BEATING STOCK MARKETS

por Esteban Manuel Sánchez García

- **INTRODUCCIÓN**

El proyecto se enmarca dentro de lo que se conoce como trading algorítmico centrado en datos. El objetivo es predecir que va a ocurrir en el mercado a corto plazo, entendiendo este corto plazo como 3, 5, 7, 14 y 21 días desde la última fecha de la que se tenga precio de cierre para evitar pérdidas, mediante técnicas de machine learning.

Este trabajo nace de cero con la intención, además de lo explicado en el párrafo anterior, de aprender de temas bursátiles a la par que todas las fases de un proyecto en ciencia de datos. A saber, adquisición de datos, tratamiento del dato, modelado y visualización.

La idea de la que parte el proyecto es encontrar patrones con los diferentes algoritmos de machine learning entre el precio de futuro y los indicadores técnicos de hoy.

- **ADQUISICIÓN Y TRATAMIENTO**

Se usan dos grandes fuentes de datos mediante vías diferentes. La primera, los datos categóricos, como pueden ser los índices bursátiles (IBEX, Dow Jones) el país o el sector al que pertenece una determinada compañía, que han sido extraídos mediante webscrapping a yahoo finance de diferentes países. La segunda, los datos financieros, como pueden ser los precios de cierre, el volumen y diversos indicadores técnicos como las medias móviles o los momentos han sido descargados mediante una API de AlphaVantage. El código usado para hacer estas descargas se encuentra dentro de GitHub - Beating Stock Markets en las carpetas de bsm_categorical y bsm_financials.

Dentro del link facilitado anteriormente se encuentra una carpeta denominada exploring_data donde se encuentran diferentes notebooks y un script de r markdown, usados para hacer análisis categóricos de los datos extraídos anteriormente. Aquí se rellenan campos nulos pertenecientes a datos financieros con regresiones lineales y se eliminan todas aquellas compañías que aparecen duplicadas en diferentes sectores, dado que para todo el proceso tenemos que mantener una clave única entre compañía y sector. Durante el desarrollo se ha encontrado que compañías como Mediaset, que es italiana, también se encuentra en España pero con otro código bursátil (o ticker) esto se ha mantenido.

- **MODELADO Y VISUALIZACIÓN**

Para tratar de cumplir el objetivo marcado en la introducción, se ha decidido atacar el problema mediante técnicas de clasificación. Lo primero que se ha realizado, una vez dividido el set de datos en entrenamiento y test, ha sido categorizar cada día en si se trata de una subida o una bajada. Para ello se han tomado las restas de los precios de cierre entre los días a predecir, con esto se obtiene información de cómo, con los datos de hoy, ha subido o bajado con respecto al día a predecir. Tras esto; para categorizar si ha sido una subida fuerte, normal o floja, se han tomado las subidas por ticker y mes y se han calculado los percentiles 25, 50 y 75 donde, si la subida ha sido mayor que el valor correspondiente al percentil 75, se considerará como subida fuerte (S. Bull, strong bull), si está entre el percentil 50 y el 75 será una subida normal (Bull) y si el valor es menor al percentil 50 entonces se considera una subida floja (W. Bull, weak bull). La misma idea subyace para categorizar las bajadas. El término bull para las subidas se usa en ámbito bursátil cuando una compañía está al alza y bear cuando está a la baja. Con estas etiquetas ya si se han podido realizar técnicas de clasificación.

Tras esto se ha llevado a cabo un estudio automático de selección del algoritmo que arroje la mejor precisión. Se ha usado la precisión dado que se busca minimizar pérdidas, y para ello es necesario que cuando el modelo prediga una categoría sea lo más fiable posible. Los algoritmos que se han usado para este estudio han sido random forest, lightgboost, regresión logística, k vecinos, árbol de decisión, votting classifier, bagging classifier y ada boost classifier. A todos ellos se les ha buscado los mejores parámetros mediante un gridsearch, una función dentro de scikit-learn que permite encontrar los mejores hiper parámetros para un algoritmo. Una vez seleccionado el mejor algoritmo, se entrena el modelo y se guarda el modelo que luego será consumido para predecir y elaborar los dashboard que un usuario podrá utilizar. Además, mediante un parámetro dentro del código, se tiene la posibilidad de hacer una extracción de variables antes de todo el entrenamiento mediante un análisis de varianza (ANOVA) o calculando la información mutua de las características.

Con el fin de mostrar visualmente que algoritmos son mejores en cada sector y por día, a la par que visualizar las precisiones que el modelado arroja, se ha desarrollado un notebook de visualización con nombre models_viz.ipynb dentro de la carpeta exploring_data del GitHub facilitado en la introducción. Además, para la elaboración de los dashboard se ha usado la herramienta Tableau. Los resultados se pueden ver en los siguientes links:

1. Dashboard sin extracción de variables: beating_stock_market
2. Dashboard ANOVA: beating_stock_market_f_classif
3. Dashboard información mutua: beating_stock_market

Los dashboard se han construido como sigue. La primera hoja es un mapa del mundo donde se ve información de la cantidad de compañías por países que hemos tratado; al clicar sobre uno de los países nos lleva a una serie temporal para cada sector que se encuentra dentro de ese país, que es un promedio de todas las compañías que se

encuentran en ese sector para ese país. Cuando haces clic sobre alguno de los sectores te dirige a una estadística de aciertos y fallos para cada una de las clasificaciones por día a predecir para dicho sector. La siguiente hoja se accede haciendo clic sobre cualquiera de las barras; y nos muestra los límites que marcan las clasificaciones para cada compañía dentro del sector. Para finalizar, y tras hacer clic sobre una compañía, se nos muestra la serie temporal de la compañía, donde además podemos ver las predicciones que arroja cuando seleccionamos un día y las categorías que realmente tiene asociada, con esto se puede comparar si se acierta o si no. Además, sobre esta misma hoja, aparecen unas gráficas de barras que indican el número de veces que, aun habiendo fallado en la predicción se ha acertado en si sube o si baja.

- **CONCLUSIONES**

En el notebook de models_viz se puede apreciar cómo las precisiones por normal general oscilan entre 0 y 40 % habiendo algunas muy buenas por encima del 75 %, también se encuentra que las clasificaciones de subidas se predicen mejor que las de bajada. Además, cuando realizamos un procesado de extracción de variables parece que no mejoramos mucho, aunque se observe cierta mejora en las clasificaciones de bajada. Se concluye que el resultado no ha sido lo que se esperaba. Sin embargo, se pueden observar ciertos casos donde se obtiene un buen resultado, por lo que cambiando el modelo usado y haciendo un preprocesado de los datos más exhaustivo, se podría llegar a conseguir mejores puntuaciones. En los dashboard de Tableau se observa que además por norma general, cuando se predice algo, aunque falle es más probable que falle en su propia categoría, es decir que cuando se predice que sube en una de las tres categoría, es más probable fallar a que sube a que baja.