Reto Atmira Stock Prediction - ETSII-Corp

realizado por: Manuel Bueno Gómez, Pablo Santos Ortiz y Jaime Raynaud Sánchez

1. Resumen del trabajo desarrollado

En este trabajo, el equipo ha desarrollado dos scripts en python, uno para explorar los datos y otro para trabajar con ellos, así como un fichero txt “ETSII-Corp.txt” con los resultados de la predicción en el formato adecuado.

En el script de exploración hemos realizado un pequeño estudio para ver la distribución de los datos, así como la correlación entre variables y los tipos de las mismas, entre otras técnicas.

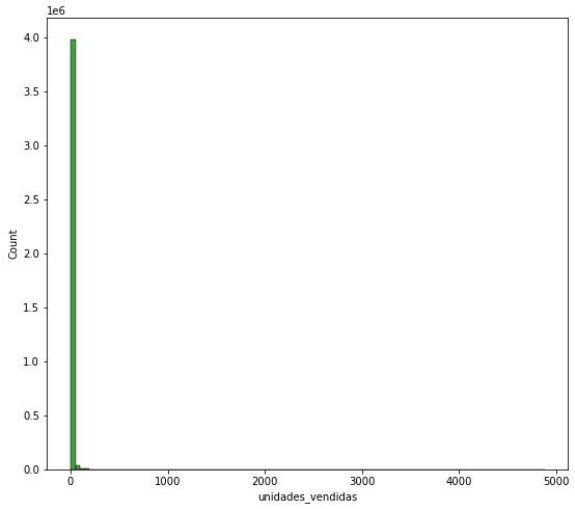
En el script de predicción, hemos refactorizado los datos de entrada de manera pertinente para poder trabajar con ellos, además de eliminar las variables que no nos interesaban, las cuales obtuvimos gracias al script de exploración, para posteriormente aplicar la técnica XGBoost Regression para predecir los resultados de unidades_vendidas, obteniendo un RMSE sobre nuestro test set de 16.310883, que teniendo en cuenta que los valores de unidades_vendidas varían entre 0 y 4881, nos parece un resultado satisfactorio.

La organización y metodología seguida ha sido Scrum aplicada a un proyecto de este tamaño, para una mejor y más ágil ejecución del mismo.

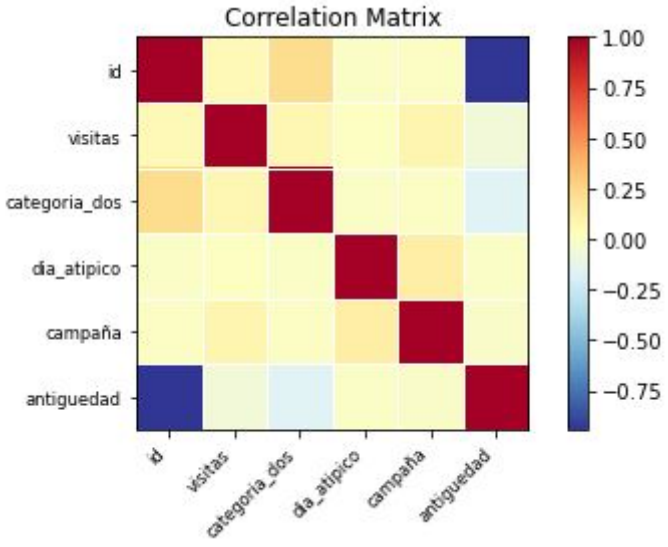
2. Resumen del análisis exploratorio

En este script se realiza una exploración básica sobre las variables y datos.
Para empezar mostramos por pantalla mediciones básicas sobre los datos del dataset, como los tipos de las features, distintas medidas estadísticas de las mismas y la cabecera del dataset.

Posteriormente hemos observado mediante un histograma de ‘*unidades_vendidas* ‘que la gran mayoría de datos tienen un valor por debajo de 200, por lo que sería interesante que en un futuro probáramos a realizar una eliminación de outliers, para mejorar la precisión del algoritmo.

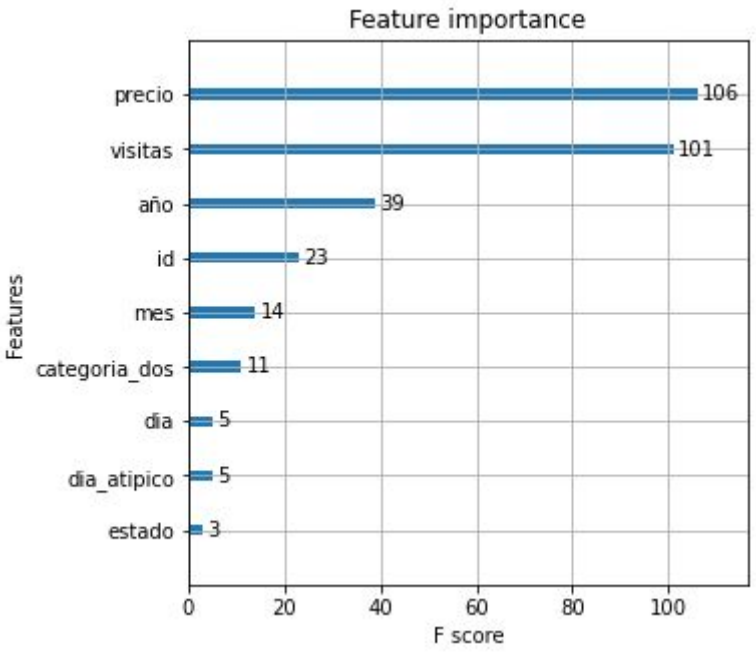


A su vez, hemos comprobado que existe baja correlación entre entre las variables:



Mediante la matriz de correlación de las mismas.

En el script “prediccion.py”, también hemos realizado un estudio de la importancia de las features. Aunque no pertenezca al script de exploración, ya que habíamos manipulado ya las variables y habíamos entrenado el modelo, consideramos que es importante mencionar este pequeño estudio sobre las variables. Con esta investigación llegamos a la conclusión de que las variables más importantes eran el precio y las visitas.



Para terminar, comprobamos el porcentaje de nulos, se nos muestra un 21% de nulos en la variable *"antigüedad"*, por lo que decidimos eliminarla. A su vez, nos da un 65% de nulos en la variable *"precio"*, y 5844 nulos en la variable *"categoria_dos"*. El problema de estas dos últimas variables lo vamos a solucionar mediante la indicación propuesta en el enunciado y tal como está explicado en los comentarios del código, aplicando un forward y backward fill cogeremos los siguientes valores más cercanos.

3. Resumen y argumentación sobre la manipulación de variables

La variable *'precio'* ha sido modificada siguiendo las indicaciones que se nos mandaron desde Cajamar, aplicando un .fillna() con el método ffill y bbfill.

Por otra parte la variable 'categoria_dos' ha seguido la misma modificación (agrupando por 'id'), puesto que había valores nulos o con valor '-' en el set Estimar y gracias a la exploración se puede ver que para un mismo *'id'* se tiene un mismo valor de *'categoria_dos'*

Queríamos mencionar también la refactorización de algunos datos (variables como *"categoria_uno"* o *"estado"*) de distintos tipos (como *strings* o *object*), a tipo integer y float mediante *LabelEncoder* de la librería scikit-learn.
También recalcar la división de la fecha a 3 columnas distintas que han pasado a ser día, mes y año.

Por último, algunos valores de *'precio'* eran del tipo str y con una coma como separador, es por ello que los pasamos a float y como separador un punto.

Todas estas modificaciones fueron realizadas con el objetivo de poder entrenar nuestro modelo con XGBoost Regression, ya que necesitábamos todos los datos de tipo int o float.

4. Justificación de la selección del modelo

Como modelo predictivo hemos escogido el XGBoost, ya que es el que mejor resultados arrojaba en este artículo y menor cota de error obtenía siendo un problema también de predicción de ventas, por lo que vimos conveniente aplicar el mismo modelo.

Nos quedará pendiente para la mejora del algoritmo una eliminación de outliers, como se mencionó anteriormente, así como una normalización de los datos que quizás mejoren los resultados obtenidos durante este proyecto.

5. Bibliografía

https://www.aprendemachinelearning.com/analisis-exploratorio-de-datos-pandas-python/
https://www.kaggle.com/abonaplata/analisis-exploratorio-de-datos-con-python
https://towardsdatascience.com/5-machine-learning-techniques-for-sales-forecasting-598e4984b109
https://pandas.pydata.org/docs/
https://scikit-learn.org/stable/