DATAHLON

Cajamar University Hack

Reto Atmira Stock Prediction

Equipo Enver:
Simón Sánchez, David Méndez y Enrique Botía

Estructura del reporte:

1. Proceso
2. Técnicas aplicadas

Proceso

Para este proyecto hemos usado python y las librerías pandas, matplotlib y sklearn como principales herramientas para el proyecto.

Preprocesamiento de los datos

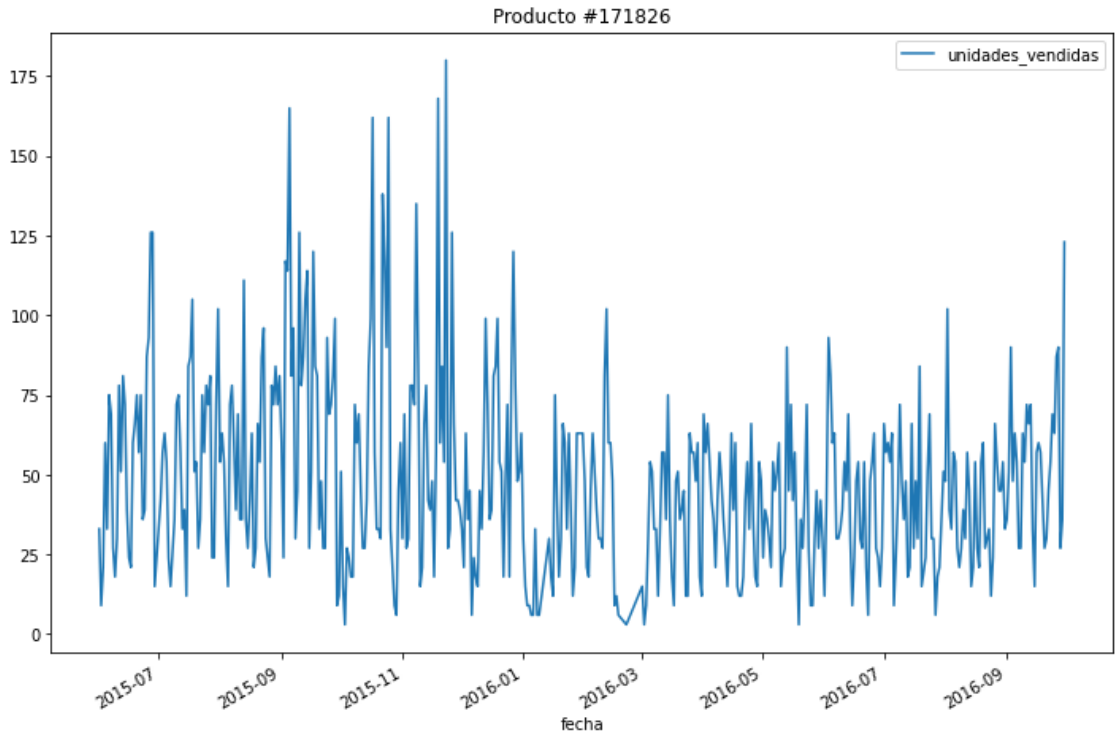
Tras inspeccionar y analizar detenidamente el conjunto de datos que se nos había proporcionado, comparando los datos que se debían modelizar con los que se requería predecir, hemos proseguido a realizar las siguientes transformaciones para poder aplicar luego las técnicas del aprendizaje automático con eficacia:

1. **Importe adecuado de los datos:** Hemos cargado los datos estableciendo los tipos apropiados para cada variable.
2. **Sustitución de los valores nulos**: como se indicaba en las bases del concurso, hemos completado los valores nulos de la variable “precio” con los valores temporalmente más cercanos para cada artículo. También hemos eliminado las instancias que contenían valores nulos en la variable “categoria_2”, puesto que representaban un porcentaje muy reducido en el conjunto de datos a modelar.
3. **Eliminación de la variable “antigüedad”**: nuestro análisis de los datos ha determinado que la variable “antigüedad” no es relevante para la estimación de la demanda, por lo que hemos eliminado esta variable.
4. **Eliminación de productos innecesarios en los datos a modelar**: hemos observado que hay productos en los datos a modelar no incluidos en el conjunto de datos a estimar . Como esto añadiría ruido en nuestro modelo, hemos eliminado estos productos para disminuir el error en nuestros resultados.
5. **Eliminación de las instancias duplicadas:** existen instancias idénticas en los datos a modelar. Las hemos eliminado para disminuir el ruido en nuestro modelo y el trabajo computacional del proceso.

6. **Eliminación de las instancias en estado “rotura”**: hemos observado que no hay datos con la variable “estado” con el valor “rotura” en los datos a estimar, por lo que hemos eliminado todas las instancias de los datos a modelar que tenían este valor en la variable “estado”.
7. **Utilización de variables binarias:** hemos creado variables binarias con la información extraída de las variables categóricas “estado”, “categoria_1”.

Después de este preprocesamiento de los datos, hemos vuelto a analizar el conjunto de datos obtenido, para poder determinar qué nuevas variables podríamos crear para minimizar el error en nuestros resultados.

Como el problema trata de hacer forecasting de las ventas de días futuros y los datos consisten de una serie de tiempo por cada producto, el método que utilizamos para estimación es Multi-step-ahead forecasting, que consiste en entrenar de manera independiente un modelo por cada día a estimar. Este método es tolerante a errores de predicción para días muy futuros, como tenemos que predecir 3 meses creemos que es el adecuado.



Para implementar el método hemos construido un conjunto de entrenamiento conteniendo variables temporales para estimar un solo día por cada uno de los 92 días que buscamos predecir del conjunto a estimar (92 modelos distintos).

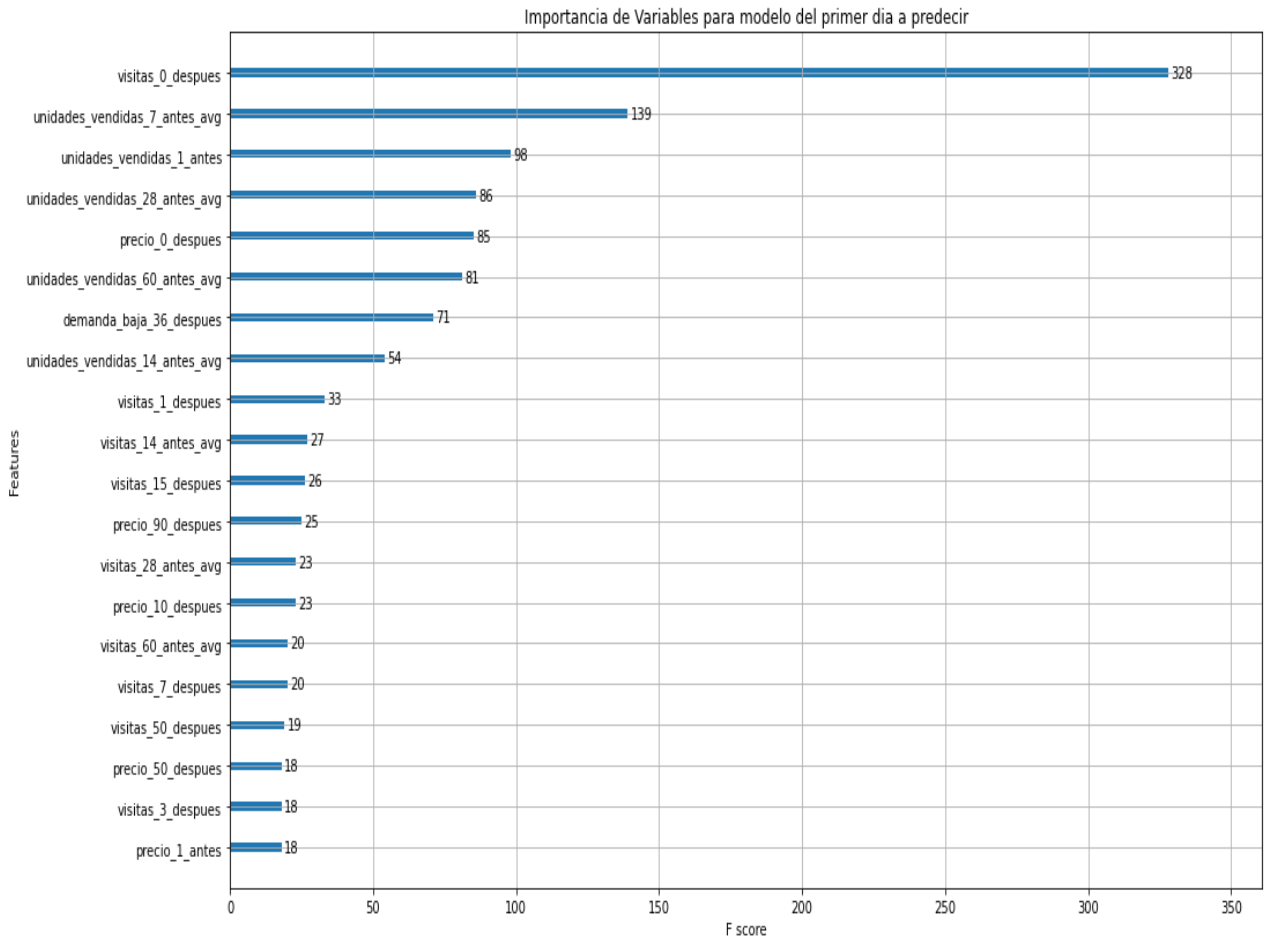
Hemos considerado en este conjunto de datos los siguientes tipos de variables:

- 1. Nuevas variables basadas en los valores futuros de *visitas*, *campaña* y *precio* por cada uno de los días a predecir.
- 2. Nuevas variables sacando la media de variables como *visitas*, *unidades_vendidas* y *precio* de n días pasados al día a predecir para varias n.
- 3. Nuevas variables basadas en pasados días de campaña (Como el número de veces que el producto ha estado en campaña en los últimos n días para varias n) y también basado en futuros días de campaña (Si/no habrá días de campaña en el futuro cercano por ejemplo).
- 4. Variables dummies para las variables categóricas como *dia_atipico* y *categoria_uno.*

Para crear estas variables hemos implementado una función de python que sirve como pipeline para generar estas variables de un dataset dado.

Técnicas aplicadas

La técnica de aprendizaje automático que hemos decidido aplicar para generar cada uno de los 92 modelos es “Gradient Boosting”. Hemos probado con distintos modelos y este ha sido el que ha obtenido mejores resultados a lo largo de varios experimentos. Los parámetros que hemos cogido para cada modelo han sido aquellos que hemos observado que producen un buen resultado de evaluación. En el futuro un método más rígido con cross-validation puede ser usado para tratar de tunear mejor estos parámetros y quizás mejorar los resultados.


Importancia de Variables para modelo del primer dia a predecir
visitas_0_despues
unidades_vendidas_7_antes_avg
unidades_vendidas_1_antes
unidades_vendidas_28_antes_avg
precio_0_despues
unidades_vendidas_60_antes_avg
demanda_baja_36_despues
unidades_vendidas_14_antes_avg
visitas_1_despues
visitas_14_antes_avg
visitas_15_despues
precio_90_despues
visitas_28_antes_avg
precio_10_despues
visitas_60_antes_avg
visitas_7_despues
visitas_50_despues
precio_50_despues
visitas_3_despues
precio_1_antes
328
139
98
86
85
81
71
54
33
27
26
25
23
23
20
20
19
18
18
18
Features
0
50
100
150
200
250
300
350
F score