CaixaBank Tech Hackathon

Para enfocar este reto he seguido una serie de pasos. Primero observar el conjunto de datos, limpiarlos y realizar un análisis exploratorio para comprender el estado de la situación. Para ello he eliminado las hileras con valores NaN ya que estos valores estaban presentes en todos los atributos de cada hilera y no representaban una parte importante del conjunto.
He decidido eliminar también las hileras que tenían adjudicado un valor de Volumen igual a 0, esto ha reducido más considerablemente el conjunto de datos, pero al observar que estas hileras pertenecían a los primeros años del conjunto he supuesto que por algún motivo no se conservan estos datos antes del año 2000 y, por lo tanto, este período tiene una falta de información y de momento se eliminan.
Después de estos procesos iniciales he preparado algunas funciones para estimar y luego entrenar y testear los futuros modelos. La función para estimar la bondad de los modelos utiliza lo que se conoce como *nested cross-validation* para que sus resultados sean realmente fiables y generalizables. La siguiente función usa un *GridSearch* para afinar los parámetros del modelo y luego aplica el que considera que es el mejor *threshold* al modelo antes de generar las predicciones y las métricas.

Una vez preparadas las funciones y los algoritmos escogidos, he hecho una primera aproximación a los resultados que generan para tener elementos para poder comparar después. Los resultados han sido especialmente malos, prácticamente azarosos.

El siguiente paso ha sido jugar un poco más con el conjunto y ver qué opciones había, he estado buscando nuevos posibles atributos a partir de la información original. De aquí han surgido algunos que finalmente se han quedado para entrenar el modelo final, pero la mayoría no han conseguido aportar ningún valor al modelo, solo ruido prácticamente. En algunos casos se ha conseguido reducir el número de falsos positivos, siendo 1 la clase positiva. Es decir, se ha conseguido reducir el número de situaciones en las cuales el modelo predice que dentro de 3 días el valor del IBEX será superior al actual y en realidad no es así. Aunque aumentar la *sensitivity* del modelo no es el objetivo, sigue siendo interesante reducir el riesgo de equivocarse en ese aspecto y construir un modelo precavido pero más seguro.

Después de probar con distintas variables nuevas he escogido alguna que parece que ha ayudado un poco. También he decidido probar con un método sencillo de aprendizaje no supervisado. Con un simple *kmeans* he clusterizado el conjunto de manera sencilla para añadir estos agrupamientos como una nueva variable, aparentemente también ha ayudado muy ligeramente. Podría ser interesante profundizar más en otras técnicas no supervisadas para ver si son capaces de encontrar información dentro del conjunto que ayude a clasificar un poco mejor.

He probado también de separar el conjunto en partes y crear un modelo para cada parte para ver si así los modelos respondían mejor. La idea luego sería decidir qué modelo usar en cada instancia basándose en el mismo criterio de separación del conjunto. Aun así, después

de probar un par de veces, los resultados estaban siendo peores y he decidido no seguir buscando una manera más adecuada por falta de tiempo.
Finalmente, entre los algoritmos *xgb, random forest y knn* escogidos inicialmente, el modelo final se ha construido con el algoritmo *random forest*. *Xgb* acostumbra a funcionarme muy bien y siempre está entre los candidatos iniciales, en este caso no ha sido posible porque he mantenido algunas variables categóricas y a día de hoy la librería *xgb* no ha implementado la posibilidad de entrenar un modelo con variables categóricas para *xgb*. *Knn* aunque en la estimación ha dado resultados muy parecidos a random forest luego al comparar las métricas de modelos entrenados ha dado peores resultados.

Los resultados en general no han mejorado, no he conseguido aumentar el *f1 macro*, aun así, con algunos atributos descartados y moviendo un poco el threshold se puede conseguir reducir más los falsos positivos.

Con respecto al conjunto de datos de *tweets*, no voy a incluirlo en el *notebook* final. No he terminado de implementar nada, me he quedado corto de tiempo. He empezado limpiando la columna con las fechas usando *backreferences* de *regex* y eliminando 4 hileras con datos erróneos. Luego he empezado a crear una función para normalizar los tweets. He reciclado una que ya tenía, pero para tweets en inglés y me he encontrado con la situación de que con la librería nltk no puedo *lematizar* en castellano, solo *stemizar*.

He buscado otra librería para hacerlo, he encontrado *Stanza* y he añadido la funcionalidad para *lematizar* en castellano dentro de mi función. El problema está en que la función para hacerlo es poco eficiente si tienes que implementarla muchas veces en textos cortos, está pensada para textos largos, y tardaba demasiado y he decidido centrarme en el otro conjunto de datos. Aun así, la idea era vectorizar las listas de palabras normalizadas utilizando *CountVectorizer* y *TfidfTransformer*, y luego tratar de añadir esta matriz de vectores al otro conjunto de datos a partir de las fechas usadas como índice y ver qué posibilidades había.