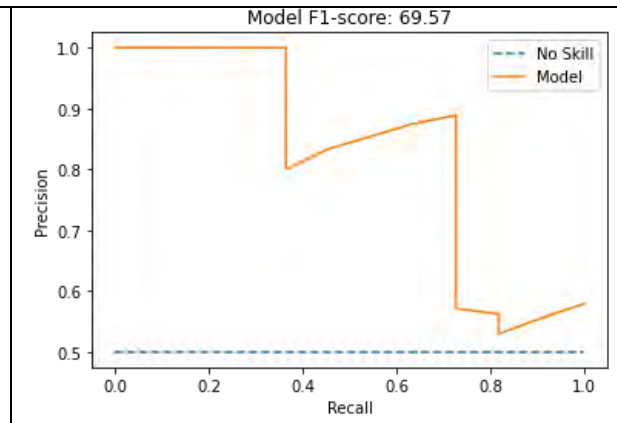
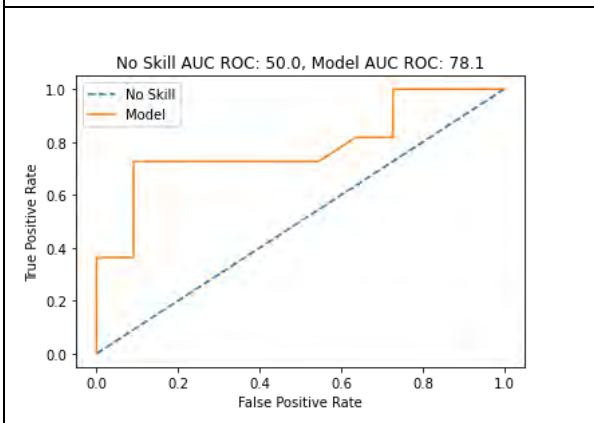
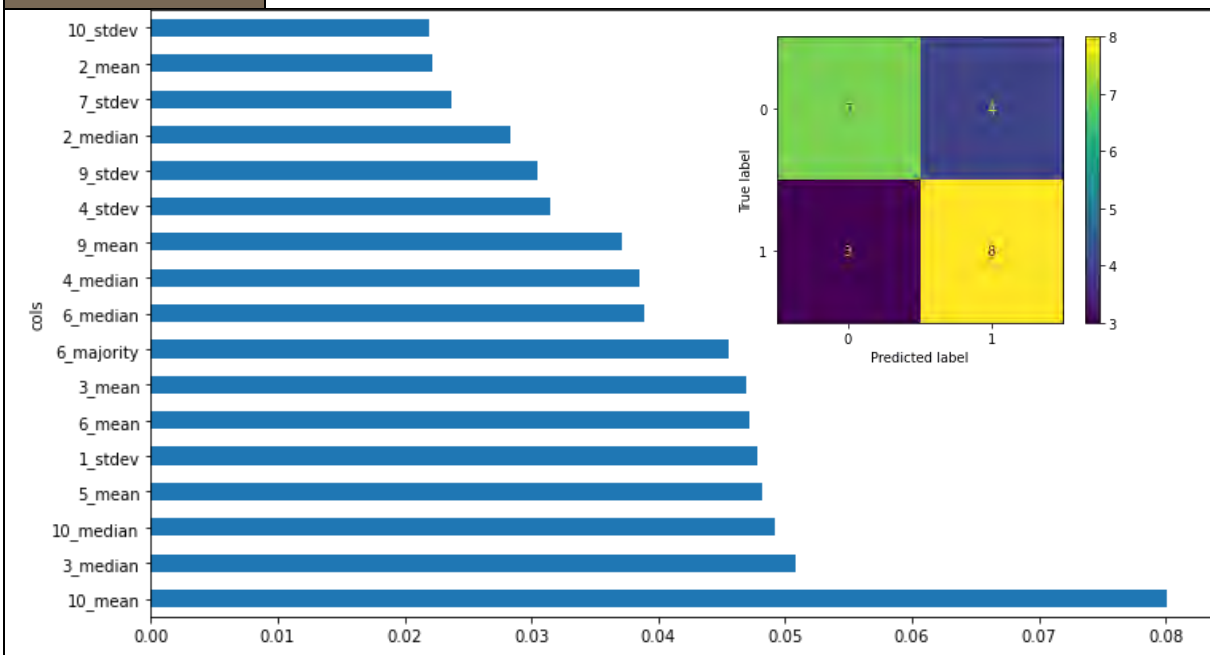
Leyenda imágenes de ejemplo



Ventana clasificador

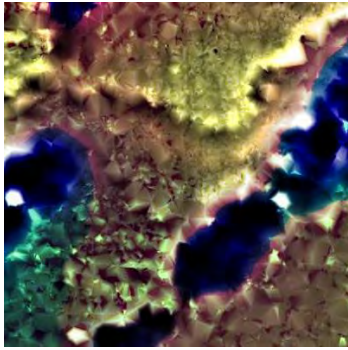


Yacimiento IAN

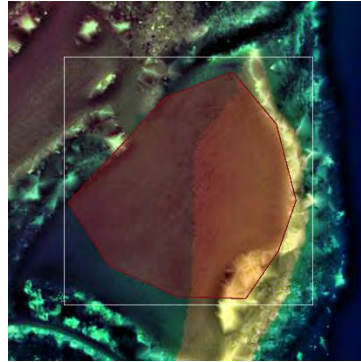


Ventana	G28	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 10 }				
Rango (m ²)	11723 - 12923	Rendimiento (GM)	74.8	ROC AUC	86.01
Model AUC PR	83.0	F1 score	72.73	Índice de Kappa	0.5
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

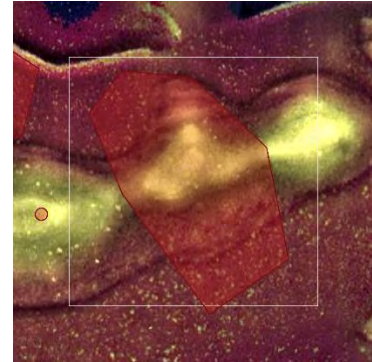
40



10



30



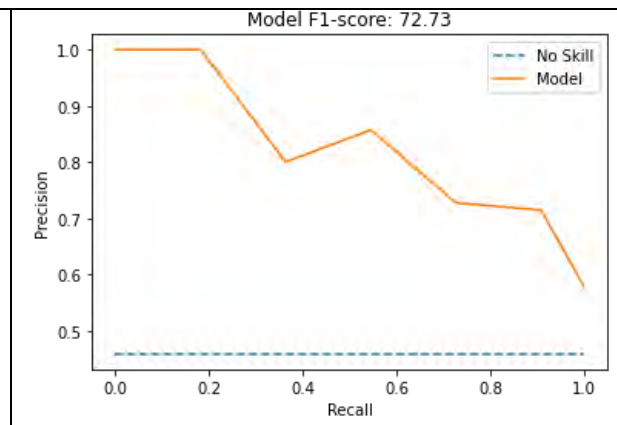
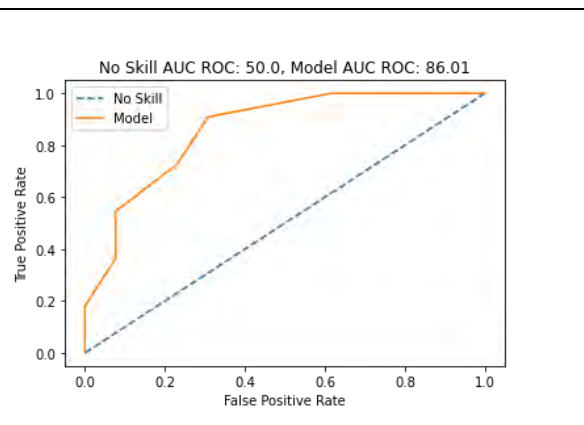
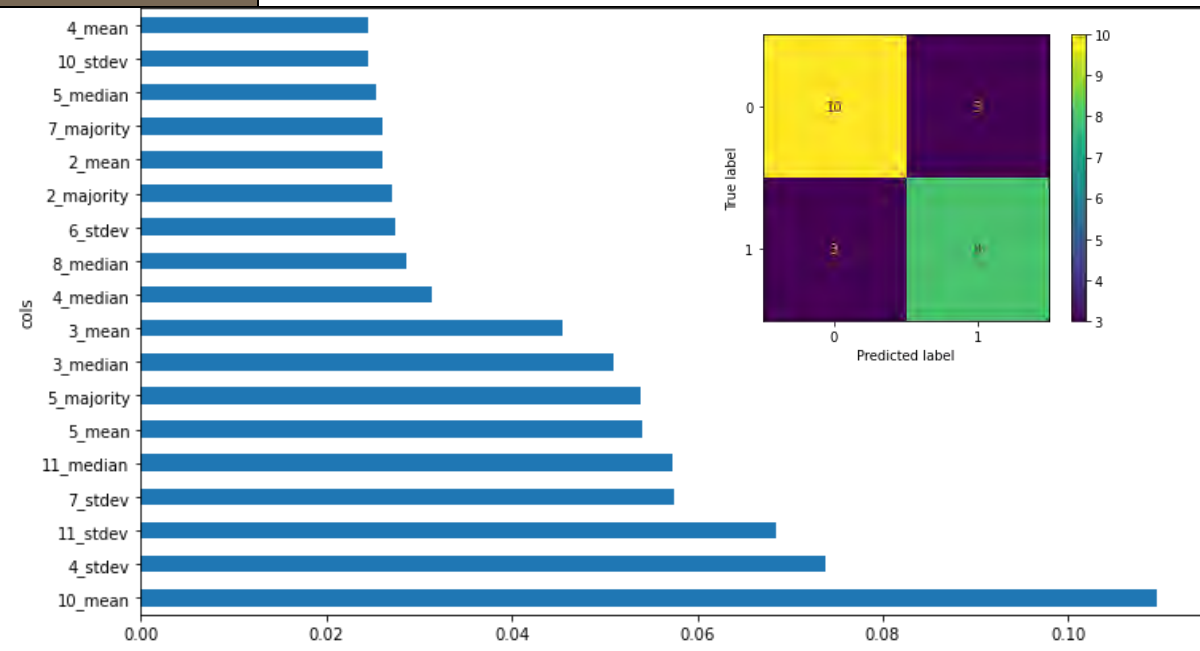
Leyenda imágenes de ejemplo



Ventana clasificador

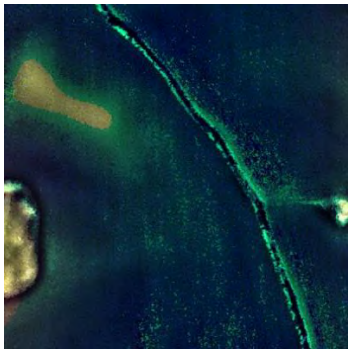


Yacimiento IAN

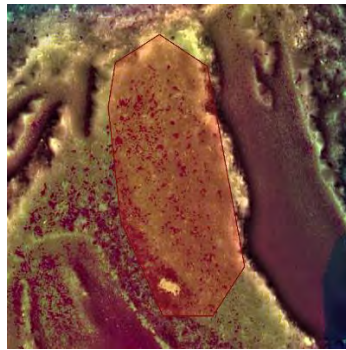


Ventana	G29	Tipo de clasificador	Débil		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': None, 'n_estimators': 50 }				
Rango (m ²)	17051 - 18032	Rendimiento (GM)	60.3	ROC AUC	52.02
Model AUC PR	67.26	F1 score	53.33	Índice de Kappa	0.27
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

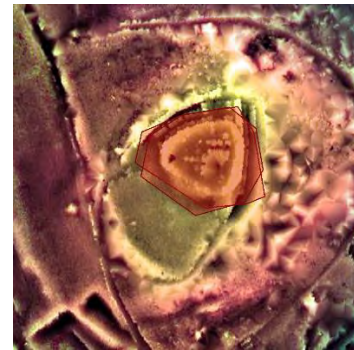
33



8



25



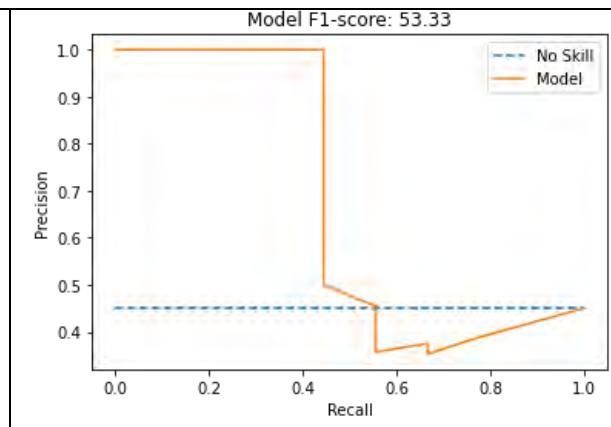
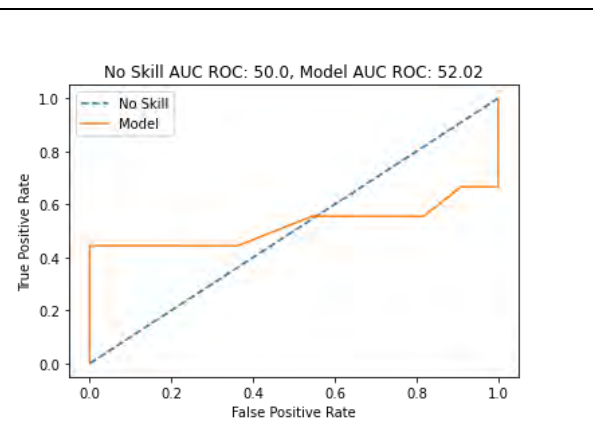
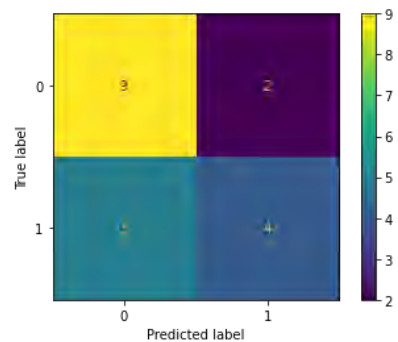
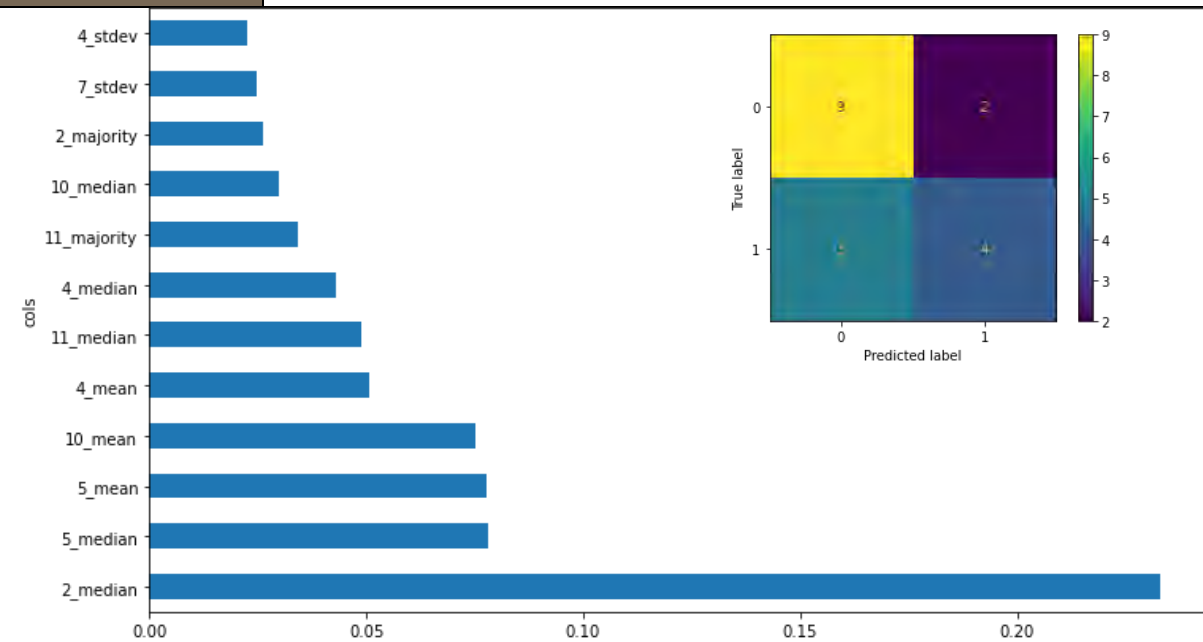
Leyenda imágenes de ejemplo



Ventana clasificador

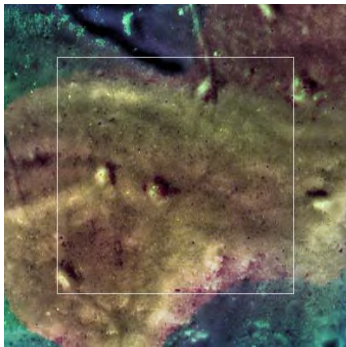


Yacimiento IAN

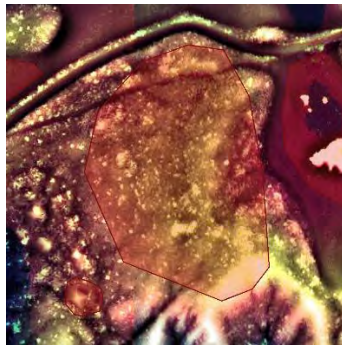


Ventana	G30	Tipo de clasificador	Moderado		
Método	LogisticRegression(multi_class='ovr', n_jobs=-1, random_state=48)				
Mejor configuración	{ 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear' }				
Rango (m ²)	18105 - 19231	Rendimiento (GM)	67.42	ROC AUC	70.00
Model AUC PR	57.27	F1 score	76.92	Índice de Kappa	0.44
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

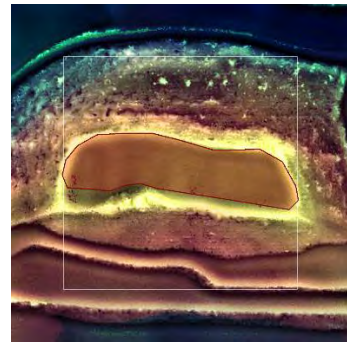
35



9



26



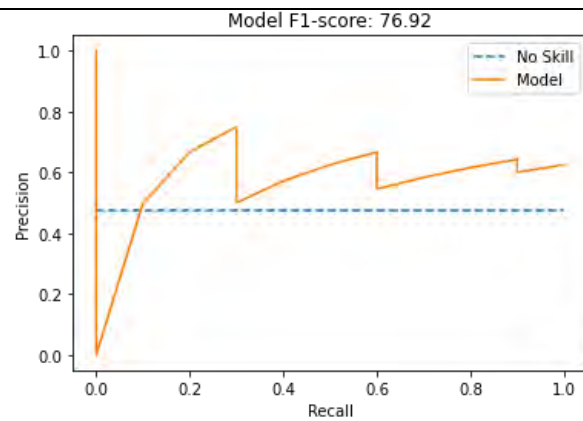
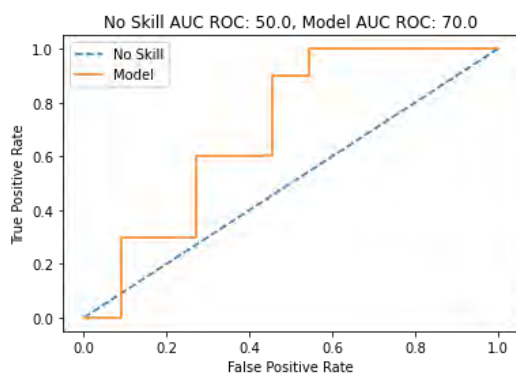
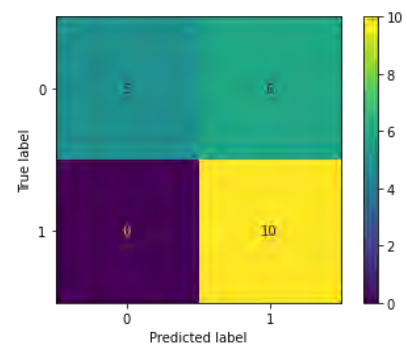
Leyenda imágenes de ejemplo



Ventana clasificador

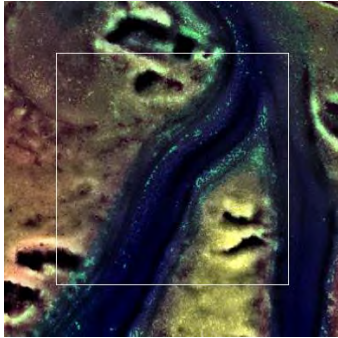


Yacimiento IAN

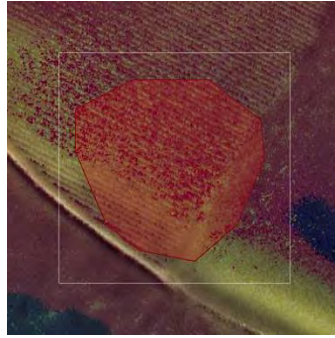


Ventana	G31	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': None, 'n_estimators': 200 }				
Rango (m ²)	13988 - 15837	Rendimiento (GM)	75.69	ROC AUC	80.56
Model AUC PR	77.65	F1 score	73.33	Índice de Kappa	0.52
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

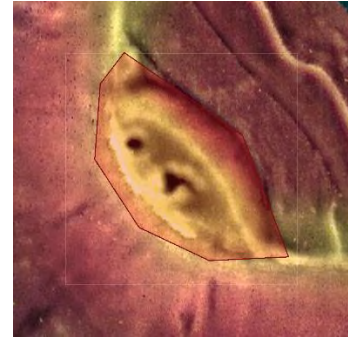
56



14



42



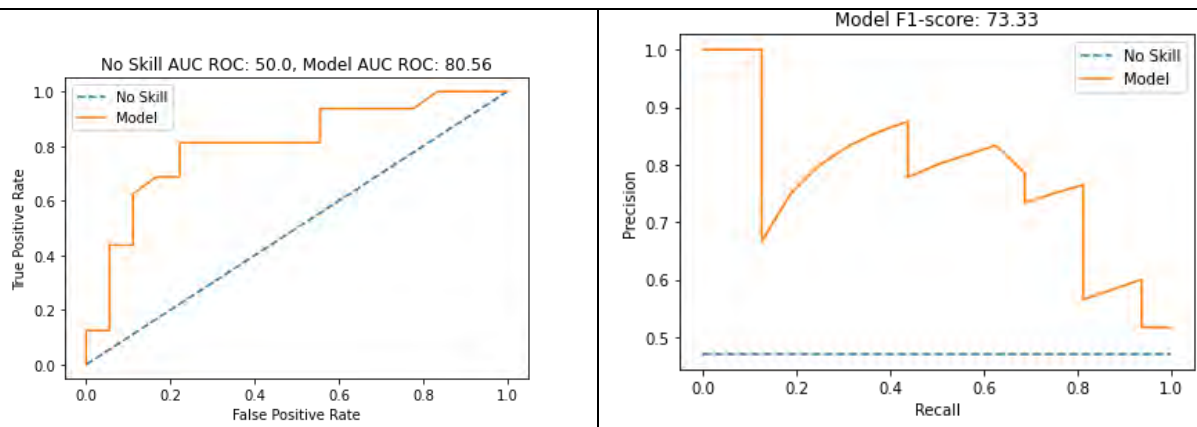
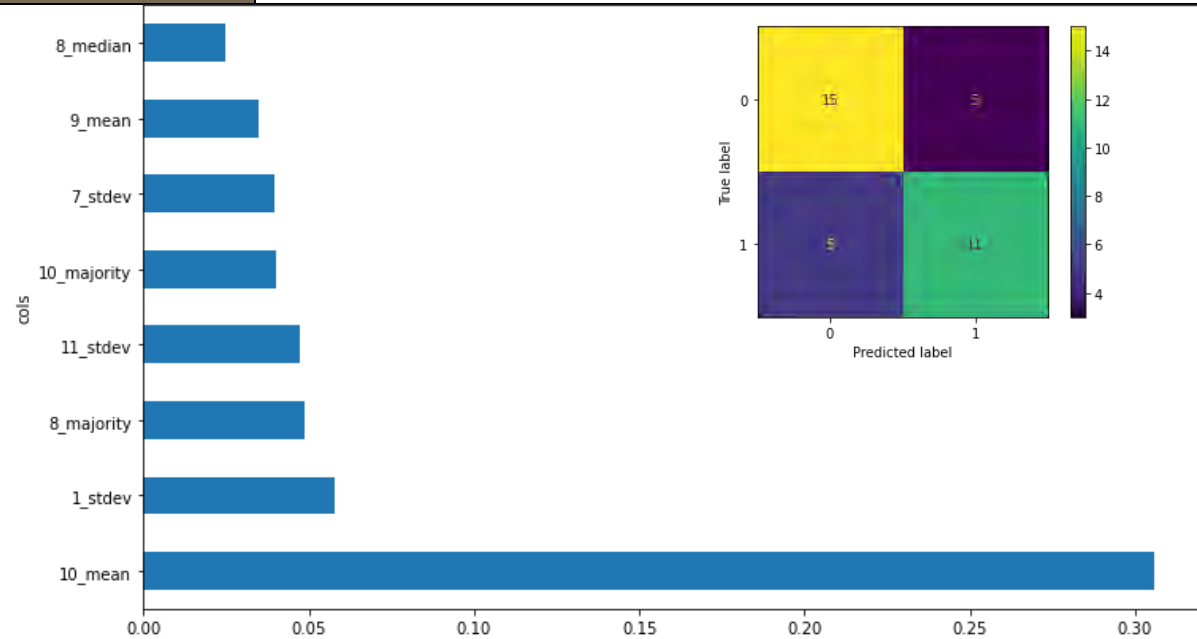
Leyenda imágenes de ejemplo



Ventana clasificador



Yacimiento IAN

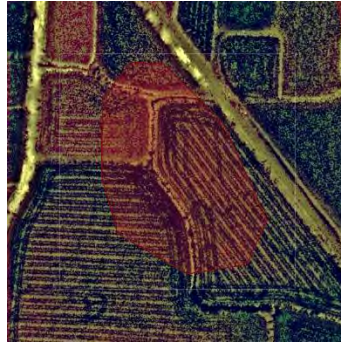


Ventana	G32	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 50 }				
Rango (m ²)	28269 - 30481	Rendimiento (GM)	73.85	ROC AUC	95.87
Model AUC PR	96.7	F1 score	81.48	Índice de Kappa	0.55
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

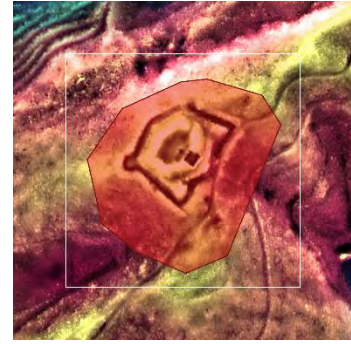
36



9



27



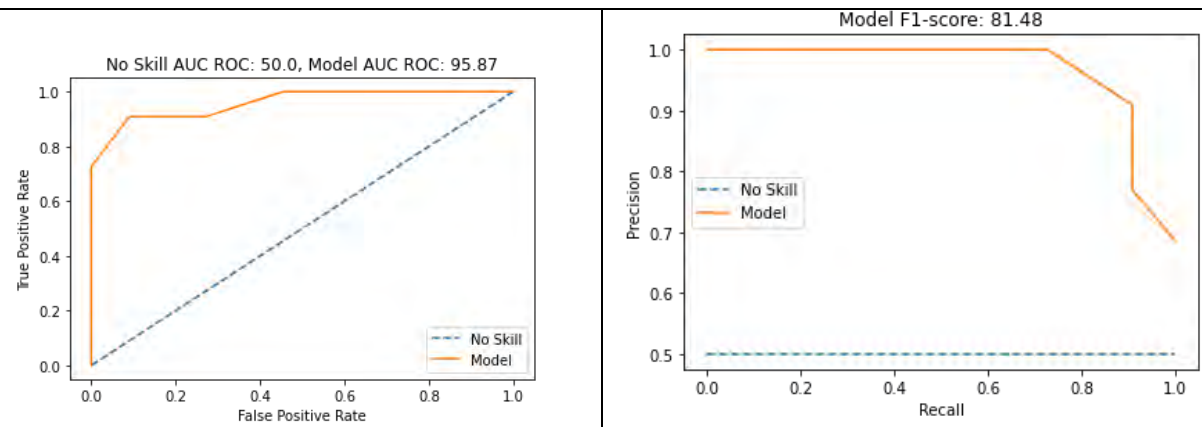
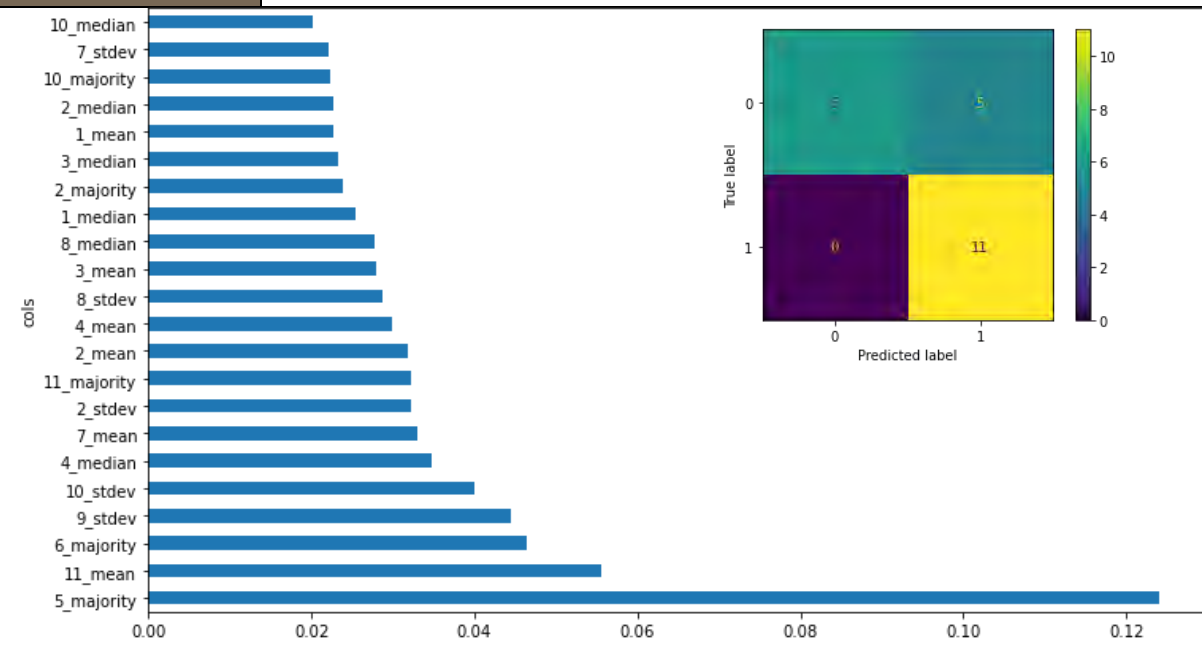
Leyenda imágenes de ejemplo



Ventana clasificador

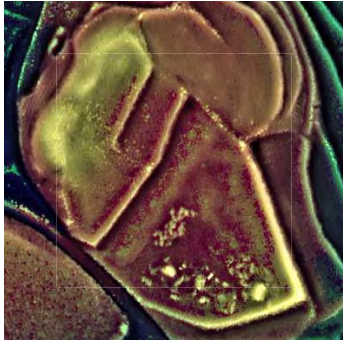


Yacimiento IAN

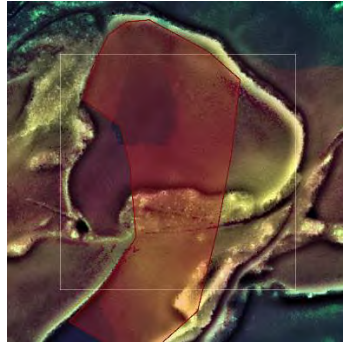


Ventana	G33	Tipo de clasificador	Bueno		
Método	GradientBoostingClassifier(learning_rate=1, max_depth=1, random_state=0)				
Mejor configuración	{ 'max_features': 'sqrt', 'n_estimators': 100 }				
Rango (m ²)	22263 - 24333	Rendimiento (GM)	80.92	ROC AUC	85.71
Model AUC PR	76.87	F1 score	80.0	Índice de Kappa	0.62
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

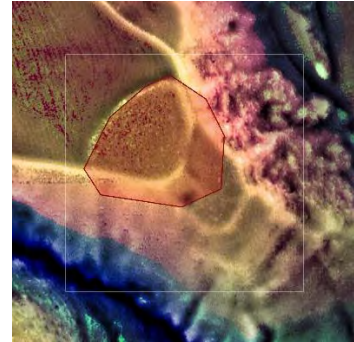
43



11



32



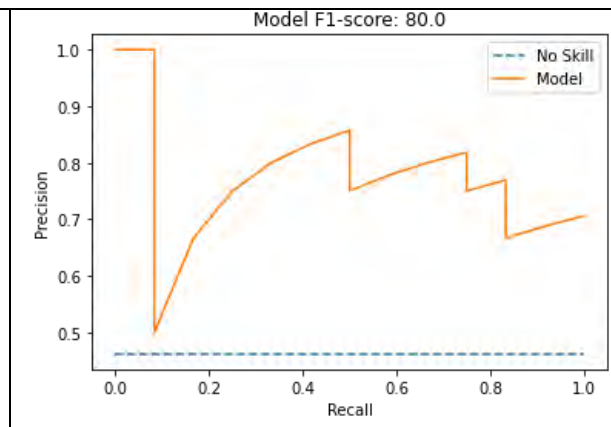
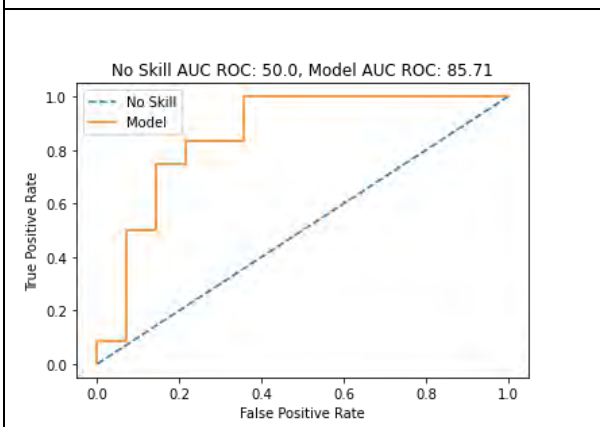
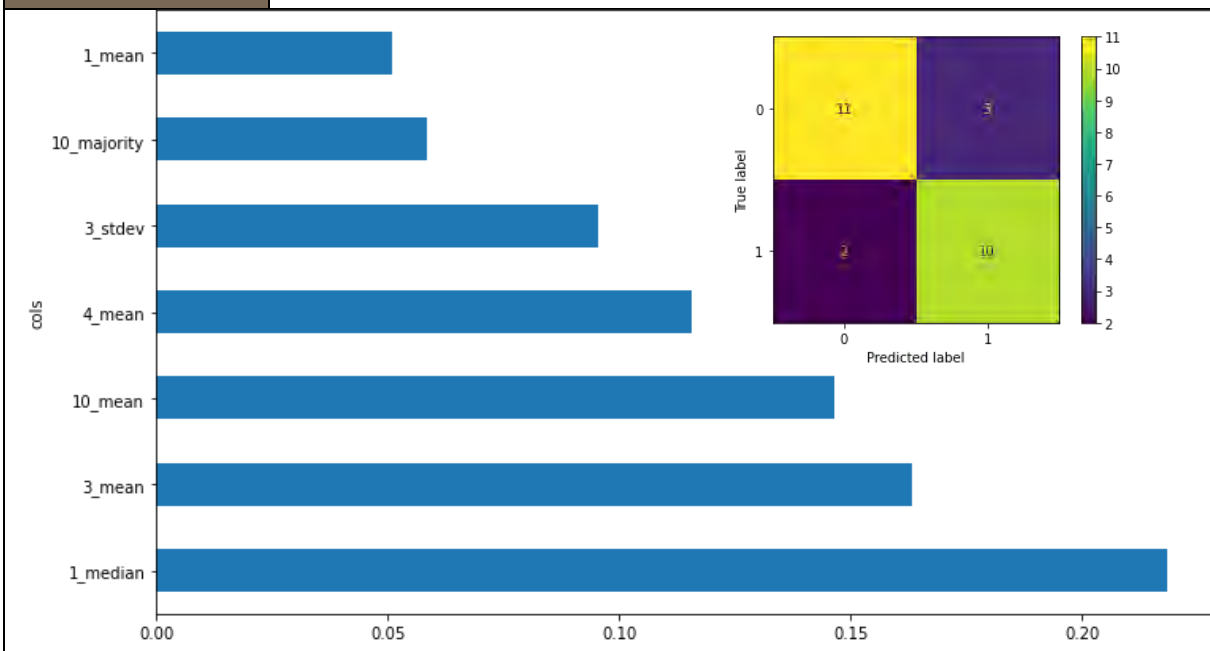
Leyenda imágenes de ejemplo



Ventana clasificador

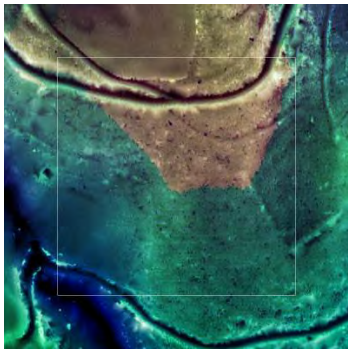


Yacimiento IAN

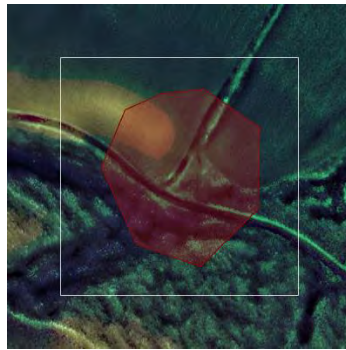


Ventana	G34	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'gini', 'max_features': None, 'n_estimators': 50 }				
Rango (m ²)	19541 - 22100	Rendimiento (GM)	73.5	ROC AUC	80.58
Model AUC PR	82.33	F1 score	73.33	Índice de Kappa	0.47
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

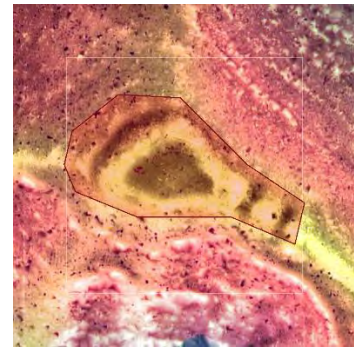
49



12



37



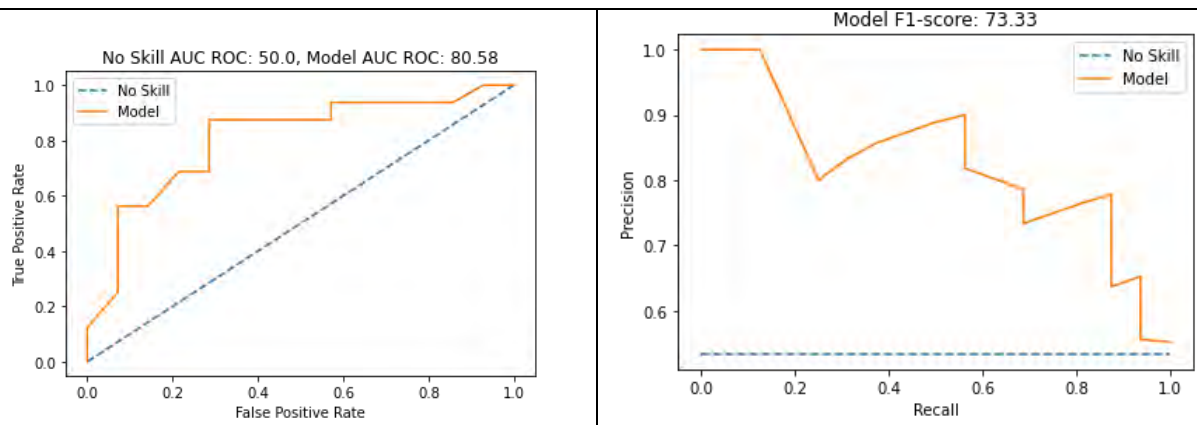
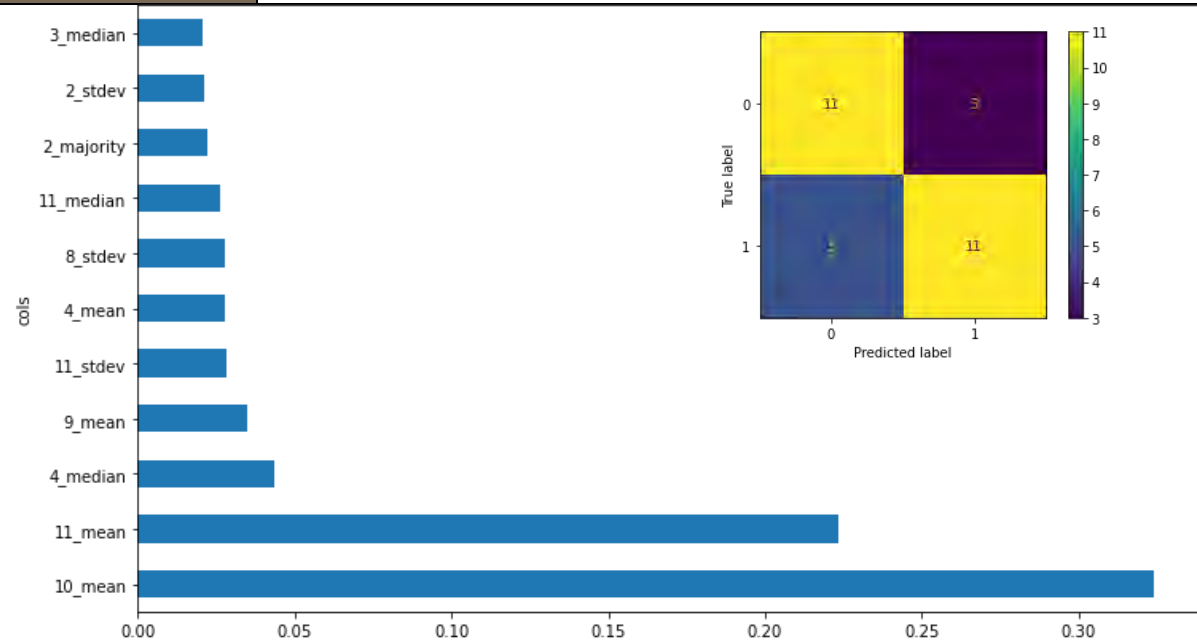
Leyenda imágenes de ejemplo



Ventana clasificador

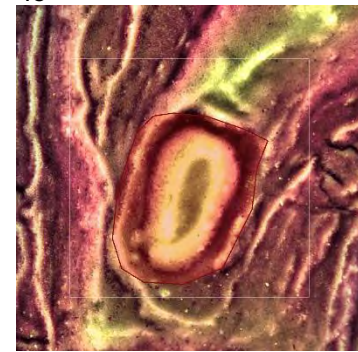
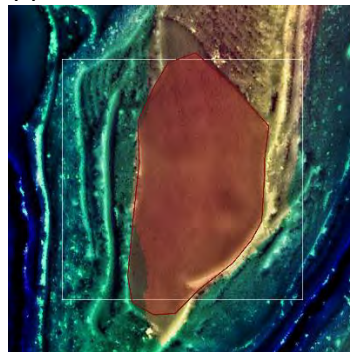
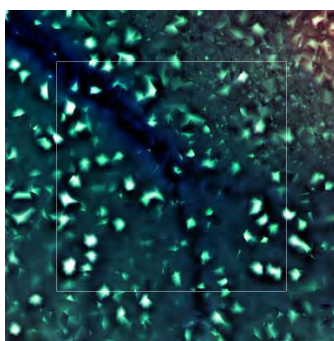


Yacimiento IAN

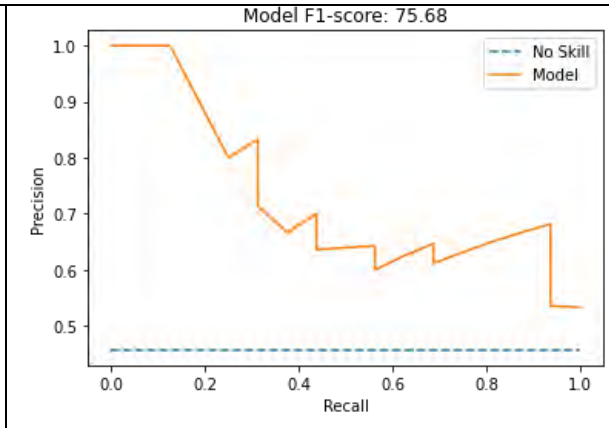
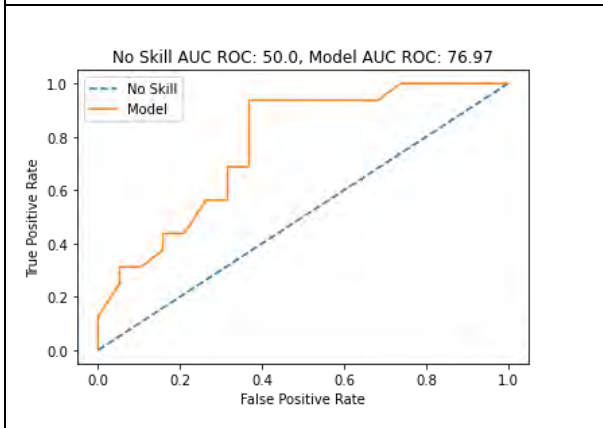
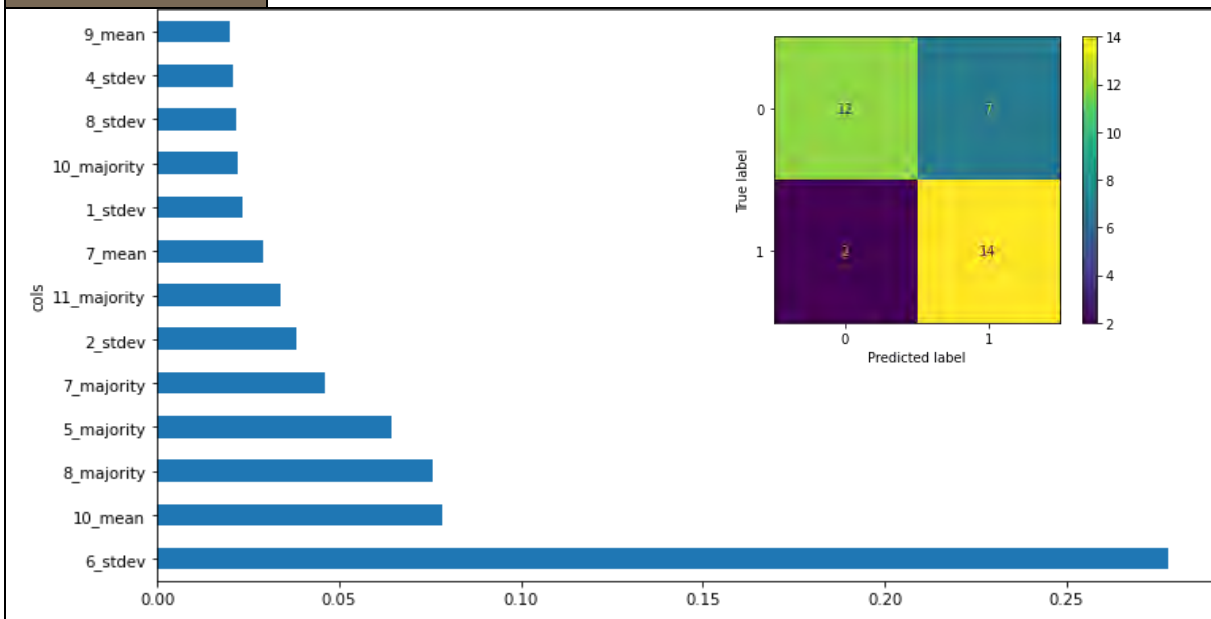


Ventana	G35	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': None, 'n_estimators': 50 }				
Rango (m ²)	24493 - 27933	Rendimiento (GM)	74.34	ROC AUC	76.97
Model AUC PR	72.8	F1 score	75.68	Índice de Kappa	0.49

Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)
57	14	43

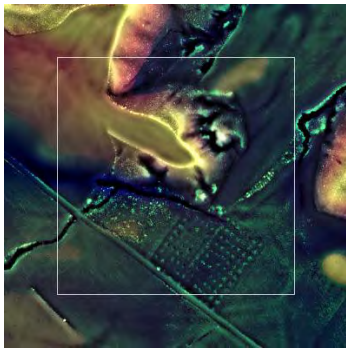


Leyenda imágenes de ejemplo		Ventana clasificador		Yacimiento IAN
-----------------------------	--	----------------------	--	----------------

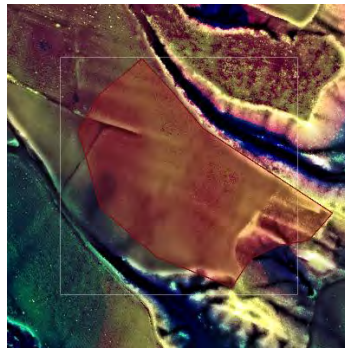


Ventana	G36	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 150 }				
Rango (m ²)	64264 - 72269	Rendimiento (GM)	73.75	ROC AUC	82.14
Model AUC PR	83.62	F1 score	72.0	Índice de Kappa	0.48
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

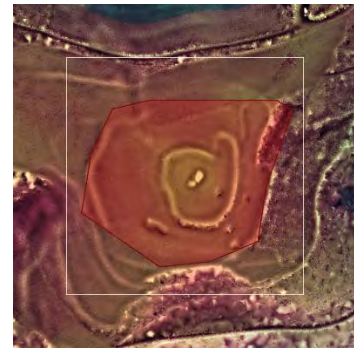
44



11



33



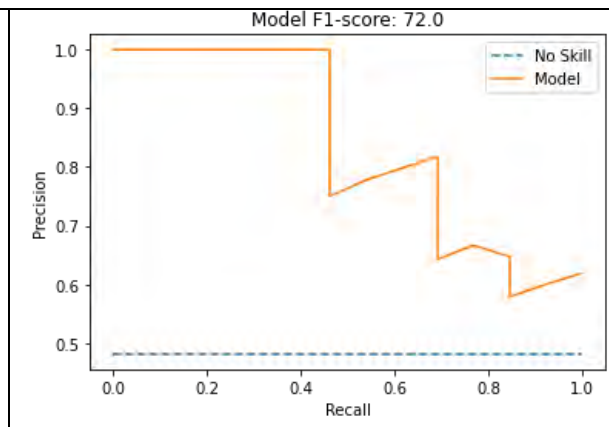
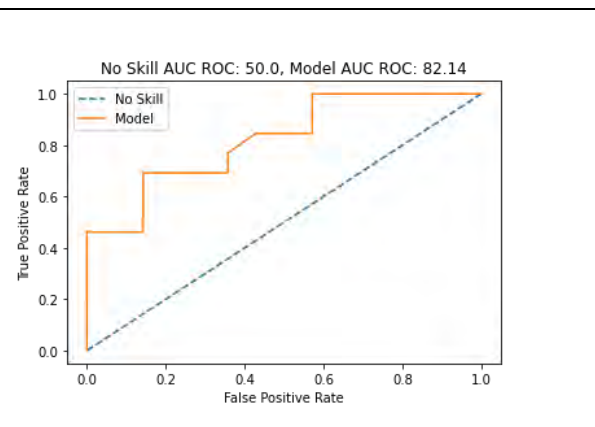
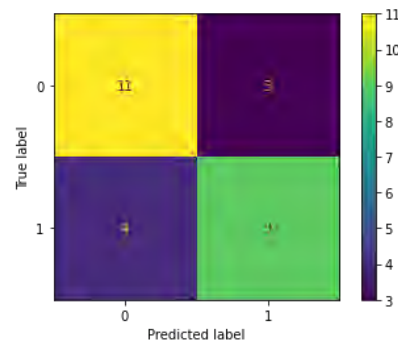
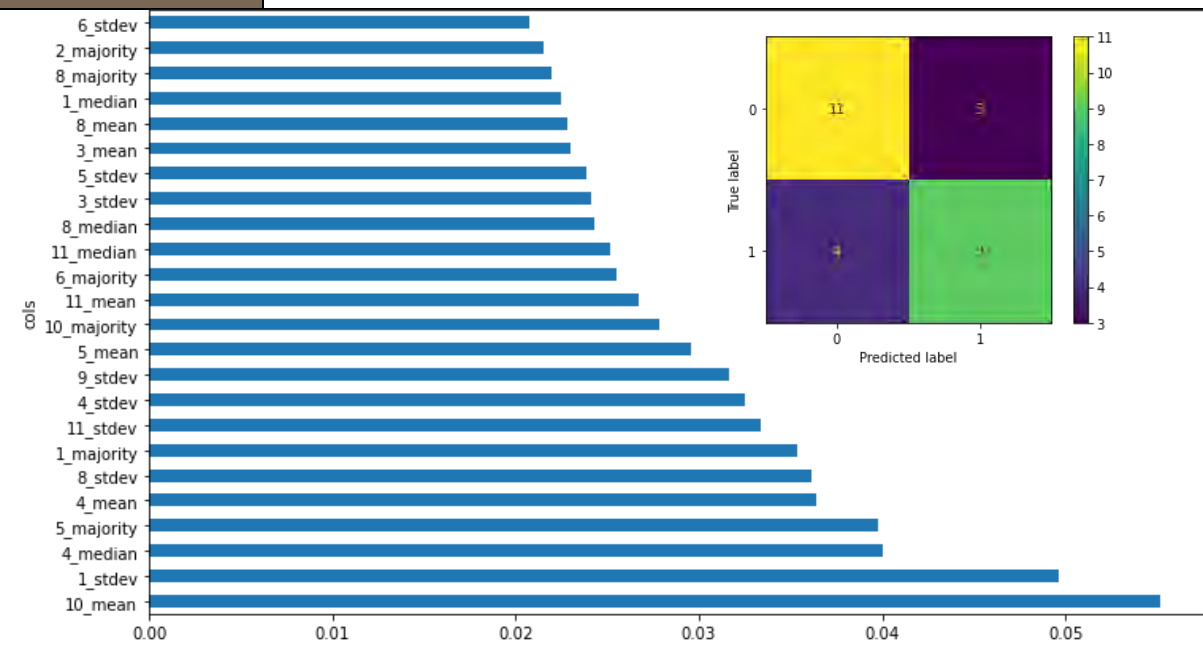
Leyenda imágenes de ejemplo



Ventana clasificador

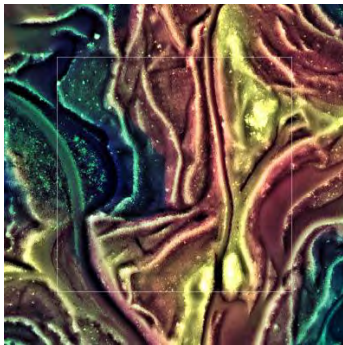


Yacimiento IAN

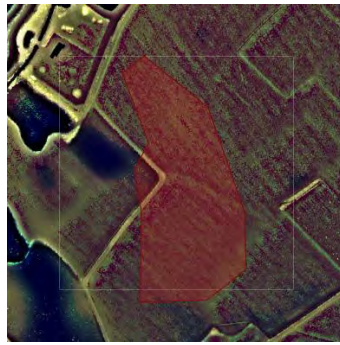


Ventana	G37	Tipo de clasificador	Débil		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 150 }				
Rango (m ²)	46772 - 51510	Rendimiento (GM)	64.78	ROC AUC	91.26
Model AUC PR	92.89	F1 score	71.43	Índice de Kappa	0.36
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

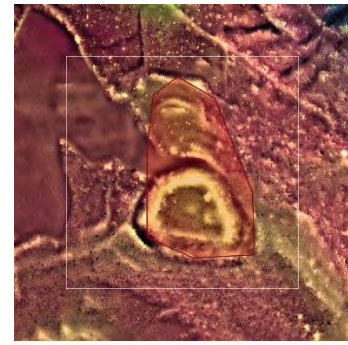
40



10



30



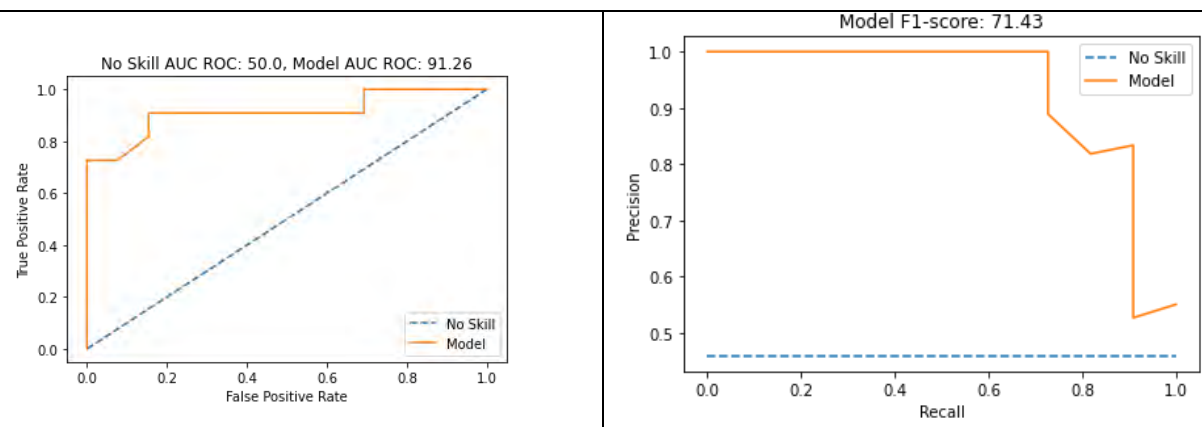
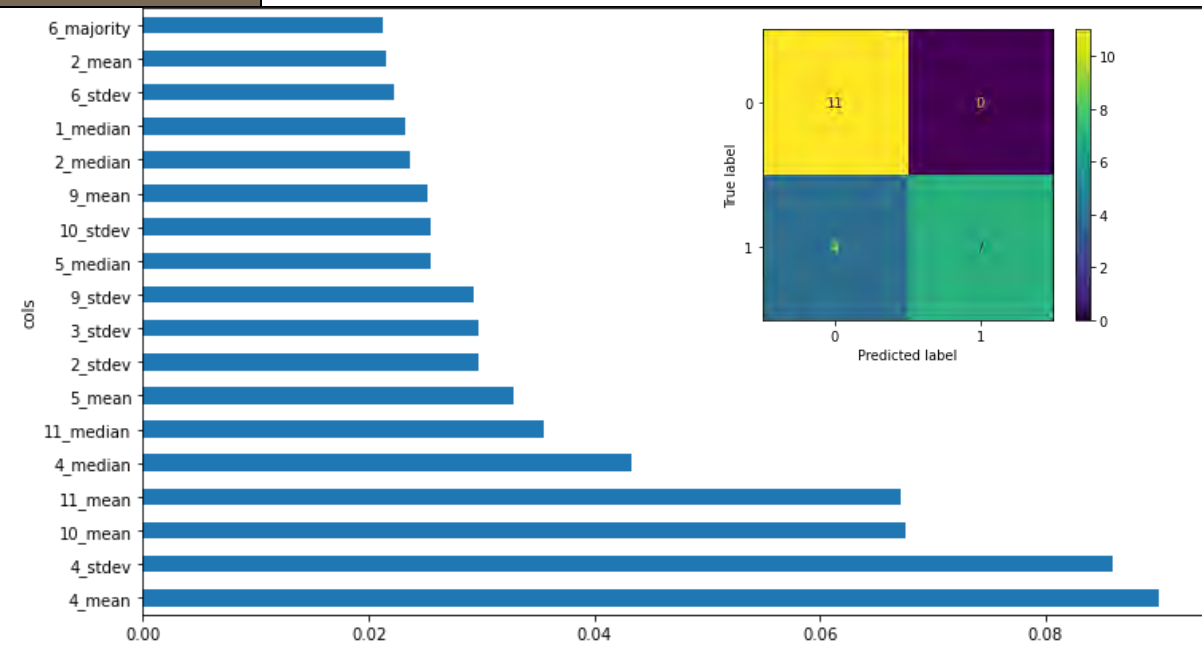
Leyenda imágenes de ejemplo



Ventana clasificador

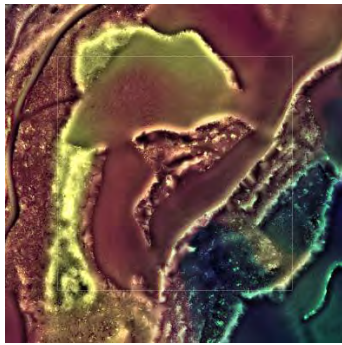


Yacimiento IAN



Ventana	G38	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 100 }				
Rango (m ²)	41324 - 46543	Rendimiento (GM)	79.77	ROC AUC	98.35
Model AUC PR	98.54	F1 score	77.78	Índice de Kappa	0.64
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

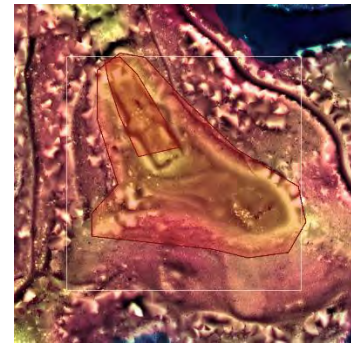
36



9



27



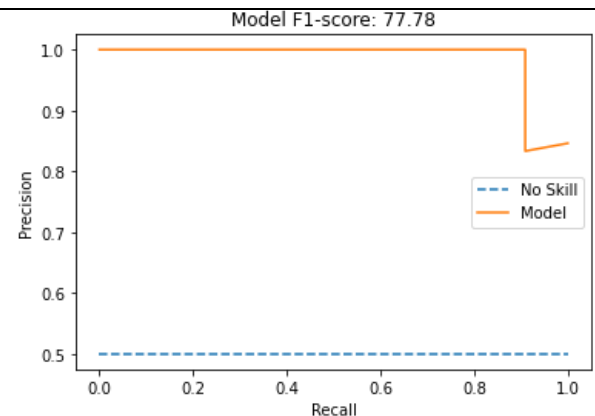
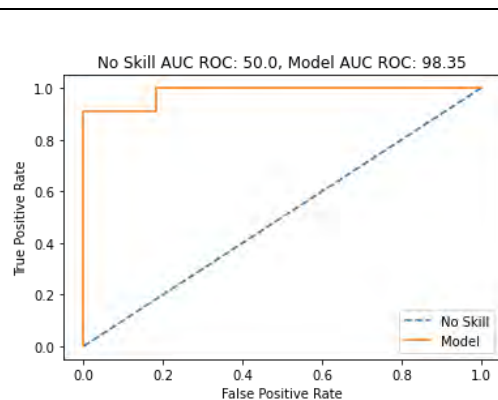
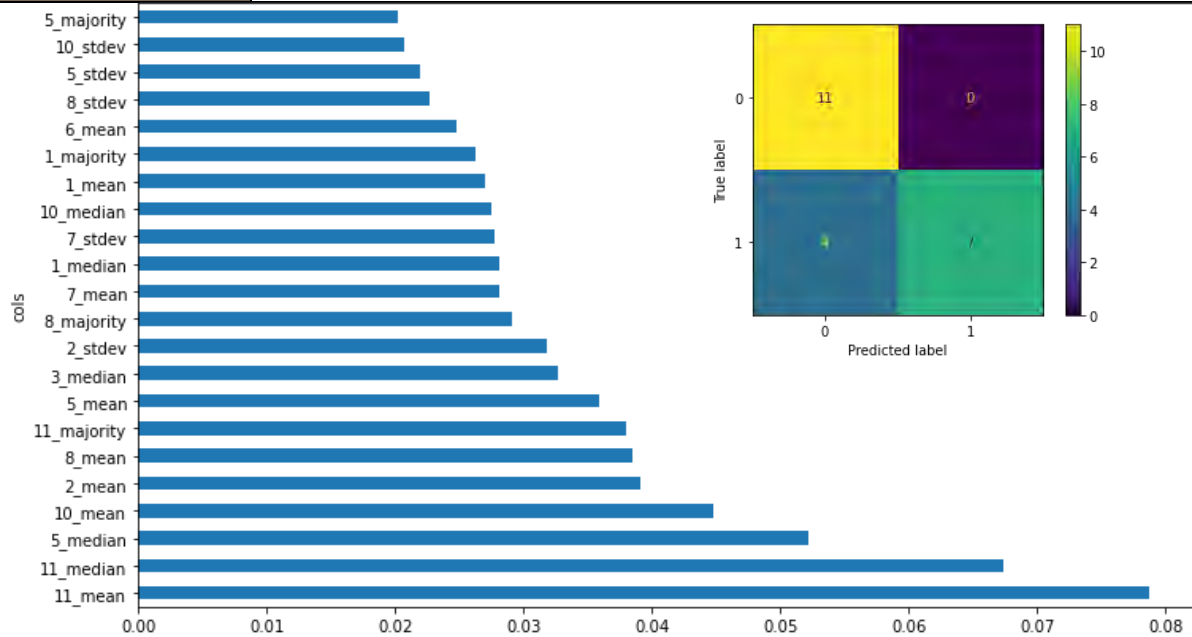
Leyenda imágenes de ejemplo



Ventana clasificador



Yacimiento IAN

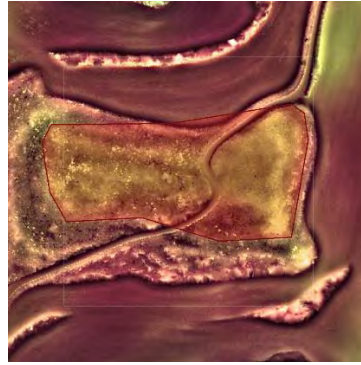


Ventana	G39	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'gini', 'max_features': 'sqrt', 'n_estimators': 50 }				
Rango (m ²)	30684 - 37118	Rendimiento (GM)	74.35	ROC AUC	87.50
Model AUC PR	82.52	F1 score	70.27	Índice de Kappa	0.53
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

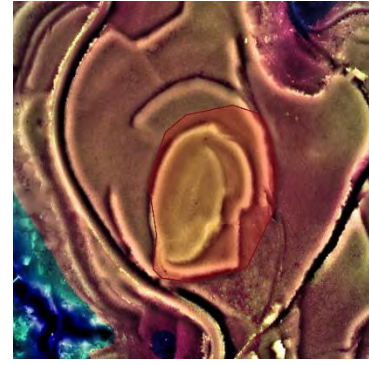
81



20



61



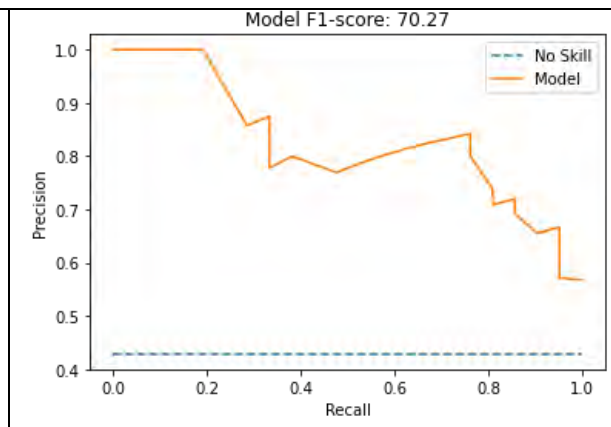
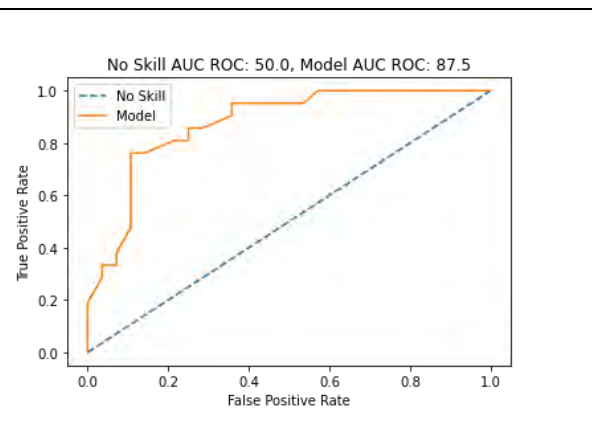
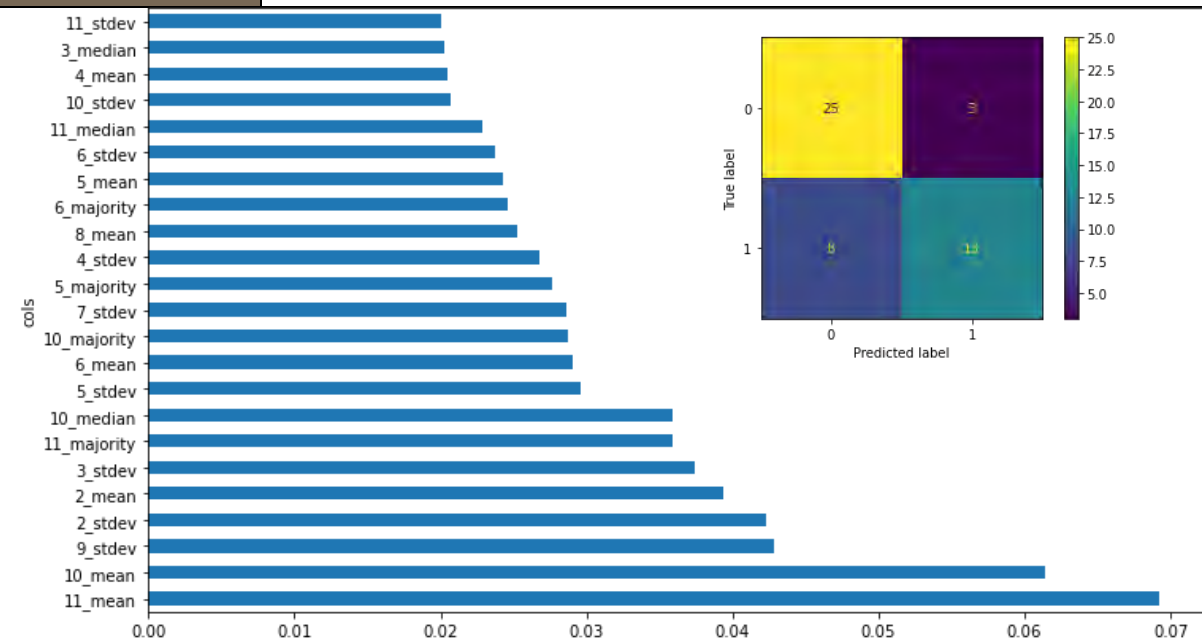
Leyenda imágenes de ejemplo



Ventana clasificador

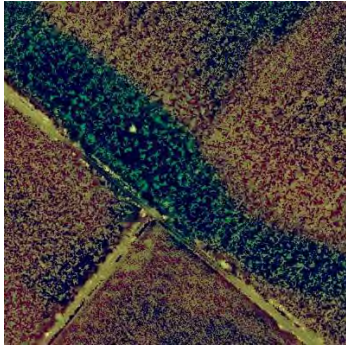


Yacimiento IAN

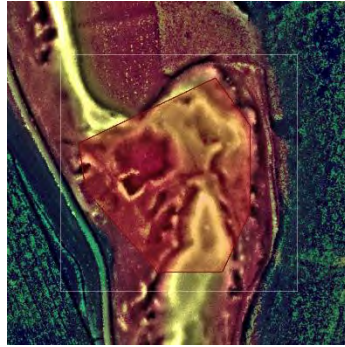


Ventana	G40	Tipo de clasificador	Bueno		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'sqrt', 'n_estimators': 50 }				
Rango (m ²)	37202 - 41197	Rendimiento (GM)	87.29	ROC AUC	84.05
Model AUC PR	93.91	F1 score	85.71	Índice de Kappa	0.75
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

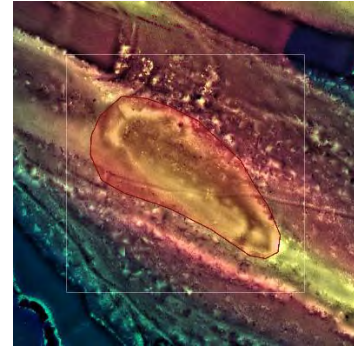
52



13



39



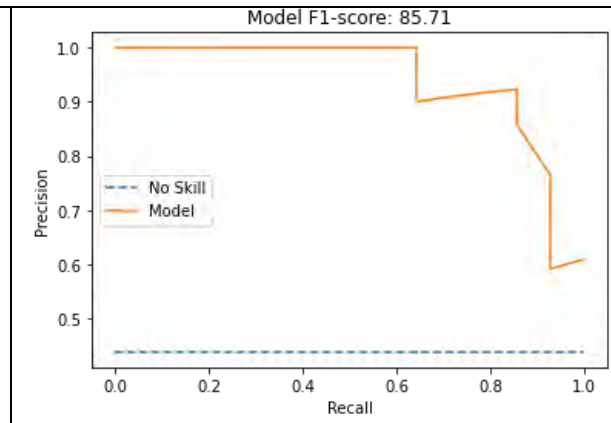
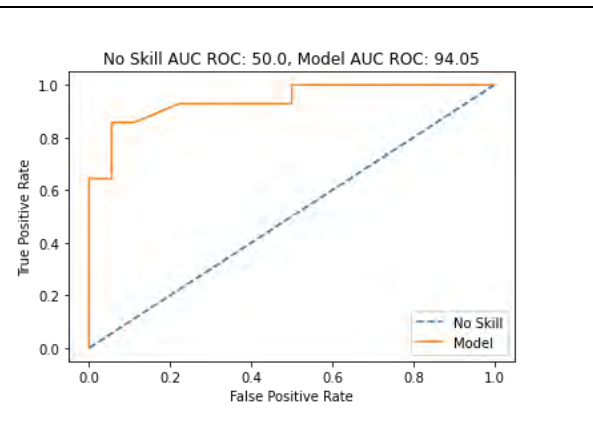
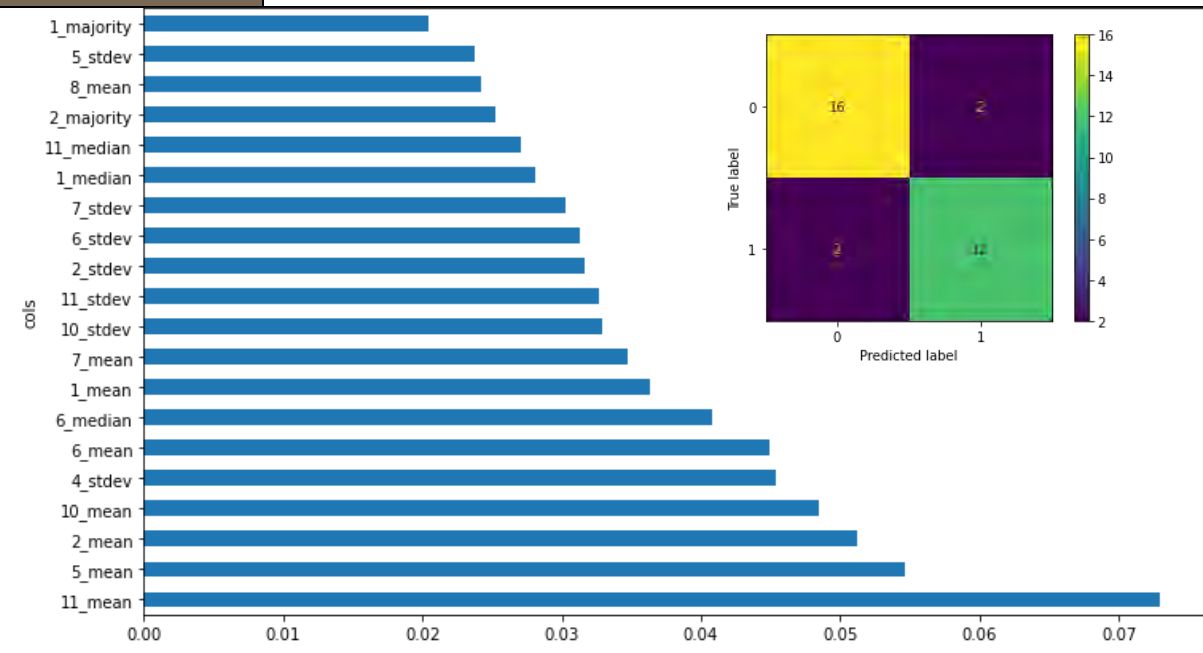
Leyenda imágenes de ejemplo



Ventana clasificador

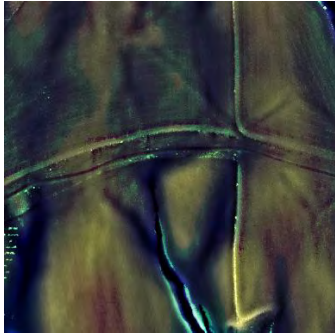


Yacimiento IAN

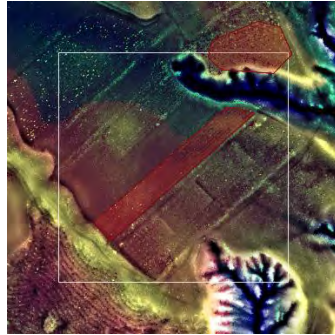


Ventana	G41	Tipo de clasificador	Débil		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': None, 'n_estimators': 50 }				
Rango (m ²)	56448 - 63121	Rendimiento (GM)	66.8	ROC AUC	52.48
Model AUC PR	46.85	F1 score	72.0	Índice de Kappa	0.36
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

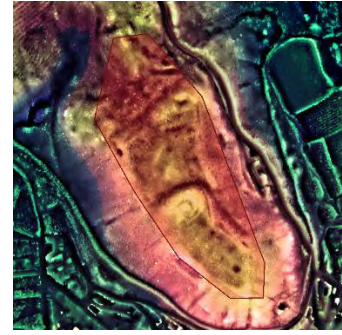
36



9



27



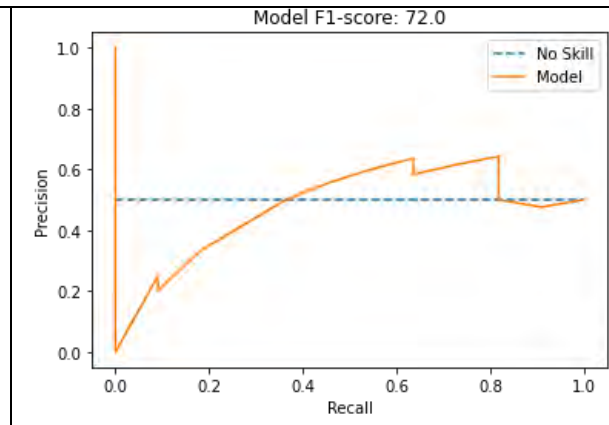
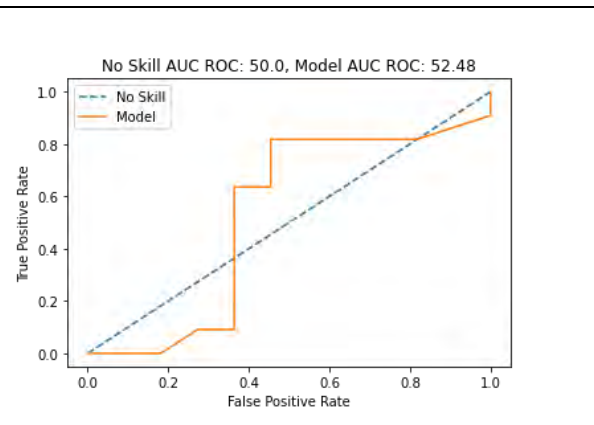
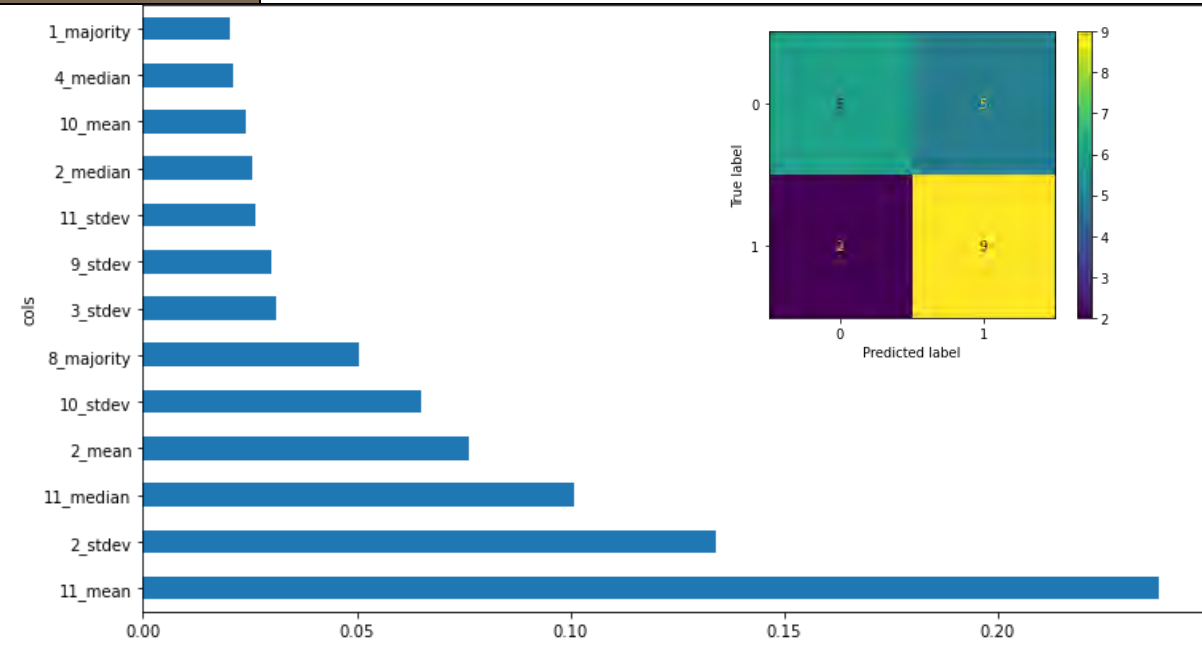
Leyenda imágenes de ejemplo



Ventana clasificador

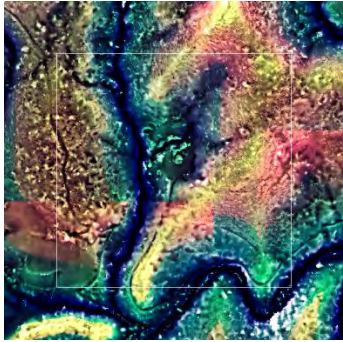


Yacimiento IAN

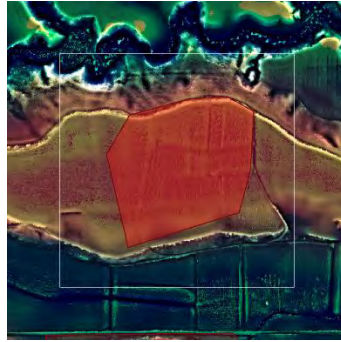


Ventana	G42	Tipo de clasificador	Moderado		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 150 }				
Rango (m ²)	51638 - 19558	Rendimiento (GM)	76.75	ROC AUC	85.31
Model AUC PR	84.19	F1 score	76.32	Índice de Kappa	0.54
Aleatorio (n°)	No topográfico (n°)	Topográfico (n°)			

258



64



194



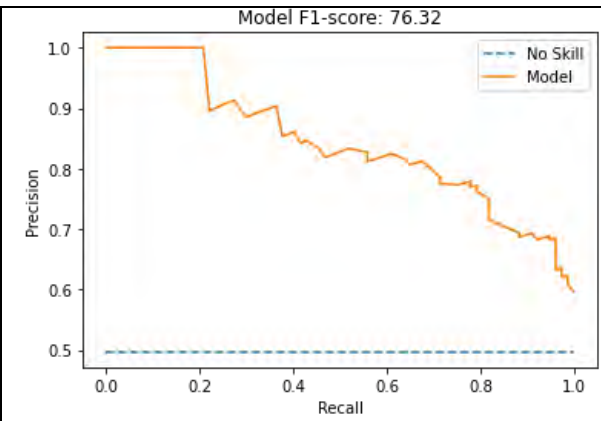
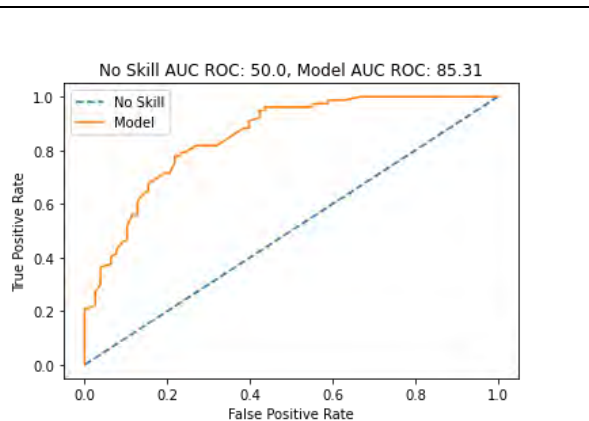
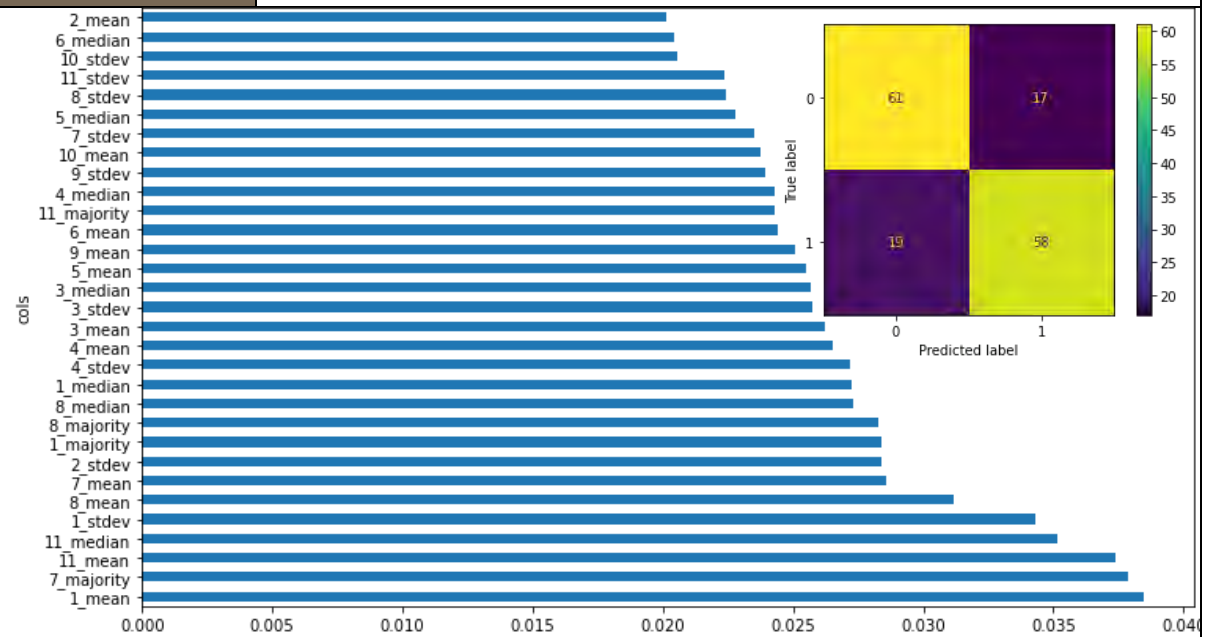
Leyenda imágenes de ejemplo



Ventana clasificador

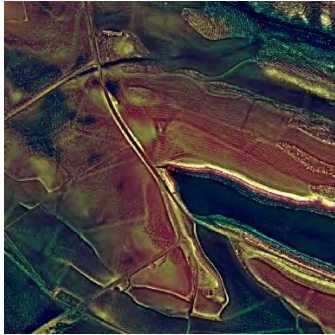


Yacimiento IAN

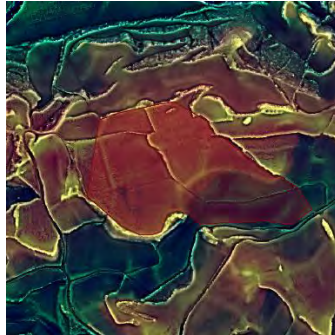


Ventana	G43	Tipo de clasificador	Pobre		
Método	RandomForestClassifier(n_jobs=-1, random_state=98)				
Mejor configuración	{ 'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 10 }				
Rango (m ²)	202448 - 392313	Rendimiento (GM)	52.22	ROC AUC	49.09
Model AUC PR	50.38	F1 score	54.55	Índice de Kappa	0.05
Aleatorio (n°)	No topográfico (n°)		Topográfico (n°)		

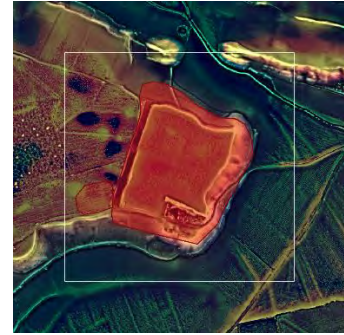
35



9



26



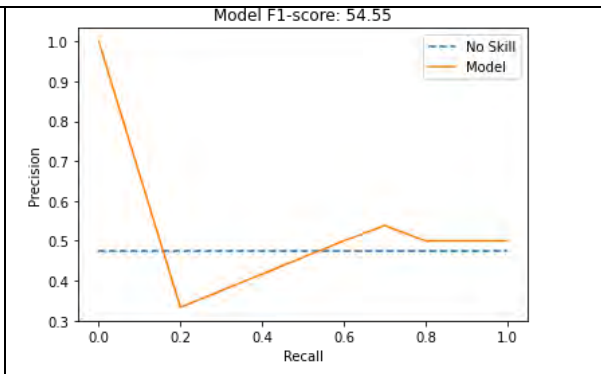
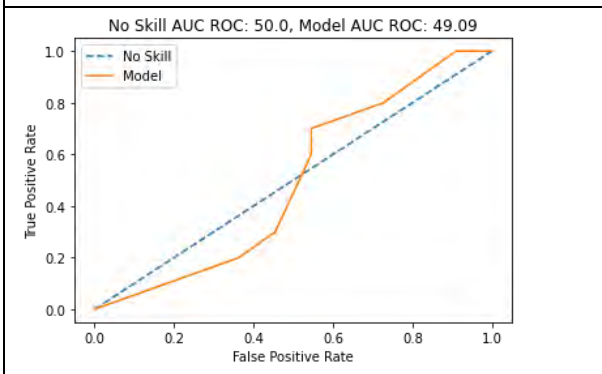
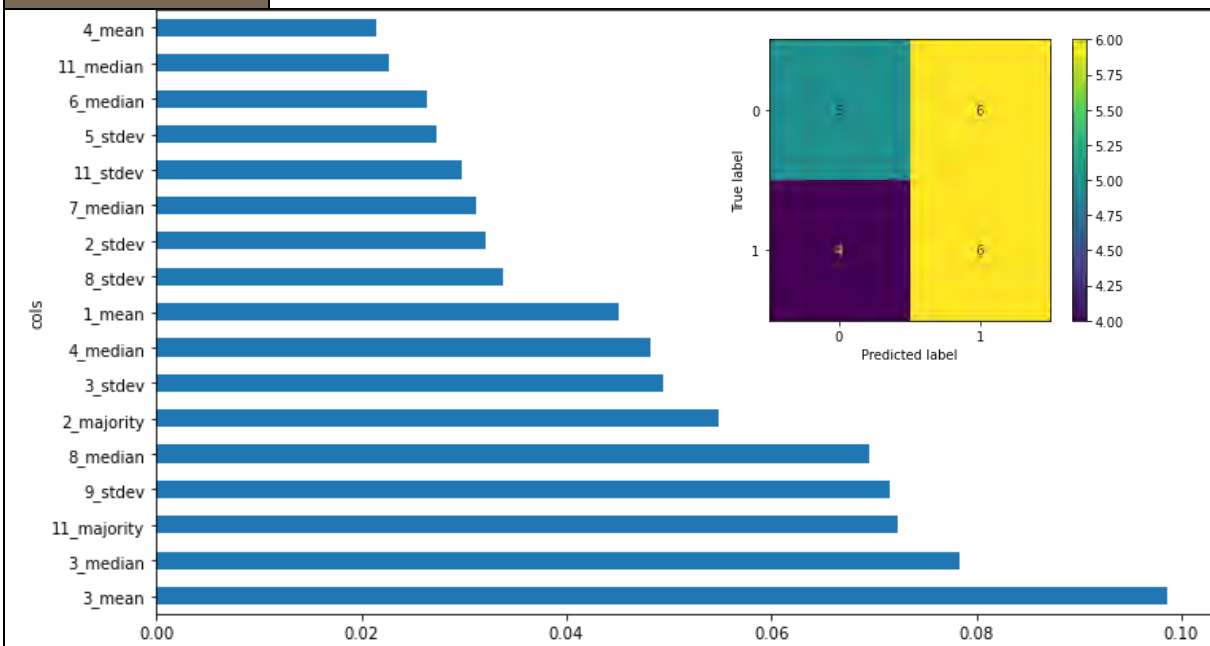
Leyenda imágenes de ejemplo



Ventana clasificador



Yacimiento IAN



Anexo III. Estadísticas descriptivas de las ventanas de observación

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Aire libre	2.03			0 (0%)
C TIPOLOGIA			Lugar de habitación	1.39			0 (0%)
C Group			TOPO	0.562			0 (0%)

G1

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Dolmen	2.03			0 (0%)
C TIPOLOGIA			Lugar funerario	1.4			0 (0%)
C Group			TOPO	0.569			0 (0%)

G2

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Desconocida	2.07			0 (0%)
C TIPOLOGIA			Lugar de habitación	1.53			0 (0%)
C Group			TOPO	0.584			0 (0%)

G3

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.49			0 (0%)
C TIPO			Aire libre	1.94			0 (0%)
C Group			TOPO	0.562			0 (0%)

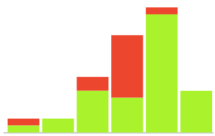


G4

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Aire libre	1.95			0 (0%)
C TIPOLOGIA			Lugar de habitación	1.31			0 (0%)
C Group			TOPO	0.58			0 (0%)

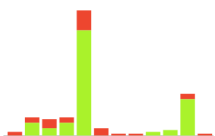
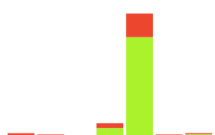

G5

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Dolmen	1.95			0 (0%)
C TIPOLOGIA			Lugar funerario	1.32			0 (0%)
C Group			TOPO	0.557			0 (0%)

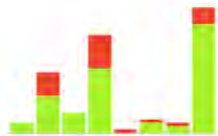
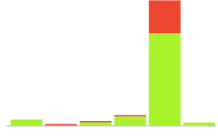

G6

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar funerario	1.53			0 (0%)
C TIPO			Dolmen	2.1			0 (0%)
C Group			TOPO	0.573			0 (0%)


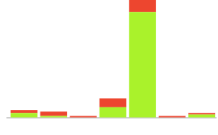

G7

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Dolmen	1.63			0 (0%)
C TIPOLOGIA			Lugar funerario	0.98			0 (0%)
C Group			TOPO	0.564			0 (0%)

G8

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Túmulo	1.58			0 (0%)
C TIPOLOGIA			Lugar funerario	0.694			0 (0%)
C Group			TOPO	0.56			0 (0%)

G9

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Dolmen	1.55			0 (0%)
C TIPOLOGIA			Lugar funerario	0.893			0 (0%)
C Group			TOPO	0.56			0 (0%)

G10

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Dolmen	1.62			0 (0%)
C TIPOLOGIA			Lugar funerario	1.08			0 (0%)
C Group			TOPO	0.559			0 (0%)

G11

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Aire libre	1.86			0 (0%)
C TIPOLOGIA			Lugar funerario	1.35			0 (0%)
C Group			TOPO	0.562			0 (0%)

G12

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.23			0 (0%)
C TIPO			Aire libre	2.18			0 (0%)
C Group			TOPO	0.571			0 (0%)

G13

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.58			0 (0%)
C TIPO			Aire libre	2			0 (0%)
C Group			TOPO	0.569			0 (0%)

G14

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.73			0 (0%)
C TIPO			Aire libre	2.18			0 (0%)
C Group			TOPO	0.573			0 (0%)

G15

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.41			0 (0%)
C TIPO			Aire libre	1.9			0 (0%)
C Group			TOPO	0.569			0 (0%)


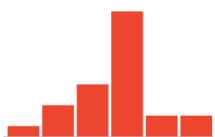

G16

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.55			0 (0%)
C TIPO			Aire libre	2.13			0 (0%)
C Group			TOPO	0.568			0 (0%)

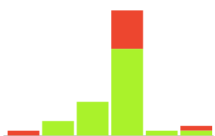


G17

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.41			0 (0%)
C TIPO			Aire libre	1.86			0 (0%)
C Group			TOPO	0.574			0 (0%)

G18

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Aire libre	2.23			0 (0%)
C TIPOLOGIA			Lugar de habitación	1.46			0 (0%)
C Group			TOPO	0			0 (0%)

G19

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.11			0 (0%)
C TIPO			Aire libre	1.99			0 (0%)
C Group			TOPO	0.562			0 (0%)

G20

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.436			0 (0%)
C TIPO			Aire libre	1.47			0 (0%)
C Group			TOPO	0.576			0 (0%)

G21

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Aire libre	1.53			0 (0%)
C TIPOLOGIA			Lugar de habitación	0.786			0 (0%)
C Group			TOPO	0.598			0 (0%)

G22

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.803			0 (0%)
C TIPO			Aire libre	1.85			0 (0%)
C Group			TOPO	0.586			0 (0%)

G23

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	1.36			0 (0%)
C TIPO			Aire libre	2.16			0 (0%)
C Group			TOPO	0.581			0 (0%)

G24

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Aire libre	1.55			0 (0%)
C TIPOLOGIA			Lugar de habitación	0.756			0 (0%)
C Group			TOPO	0.584			0 (0%)

G25

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPO			Aire libre	1.48			0 (0%)
C TIPOLOGIA			Lugar de habitación	0.703			0 (0%)
C Group			TOPO	0.58			0 (0%)

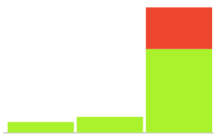
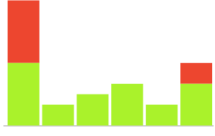

G26

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.569			0 (0%)
C TIPO			Aire libre	1.73			0 (0%)
C Group			TOPO	0.586			0 (0%)

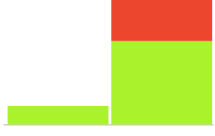


G27

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.683			0 (0%)
C TIPO			Aire libre	1.59			0 (0%)
C Group			TOPO	0.576			0 (0%)

G28

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.576			0 (0%)
C TIPO			Aire libre	1.57			0 (0%)
C Group			TOPO	0.589			0 (0%)

G29

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.385			0 (0%)
C TIPO			Núcleo de población	1.47			0 (0%)
C Group			TOPO	0.602			0 (0%)

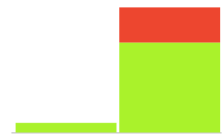

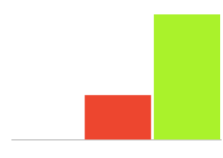
G30

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.471			0 (0%)
C TIPO			Aire libre	1.62			0 (0%)
C Group			TOPO	0.593			0 (0%)

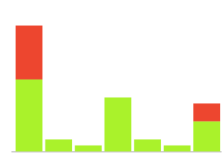

G31

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.658			0 (0%)
C TIPO			Aire libre	1.71			0 (0%)
C Group			TOPO	0.586			0 (0%)

G32

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.257			0 (0%)
C TIPO			Aire libre	1.45			0 (0%)
C Group			TOPO	0.575			0 (0%)

G33

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.356			0 (0%)
C TIPO			Aire libre	1.44			0 (0%)
C Group			TOPO	0.586			0 (0%)

G34

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.599			0 (0%)
C TIPO			Aire libre	1.59			0 (0%)
C Group			TOPO	0.562			0 (0%)

G35

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.308			0 (0%)
C TIPO			Núcleo de población	1.1			0 (0%)
C Group			TOPO	0.582			0 (0%)

G36

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.119			0 (0%)
C TIPO			Núcleo de población	1.16			0 (0%)
C Group			TOPO	0.569			0 (0%)

G37

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.219			0 (0%)
C TIPO			Aire libre	1.3			0 (0%)
C Group			TOPO	0.57			0 (0%)

G38

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.356			0 (0%)
C TIPO			Núcleo de población	1.41			0 (0%)
C Group			TOPO	0.569			0 (0%)

G39

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.375			0 (0%)
C TIPO			Núcleo de población	1.39			0 (0%)
C Group			TOPO	0.573			0 (0%)


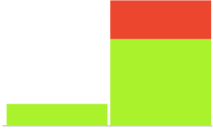

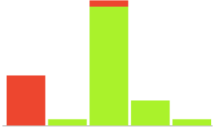


G40

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.38			0 (0%)
C TIPO			Núcleo de población	1.46			0 (0%)
C Group			TOPO	0.602			0 (0%)

G41

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
C TIPOLOGIA			Lugar de habitación	0.391			0 (0%)
C TIPO			Núcleo de población	1.36			0 (0%)
C Group			TOPO	0.581			0 (0%)

G42

Name	Distribution	Mean	Median	Dispersion	Min.	Max.	Missing
 TIPOLOGIA			Lugar de habitación	0.418			0 (0%)
 TIPO			Núcleo de población	1.11			0 (0%)
 Group			TOPO	0.578			0 (0%)

G43

