TFM DSMarket

Grupo 3

29/04/2024

Roberto Reig Torres
Berta Gómez Puigbarraca

Máster en Data Science

Tutora: Raquel Revilla

Índice

Introducción

Este proyecto que presentamos tratará el caso de uso de DSMarket. DSMarket es una pequeña cadena de centros comerciales en los Estados Unidos, la cual se ha sumado a la transformación digital con el fin de implementar cambios en sus procesos de empresa.

En este proyecto la misión es, mediante el análisis de los datos que dispone la empresa, mejorar los enfoques y los procesos que afectan a la gestión diaria de la empresa, para conseguir más beneficios y mayor eficiencia a nivel global.

Para empezar, el análisis de centrará en las tres ciudades grandes de Nueva York, Boston y Filadelfia. Se nos proporcionan datos de ventas en estas tres ciudades, junto con datos temporales y de producto, con los cuales deberemos realizar los diferentes análisis.

El primer análisis que realizamos se centra en saber qué tipología de productos son los más vendidos e identificar los patrones de venta: según cada ciudad, momento temporal, precios, cantidades.

Para ello, el primer paso es juntar todas las bases de datos proporcionadas en una sola, limpiar los datos y así conseguir una única base de datos apta para poder trabajar y estudiar. De esta base de datos podremos sacar los primeros indicadores y patrones de ventas para entender el negocio.

El siguiente paso será identificar grupos de productos y tiendas similares, para poder así evaluar los rendimientos de los procesos y campañas realizados. Mediante el Clustering de los datos, conseguiremos segmentarlos en grupos que nos aportarán más información útil para aplicar en los procesos.

La siguiente tarea consistía en realizar una predicción de ventas. Mediante técnicas de Machine Learning, lanzaremos diferentes modelos con el fin de obtener una predicción válida. Utilizamos en primera instancia, algoritmos de regresión. Los cuales nos dieron resultados de predicción con valor de error elevados.

Posteriormente cambiamos a metodología Time Series utilizando el modelo xgb.XGBRegressor. Los resultados de las predicciones no fueron buenos debido a la poca dimensionalidad de los datos que disponemos. Al realizar el rebalanceo, conseguimos obtener una predicción y disminuir el error.

Finalmente se exponen las conclusiones.

Objetivos

Uno de los objetivos es adaptar la empresa a la transformación digital para poder utilizar los datos que dispone la empresa a lo largo del tiempo y que no se estaba aprovechando su potencial.

Se quiere conseguir, a través de estos datos, obtener predicciones de ventas y mejorar la gestión de las diferentes áreas de la empresa.

Metodología empleada

Una vez obtenemos todas las bases de datos de la empresa, repartidos en tres csv distintos, debemos unificar todas las bases de datos en una única para su análisis. Partiendo de la premisa de respetar el número de registros de ventas.

Para ello, la nueva tabla obtenida necesitamos que contenga las fechas, las ventas y los precios en una única base de datos. Para luego poder aplicar los modelos predictivos y los clustering.

Después de un primer análisis de la base de datos se plasma en un dashboard de PowerBI. Mediante el cual sintetizamos la información más relevante para el negocio.

Para general el clustering, creamos nuevas variables que nos aporten información relevante extra para que el algoritmo pueda segmentar mejor. Tras ajustar diferentes parámetros y combinaciones, guiamos al algoritmo a una combinación de variables en la que vemos que segmenta por grupos con unas características definidas. El algoritmo utilizado para los clusterings es el KMeans.

Este proceso lo realizamos en dos clusterings, el primero agrupando los productos, y el segundo agrupando las tiendas. Las variables utilizadas en cada clustering son ad hoc para cada caso.

A continuación proporcionamos una lista detallando las nuevas variables creadas para el clúster de Ítems:

'Revenue': es sell_price por la cantidad.

'top_items': los 15 ítems que más se venden.

'less_items',: los 15 ítems que menos se venden.

'sell_price_evolution',: variación del precio de cada ítem a lo largo del tiempo.

'max_Revenue': Ítems por mayores ingresos.

'min_Revenue',: Ítems por menores ingresos.

'mean_Revenue': media de los ingresos.

'month_sale',: Mes de la venta.

'year_sale',: Año de la venta.

'sales_total_evolution': variación de las ventas de cada ítem a lo largo del tiempo.

'sales_total': Ventas totales.

'time_since_last_sale': tiempo desde la última venta.

Variables creadas para el clústering por Tiendas:

'Revenue',

'top_store': Las 3 tienda con más ventas

'less_store': Las 3 tiendas con menos ventas.

'sales_store_category_sum': ventas de cada tienda por categoría de productos.

'revenue_store_category_mean': ingresos medios por categoría de cada tienda.

'sales_store_region_sum': ingresos de cada región en cada tienda.

'revenue_store_region_mean': ingresos medios por región y tienda.

'sell_price_evolution_mean': variación media del precio de venta.

'max_Revenue': ingresos máximos por tienda.

'min_Revenue': ingresos mínimos por tienda.

'mean_Revenue': promedio de ingresos por tienda.

'month_sale': mes de venta.

'year_sale': año de venta.

'sales_total_evolution': variación de las ventas en el tiempo.

'sales_total': ventas totales.

'time_since_last_sale': tiempo desde la última venta.

Para la predicción de las ventas, decidimos utilizar un enfoque de machine learning supervisado de regresión. Ya que disponíamos de los datos etiquetados y de un target concreto. Utilizamos diferentes algoritmos como modelos a entrenar:

- RandomForestRegressor
- DecisionTreeRegressor
- GradientBoosting
- LinearRegression

Tras analizar los resultados obtenidos, decidimos probar otro enfoque y plantearlo como un problema de Forecasting Time Series para hacer la predicción de ventas.

Debido a la distribución de los datos en el tiempo, y viendo los gráficos de la evolución, decidimos utilizar el algoritmo xgb.XGBRegressor.

Resultados y Conclusiones

Los resultados obtenidos de los modelos de Machine Learning fueron los siguientes.

MSE for Random Forest depth3: train 0.7074, test 0.693, val 1.547

MSE for DecissionTree: train 0.7544, test 0.7486, val 1.5906

MSE for GradienBoosting-n_estimators80: train 0.511, test 0.5344, val 1.3492

MSE for Linear Regression: train 0.8986, test 0.8947, val 0.8903

Al tratarse de un dataset con pocos registros (30490) y al hacer las particiones en validación, se quedan apenas 90000 registros, lo que es un poco justo para hacer predicciones y por eso es posible que haga que el error en las predicciones sea bastante alto. También destacar que el dataset está desbalanceado ya que el número de registros con el tiempo ha decrecido de forma drástica.

En RandomForest y DecisionTree tienen un rendimiento parecido, es posible que estén sobre ajustando ligeramente los datos ya que el MSE de validación es significativamente más alto.

El GradienBoosting parece tener el MSE más bajo tanto en train como en test, lo que indica que podría ser el modelo con mejor rendimiento para la generalización. Aunque en validación parece que sobre ajusta un poco.

En el LinearRegression vemos que tiene el MSE de entrenamiento y de prueba más cercano, lo que sugiere que podría estar subajustando los datos.

Las predicciones obtenidas fueron las siguientes:

item	Target	Prediction-RF	error-RF	squared_error-RF	root_squared_error-RF
SUPERMARKET_3_090	12.431226	8.560694	3.870532	14.981016	3.870532
SUPERMARKET_3_090	12.431226	8.560694	3.870532	14.981016	3.870532
SUPERMARKET_3_090	12.431226	8.560694	3.870532	14.981016	3.870532
SUPERMARKET_3_090	11.691047	8.560694	3.130352	9.799104	3.130352
SUPERMARKET_3_090	11.691047	8.560694	3.130352	9.799104	3.130352

Observamos que al modelo le cuesta predecir los valores altos. Decidimos probar a rebalancerlo para que prediga valores más altos.

Las nuevas métricas resultantes fueron:

MSE for Random Forest depth3: train 0.7827, test 0.7635, val 1.7352

MSE for DecissionTree: train 0.8204, test 0.8033, val 1.8193

MSE for GradienBoosting-n_estimators80: train 0.4408, test 0.511, val 1.4516

MSE for Linear Regression: train 0.9405, test 0.9071, val 0.9138

Y las nuevas predicciones obtenidas fueron las siguientes:

item	Target	Prediction-Random Forest depth3	error-Random Forest depth3	squared_error-Random Forest depth3	root_squared_error-Random Forest depth3	Prediction-DecissionTree	error-DecissionTree	squared_error-DecissionTree	root_squared_error-DecissionTree	Prediction-GradienBoosting-n_estimators80	error-GradienBoosting-n_estimators80
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.377429	2.541781
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.640798	2.278413
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.377429	2.541781
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.377429	2.541781
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.640798	2.278413
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.640798	2.278413
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.377429	2.541781
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.377429	2.541781
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.640798	2.278413
SUPERMARKET_3_586	11.91921	9.063197	2.856013	8.156809	2.856013	9.12796	2.79125	7.791077	2.79125	9.640798	2.278413

Aunque no mejoran las métricas, vemos que ahora predice valores más altos.

Los resultados obtenidos del modelo de Series Temporales fueron los siguientes que presentamos a continuación:

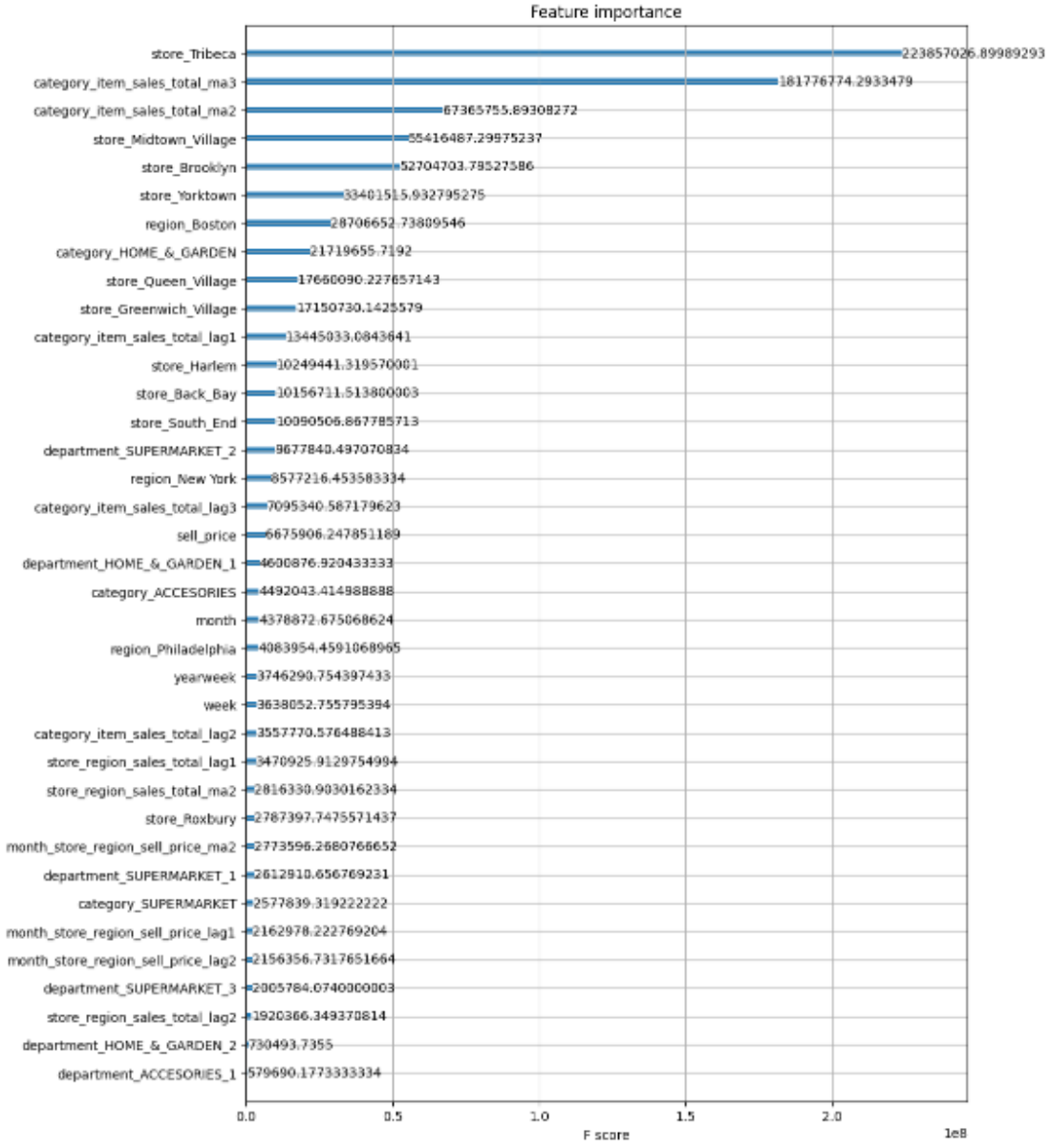
En un principio, el algoritmo no conseguía realizar la predicción al no tener un conjunto de datos suficientemente grande. Por esa razón rebalanceamos y al hacerlo obtenemos estas métricas:

Train RMSE: 237_187

Validation RMSE: 211_343

- Observamos que el algoritmo tiene más dificultad en predecir los valores más altos.
- Dado que la escala de valores de los últimos años es pequeña, el ERMS es un poco alto a pesar del rebalanceo.
- Sin embargo, en el primer año la escala de valores es bastante mayor y por tanto, allí la dimensión del error ya no es tan significativa.

Las Features Importance del modelo Time Series han sido:



Observamos que las variables que hemos creado que tienen que ver con el pasado (LAG) y las de medias móviles, son significativas para el modelo.

Tras los resultados obtenidos vemos que la obtención de datos es primordial para disponer de una base de datos representativa suficiente. Para poder luego aplicar modelos predictivos de forma eficiente.