

Reto Minsait Land Classification

UniversityHack2020 - Equipo: CMBC



UNIVERSITYHACK 2020®
DATATHON

Tabla de contenidos

1. [Introducción](#)
2. [Tecnología necesaria para la realización de este proyecto](#)
3. [Breve resumen del trabajo desarrollado](#)
4. [Análisis exploratorio y manipulación de variables](#)
 - [Variables numéricas](#)
 - [Geoposición](#)
 - [Color](#)
 - [Geometría](#)
 - [Otras](#)
 - [Variables discretas](#)
5. [Construcción y justificación selección de los modelos](#)
 - [Estrategias probadas pero no implementadas](#)
6. [Conclusiones](#)

1. Introducción

Este proyecto fundamenta la propuesta ideada por el equipo **CMBC** para el reto de [UniversityHack 2020 DATATHON](#)

El equipo está compuesto por:

- [Cristian Cifuentes García](#)
- [Manuel Bermúdez Martínez](#)

El objetivo del reto **Minsait Land Classification** consiste en maximizar la *exactitud*, que se define como el *número de registros correctamente clasificados / número total de registros proporcionados por la Organización*, tal y como se indica en la propia página oficial.

Para ello se cuenta con dos ficheros, los cuales contienen un listado de superficies sobre las que se han recortado la imagen del satélite Sentinel II del servicio Copernicus de la Agencia Espacial Europea y se han

extraído una serie de características de sus geometrías, posición, colores, etc. Y finalmente, se ha etiquetado el conjunto de los datos según la clasificación de suelo.



Se cuenta con un fichero para realizar el análisis y la generación de los algoritmos de clasificación empleados, denominado *Modelar.txt* y también con un segundo fichero, *Estimar.txt*, el cual se utilizará para realizar la entrega a la organización. Este segundo fichero contiene una serie de registros con las mismas variables pero sin la etiqueta que clasifique el suelo.

2. Requisitos y estructura del proyecto

Los requisitos necesarios para la realización del proyecto son los siguientes:

- [Python](#)
- [Jupyter notebook](#)
- [scikit-learn](#)
- [Pandas](#)
- [XGBoost](#)

Por otro lado, la estructura del contenido del proyecto es:

```
├── Minsait_Land_Classification_CMBC.ipynb
├── UCLM_CMBC.txt
├── data
│   ├── Estimar_UH2020.txt
│   ├── Modelar_UH2020.txt
│   └── img
│       └── AUCPredictionsRanked.svg
```

```
├── diagrama_modelos.png
├── logo_light.png
├── mapa_introduccion.jpg
├── sentinel_resolution.jpg
├── validacion_cruzada.jpeg
├── models
│   ├── binary_final.model
│   └── multilabel_final.model
```

Donde tenemos en el directorio raíz la libreta (*Minsait_Land_Classification_CMBC.ipynb*) que contiene el trabajo realizado, el dataset de entrega *UCLM_CMBC.txt*, la carpeta **data** con los ficheros de datos utilizados y las imágenes que se muestran en el interior de la libreta, la carpeta **models** con los modelos aprendidos (necesarios si no se desea compilar por completo la libreta) y un fichero *pdf* con el resumen del trabajo desarrollado.

3. Breve resumen del trabajo desarrollado

Con el fin de cumplir el principal objetivo del reto, se ha realizado un extenso análisis previo para comprender y aprender lo máximo posible acerca de las imágenes extraídas por el satélite Sentinel II del servicio Copernicus de la Agencia Espacial Europea, ya que la mayoría de las variables del conjunto de datos pertenecen a la misma. Además, se ha comprobado la importancia de cada una de estas variables en función de nuestro propósito.

Después de revisar toda la información posible y de un análisis exploratorio, el cual se comenta en el siguiente apartado, se ha desglosado el conjunto de datos en función de los problemas que han ido surgiendo (valores atípicos, valores perdidos, etc). Una vez estudiado el conjunto de datos, se ha continuado con **una estrategia de apilamiento de modelos**, es decir, se ha desarrollado un apartado comparando diversos modelos binarios para finalmente, seleccionar aquel que ofrezca un resultado más preciso y robusto. Posteriormente se ha desarrollado un modelo multietiqueta que se utiliza sobre la predicción obtenida en el primero modelo.

En todos y cada uno de estos modelos se ha realizado un intenso estudio de los hiperparámetros y distintos conjuntos de entrenamiento y validación mediante técnicas como la validación cruzada. Finalmente se han comprobado los resultados con un conjunto de validación previamente definido, el cual no se ha utilizado en ninguna otra fase del proyecto.

4. Análisis exploratorio y manipulación de variables

El principal problema de este reto es el desbalanceo que existe en el conjunto de datos proporcionado por la Organización, por lo tanto, para intentar reducir o comprender los registros y las variables, se ha realizado un profundo análisis del mismo y previamente de los aspectos que pudiesen favorecer la tarea del mismo.

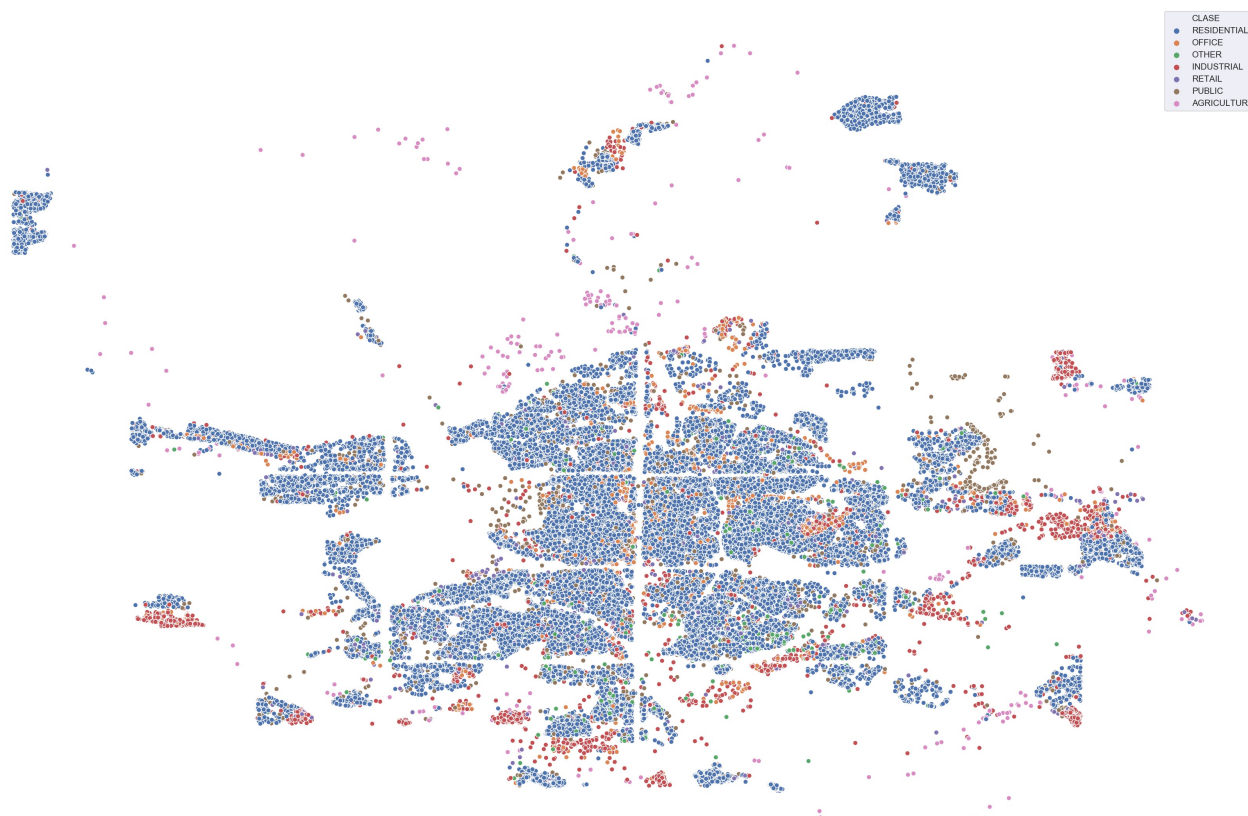
Primeramente y tras comprobar el total de registros que presenta cada una de las diferentes etiquetas de la variable objetivo se ha llevado a cabo una división en función del tipo de cada una de las variables, dando como resultado una lista de variables numéricas y otra de variables categóricas. El análisis de las mismas se ha realizado de forma independiente ya que cada tipo requiere un tratamiento distinto.

Variables numéricas

Para las variables numéricas se ejecutó otra separación en función de la información que aporta cada una de las variables, produciendo así cuatro grupos distintos: geoposición, color, geometría y otras.

Geoposición

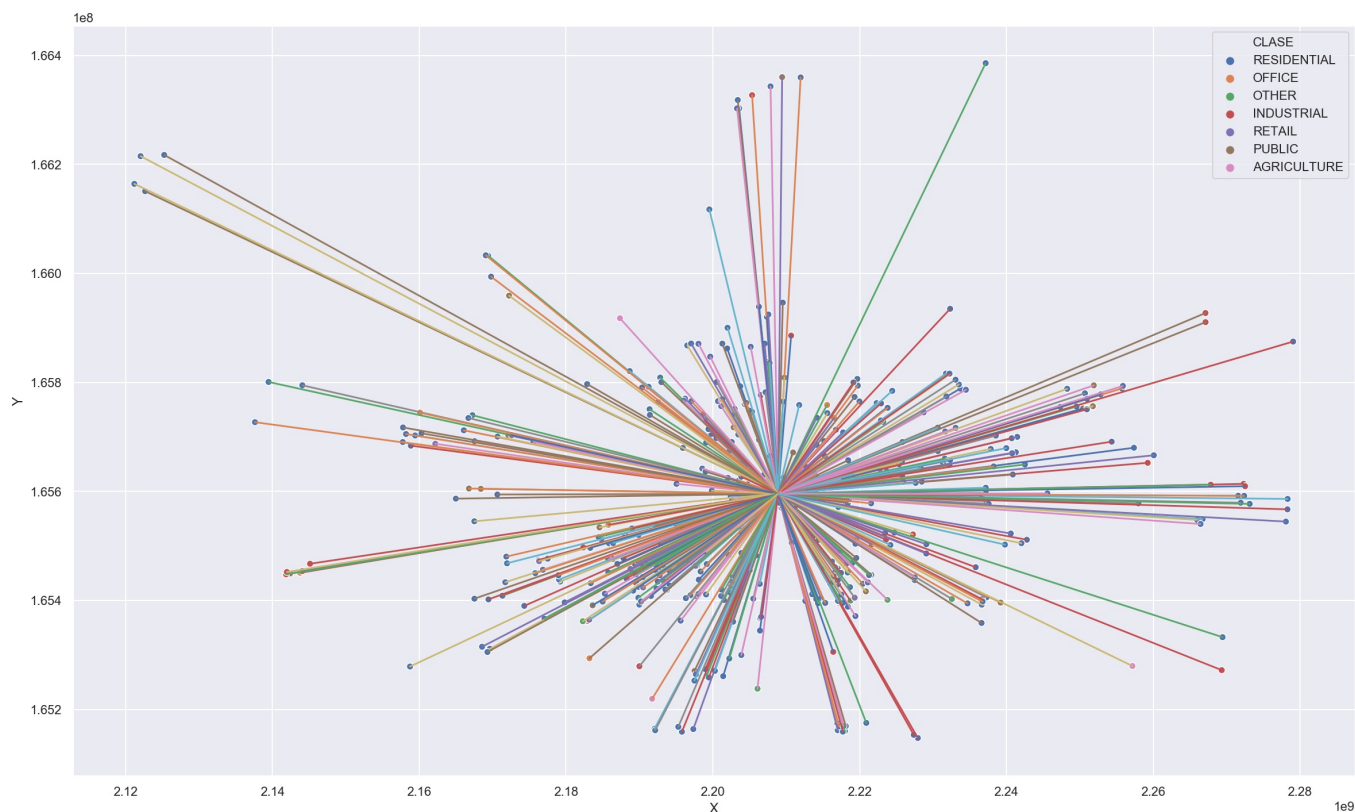
Estas variables, como ya sabemos gracias a la información proporcionada por la Organización, han sido escaladas y desplazadas aleatoriamente, manteniendo la relación con el resto de registros. Aún así, analizando de forma gráfica este tipo de variables hemos podido comprobar que la mayor parte de los registros están situados de forma uniforme en un mapa y que además, no existe una distinción por zonas ya que podemos encontrar terreno de cualquier tipo cerca de otro, aunque bien es cierto que en ciertos casos esto se produce con muy poca frecuencia. Por ejemplo, etiquetas como **Agriculture** o **Industrial** están situadas en las afueras del lo que sería el centro de la imagen tomada como referencia.



Se han optado por probar las estrategias de clasificación *KNeighborsClassifier* y *KMeans* pero nos hemos decantado por la primera de ellas. La cual nos permite obtener una estimación en función de las variables de geoposición. El algoritmo implementado devolvía la probabilidad con respecto al *k* valor elegido de que fuese de un tipo u otro sus correspondientes vecinos, esto nos permite obtener una especie de densidad con respecto a la vecindad en función a la localización en la que se encontrase el registro a predecir.

Los resultados obtenidos tras entrenar los modelos con esta nueva serie de variables no han resultado demasiado positivos tras la comprobación realizada en una de las entregas intermedias propuesta por la organización, por lo que hemos decidido descartar esta opción.

Aunque la otra opción no haya tenido éxito, **hemos generado una nueva variable que nos permite saber la distancia de cada uno de los registros con respecto a un punto céntrico**. Este valor es el centro calculado por todos los registros que conforman nuestro actual conjunto de datos.



Color

En cuanto a las variables correspondientes al color de la imagen sabemos que, después de leer información sobre este tipo de satélite, se ha extraído información de 4 canales (R, G, B y NIR), correspondientes a las bandas de color rojo, verde y azul, y el infrarrojo cercano. El valor mostrado en cada una de estas variables corresponde a la intensidad por deciles en cada imagen.

La banda de color **rojo corresponde a la banda 4 del espectro visible**, el **verde a la banda 3** y el **azul a la banda 2**, mientras que el **infrarrojo cercano hace referencia a la banda 8** del espectro electromagnético. Dichas bandas junto con el valor de su longitud de onda central se puede observar en la imagen adjunta. Además, todas ellas presentan una resolución de **10m/px**.

De primeras podemos pensar que al ser variables relacionadas con la intensidad del color, se podría obtener un histograma del mismo pero se ha comprobado que la suma total de los mismos no es idéntica y por lo tanto, se ha descartado la posibilidad de representar un histograma. Junto a esto, se ha observado que para todos los registros el valor de los deciles aumenta progresivamente, presentando unos valores elevados en el último decil y valores cercanos a 0 en el menor de los deciles.

Durante este análisis se ha realizado una **reducción de la dimensionalidad** de estas variables, mediante PCA, con el fin de seleccionar las características más influyentes para nuestro modelo y hemos comprobado que no obtenemos los resultados esperados, por lo que se ha terminado por descartar esta metodología. Además, se ha podido comprobar que cada una de estas variables presenta una correlación muy elevada con el decil anterior y posterior al mismo.

Como consecuencia de ello, hemos optado por reducir la dimensionalidad de algunos de las variables referentes a los deciles que presenten una alta correlación entre sus vecinas. Los deciles que no se van a ver afectados en las variables referentes a los colores van a ser:

- Decil 0: Debido a que se trata del valor mínimo de cada uno de los grupos.

- Decil 10: Debido a que representa el valor máximo de cada uno de los grupos.
- Decil 1 y 9: Puesto que no presentan correlación directa con los deciles 0 y 10.
- Del resto de deciles, nos quedaremos con los que representan los deciles impares, **eliminando de este modo los deciles 2, 4, 6 y 8.**

Geometría

Otro de los aspectos importantes de cara a la clasificación del terreno son las variables geométricas, por ello mismo hemos llevado a cabo un estudio en profundidad de las mismas y una comparación con el resto de variables, independientemente del tipo que presente. Con ello se ha llegado a la conclusión, como era de esperar, que el área más elevada se presenta en etiquetas como **Agriculture**, **Retail** y **Other**.

Asimismo se han estudiado todos los valores de estas variables ya que presentan *outliers*, dando como resultado la eliminación de los mismos para comprobar el conjunto de datos resultante, el cual ha disminuido notablemente el número de registros. A consecuencia de ello, se ha comprobado que no todos ellos son valores atípicos y por lo tanto, no se ha realizado una eliminación sobre el conjunto de datos final. Además, esta técnica se probó en una de las entregas intermedias y vimos que no se producía mejora en los resultados, probablemente a un sobreajuste del modelo.

Otras

Este grupo de variables es el más reducido y además, aportan información sobre los edificios o terrenos colindantes y no sobre el cual se pretende realizar una clasificación. Aún así, en el estudio de estas dos variables se ha visto que presentan valores perdidos, los cuales se han tratado en un *Pipeline*.

Variables discretas

Este conjunto de variables es reducido en comparación con las variables numéricas. Para este caso no he hecho falta realizar ninguna subdivisión de las variables y en el estudio de las mismas se implementado una función que permita posteriormente preprocesar la variable realacionada a la calidad del catastro.

Junto a este implementación se ha demostrado que la mayor parte de los terrenos del conjunto de datos pertenecen a una calidad media del catastro, situandose en los valores intermedios del rango establecido.

5. Construcción y justificación selección de los modelos

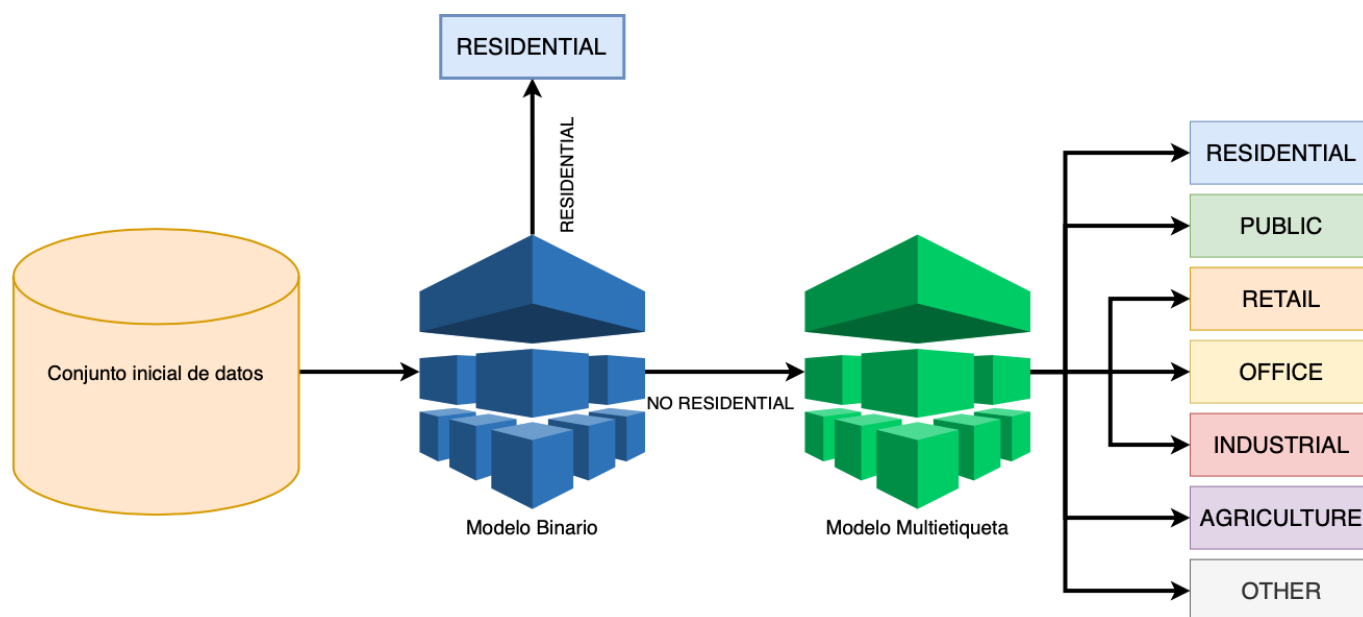
Como se ha comentado al inicio, se han desarrollado diversos modelos con el fin de buscar una diversidad entre los mismos y seleccionar aquel que ofrezca unos resultados más precisos y robustos. Además, cabe destacar que se ha realizado una técnica, la cual consiste en el **apilamiento de modelos**, es decir, primeramente se ha seleccionado y entrenado un modelo binario y posteriormente, con los resultados obtenidos se ha desarrollado un modelo multietiqueta.

Para poder trabajar correctamente, en este apartado se han utilizado elementos como *Pipelines* y *ColumTransform* con el fin de poder obtener un conjunto de datos óptimo para el entrenamiento de los modelos necesarios. En estos *Pipelines* se realiza una normalización de los valores de cada una de las variables, un **over-sampling** mediante la técnica de *SMOTE* con el fin de balancear el conjunto de datos y por último, se realizan una serie de funciones previamente implementadas.

En el primero de ellos, el modelo binario, se ha buscado maximizar los hiperparámetros usando siempre el sentido común. Para conseguir esto se ha hecho uso de técnicas de validación cruzada y *GridSearchCV*. El resultado de este procedimiento fue la obtención de dos modelos robustos en función de la métrica establecida, *f1-score*, un **Random Forest** y un **XGBoost**.

En el segundo de ellos, el modelo multietiqueta, de igual forma se han buscado maximizar los hiperparámetros pero debido a la complejidad del problema y al elevado número de registros se ha optado por un **Random Forest** el cual ha mostrado siempre resultados más que decentes y robustos en todas las pruebas realizadas. Este modelo cuenta con todas las etiquetas de la variable a predecir ya que clasificará en función de los datos de salida que genere el modelo binario. Como se espera que el modelo binario presente una tasa de fallos se mantiene la etiqueta **Residential** en este modelo multietiqueta para reducir el error del primer modelo.

Un esquema muy sencillo de la estrategia utilizada es el mostrado a continuación:



5.1 Estrategias probadas pero no implementadas

Algunas de las estrategias que se tuvieron en cuenta y se han ido probando a lo largo del desarrollo del proyecto han sido las siguientes:

- Con respecto a la técnica de **over-sampling** utilizada, se han probado algunos algoritmos como son: **SMOTETomek** o **SMOTEENN**, pero seleccionamos la que mejores resultados obtenía.
- También se probó la utilización de otra técnica denominada **under-sampling** mediante algoritmos como **RandomUnderSampler** o **NearMiss**, dando peores resultados que la utilizada finalmente.

Todas estas técnicas comentadas se han ido realizando y comprobando junto a la creación de algunas variables o la utilización de otras técnicas para la reducción de la dimensionalidad del conjunto inicial de datos. Finalmente, nos hemos decantado por aquellas técnicas y variables que, en conjunto, nos otorgan un mejor resultado de cara al conjunto de validación.

6. Conclusiones

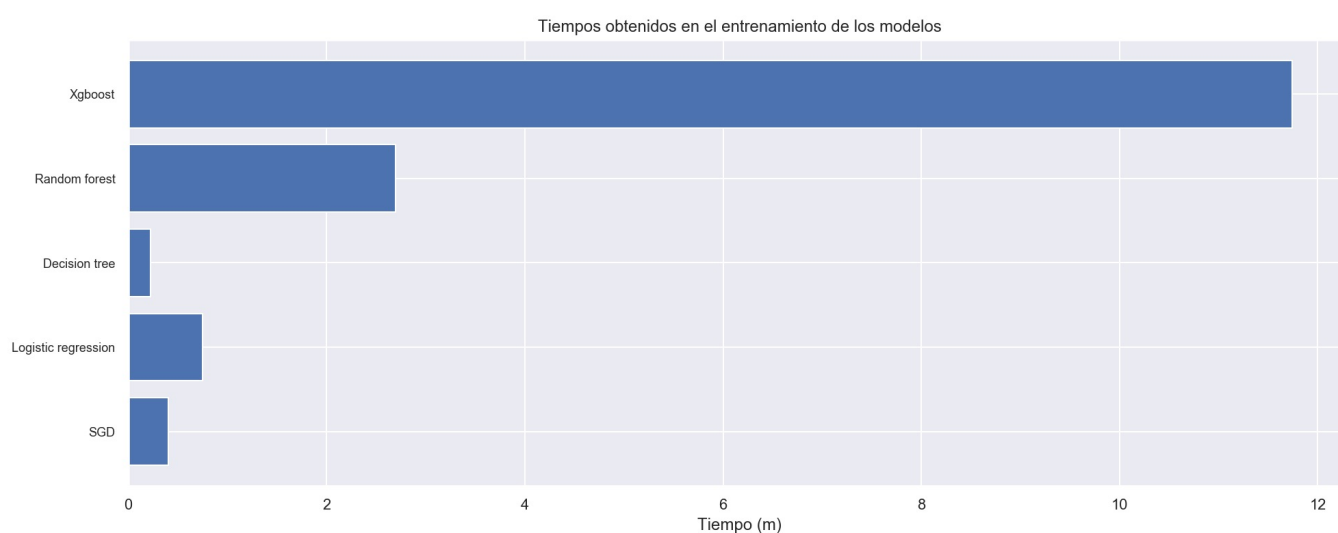
Una vez se ha realizado la estrategia anteriormente comentada, se ha generado una tabla con los resultados obtenidos en el entrenamiento de cada uno de los algoritmos de clasificación para el modelo binario, la cual se muestra a continuación:



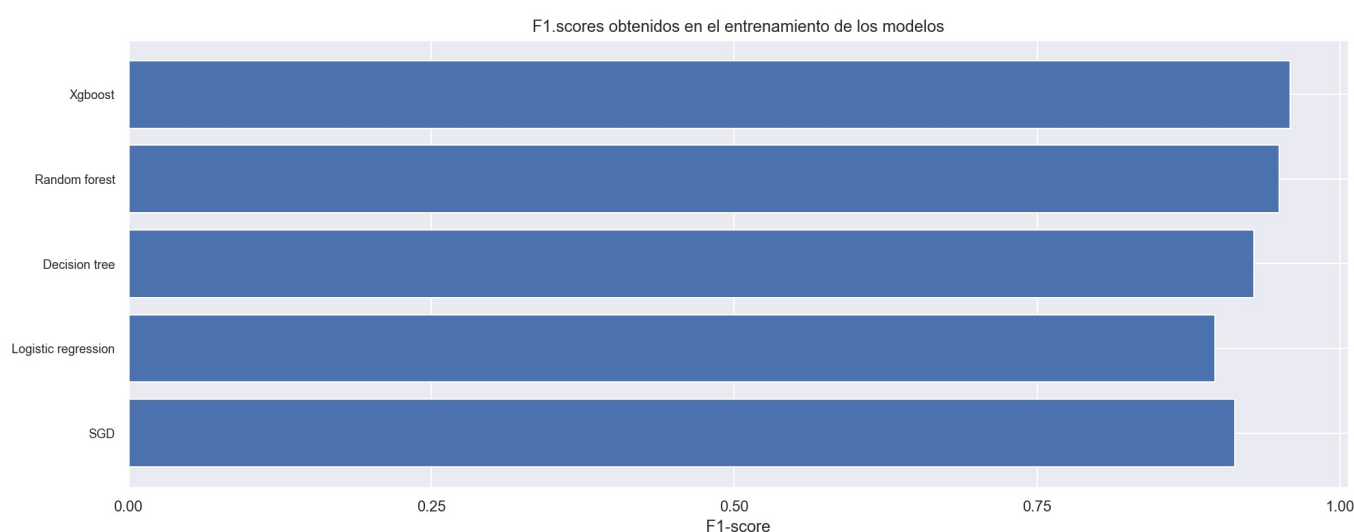
Estos tiempos dependen principalmente del ordenador utilizado para el entrenamiento de los modelos.

Modelo	Tiempo (min)	F1 score	Parámetros
SGD	0.403	0.913	max_iter: 50, tol: 0.0001
Logistic regression	0.751	0.897	C: 1.0, tol: 0.001
Decision tree	0.224	0.929	max_depth: 10
Random forest	2.697	0.95	max_depth: 20, n_estimators: 100
XGBoost	11.743	0.959	max_depth: 20, n_estimators: 200

Aquí se puede observar de forma gráfica:




Y los *f1-score* obtenidos son:



Como se puede comprobar, el **Random Forest** y el **XGBoost** no presentan una gran diferencia, pero a fin de obtener el más robusto se ha optado por el segundo.

Tras aplicar el segundo modelo, el multietiqueta, al conjunto de validación previamente definido, los resultados obtenidos son:

Modelo	Tiempo (min)	Accuracy	Parámetros
Random forest	13.1	0.8484	max_depth: 20, n_estimators: 200, class_weight: 'balanced'

 **Los tiempos obtenidos han sido mediante la ejecución de la libreta con el MacBookPro de 16' cuyas especificaciones son un procesador Intel i9 y 16GB de RAM.**

Las clasificaciones obtenidas con nuestros modelos sobre el conjunto de datos a estimar son las que se muestran en el siguiente gráfico:

