FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 2 — Machine Learning

[75.06/95.58] Organización de Datos
Curso Martinelli
Primer cuatrimestre de 2024

Alumno:	DELLISOLA, Florencia
Número de padrón:	109897
Email:	fdellisola@fi.uba.ar

Alumno:	PELLICIARI, Agustín
Número de padrón:	108172
Email:	apelliciari@fi.uba.ar

Índice

1. Introducción **2**

2. Análisis Exploratorio de Datos **2**
2.1. Cantidad total de productos vendidos por mes 2
2.2. Distribución de los ingresos por año 4
2.3. Correlación entre los datos a analizar 5
2.4. Las 10 categorías que mas productos venden 6
2.5. Outliers 7

3. Modelos de Machine Learning **9**
3.1. Regresión Lineal 10
3.2. K-Nearest Neighbours 10
3.3. LightGBM 11

4. Referencias **13**

1. Introducción

Este informe presenta un análisis exploratorio de datos y el proceso de implementación de dos modelos de Machine Learning sobre la variable objetivo. El objetivo es predecir la cantidad de ventas que se realizarán durante el siguiente mes de una empresa de software de Rusia. Para esto, se seleccionaron los modelos Regresion Lineal (como baseline), LightGBM y KNN (K-Nearest Neighbors).

2. Análisis Exploratorio de Datos

2.1. Cantidad total de productos vendidos por mes

Para darnos una idea de como fue variando la cantidad de productos vendidos a lo largo de los meses y como debería ser, aproximadamente, el valor de la variable que debemos calcular, decidimos visualizar la información en los siguientes gráficos:

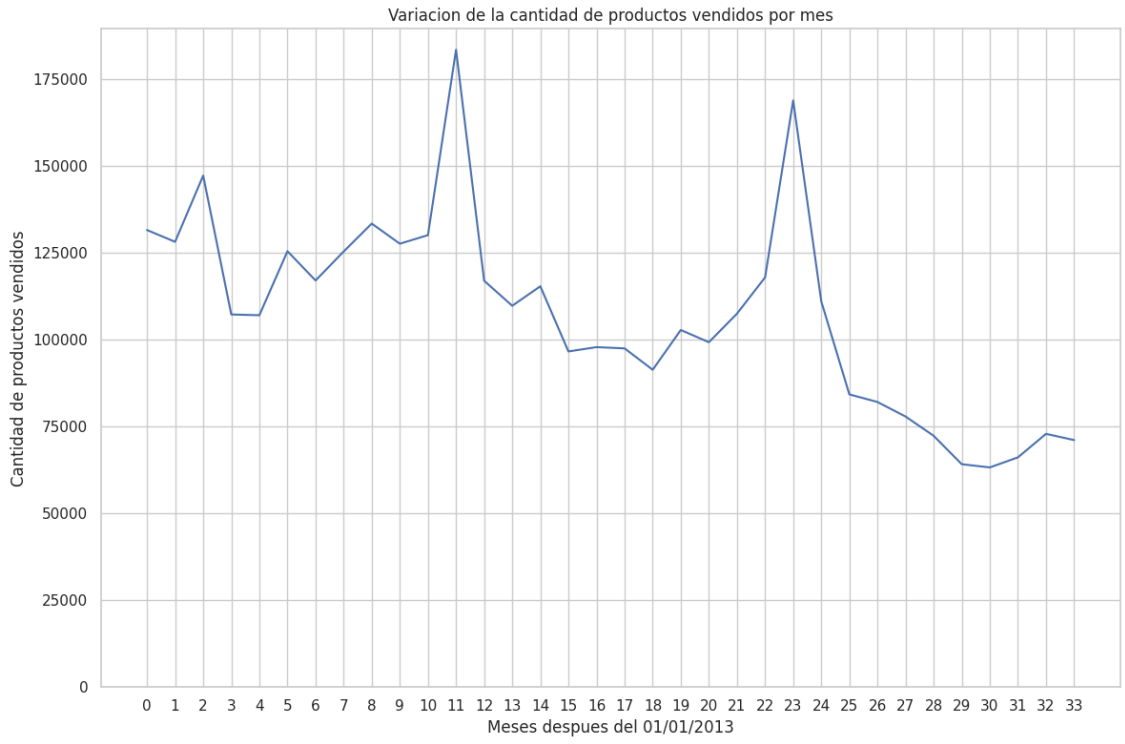

Figura 1: Line Plot.

En esta primera visualización se puede observar una visión unificada del comportamiento de las cantidades de productos vendidos a lo largo de 33 meses, comenzando desde enero del año 2013 (0) y terminando octubre del 2015 (33).

Se puede apreciar que hay picos de ventas en el mes 11 y el 23 (Diciembre) y luego se nota una baja de productos vendidos hasta el último mes analizado.
Para obtener una mejor visualización acerca de la variación y realizar una comparación por año optamos por modificar el gráfico anterior y nos quedo el siguiente modelo:

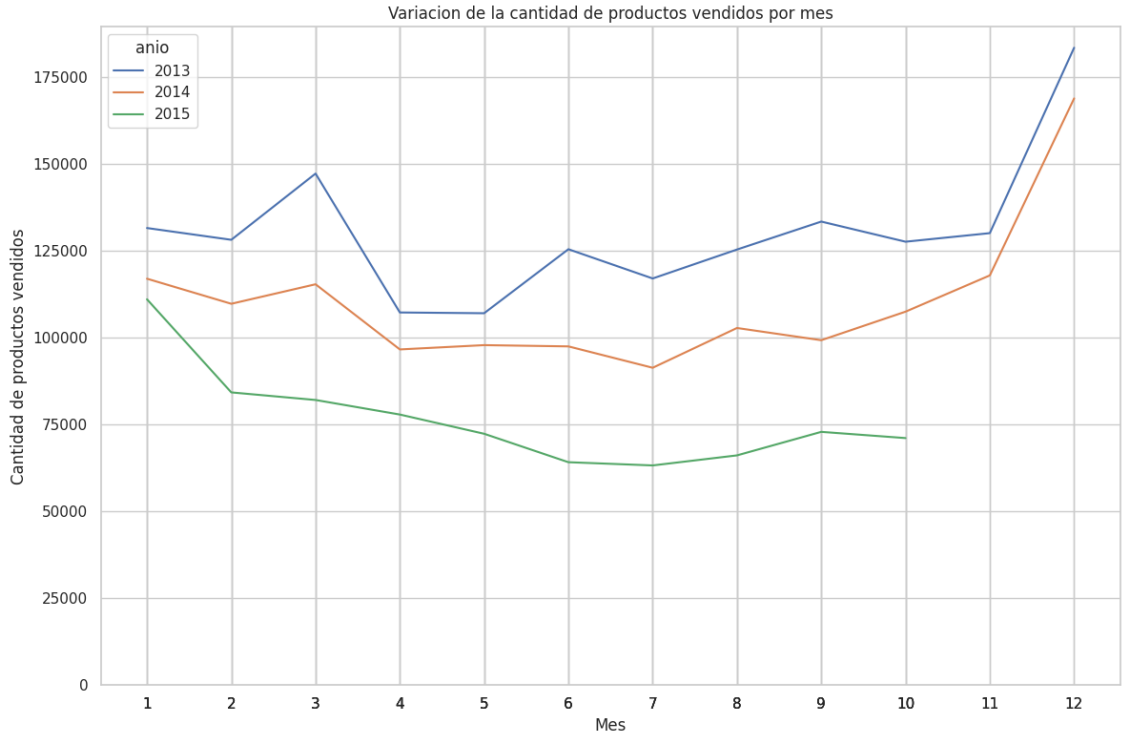

Figura 2: Line Plot.

A diferencia de la anterior visualización, los datos fueron agrupados por intervalos de años: 2013, 2014 y 2015; y los meses fueron acomodados para que se vean representados como en un calendario anual: Enero (1), Febrero (2), etc.

Se muestra un notable aumento de las ventas en el mes de diciembre en los años 2013 y 2014, aunque con una disminución de productos vendidos en este último año mencionado. Sin embargo, en el 2015 se percibe una menor cantidad inicial de ventas con una tendencia decreciente.

En este gráfico podemos observar que en el rango objetivo Octubre-Noviembre (10-11) en el año 2013 hubo un crecimiento, no muy exponencial, en relación a la cantidad de productos vendidos, sin embargo en el año siguiente la compañía tuvo un gran mes obteniendo uno de sus mejores registros. A partir de esto podemos inferir que la cantidad de productos vendidos en el mes y año objetivo tendrá un aumento en comparación a la cantidad vendida en el mes anterior, es decir, deberá ser mayor a 70000.

2.2. Distribución de los ingresos por año

Para modelar la distribución de ingresos por ventas realizadas por mes según cada año analizado elegimos el siguiente tipo de gráfico:

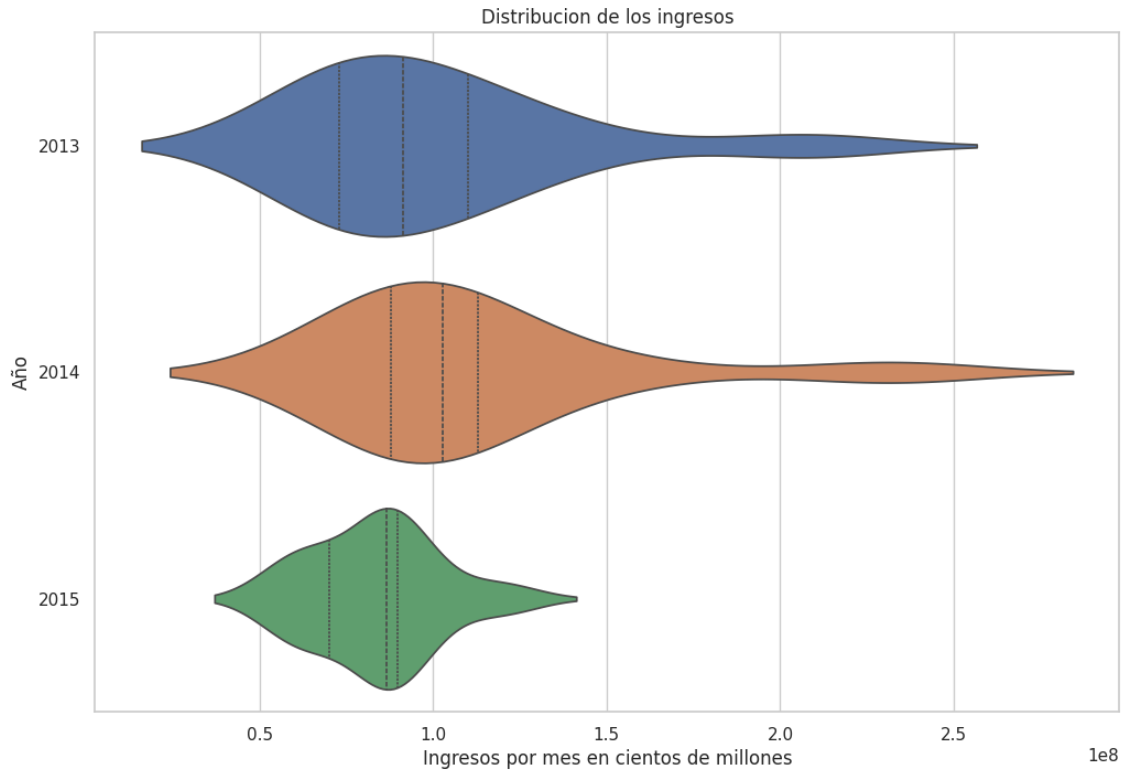

Figura 3: Violin Plot.

Podemos observar una similitud en la distribución en los años 2013 y 2014, a diferencia de los gráficos anteriormente presentados, con una pequeña mejoría en este ultimo ya que se puede apreciar como los ingresos en este año alcanzan valores mas altos, sobrepasando los 2.5 cientos de millones. También se puede ver como la mediana en este año es casi el doble en comparación al año anterior.

Basándonos en nuestro año objetivo, 2015, se puede visualizar que las ganancias estuvieron concentradas dentro de los valores 0.3 y 1.3 cientos de millones y es muy pobre en comparación a los otros años. Cabe destacar que en la distribución de este año faltan los últimos dos meses (Noviembre y Diciembre), creemos que a partir de la agregación de estos la distribución puede llegar a modificarse, alargando la forma del plot, ya que por lo general suelen ser los años donde la compañía obtiene sus mejores números.

2.3. Correlación entre los datos a analizar

Para darnos una idea de como se relacionan las variables del set de datos a utilizar decidimos implementar el siguiente tipo de gráfico:

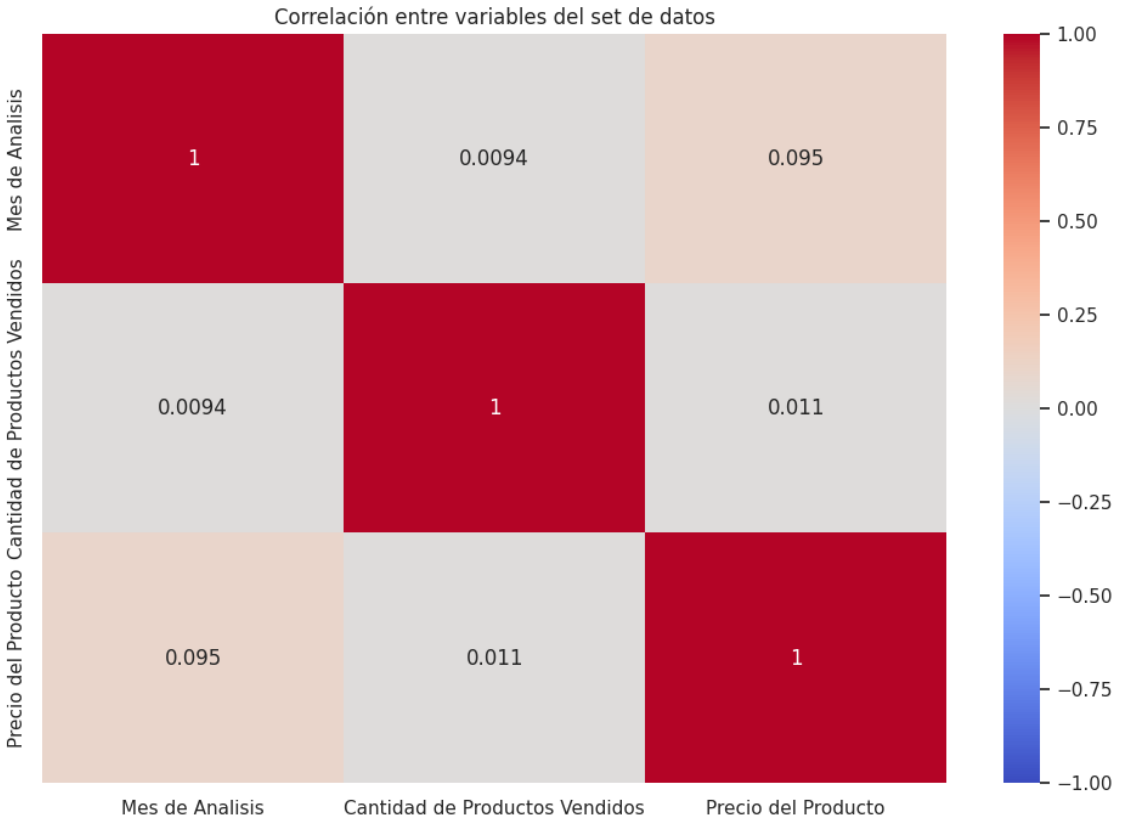

Figura 4: Heat Map.

En este plot, analizamos la relación entre el mes de análisis, la cantidad de productos vendidos por día y el precio de dicho producto. Contemplando con los datos analizados podemos ver que no existen relaciones temporales consistentes entre el mes de análisis, la cantidad de productos vendidos y el precio del producto.

Esta visualización nos da una idea que el resultado que podemos llegar a obtener no esta tan ligado a una linealidad o constancia en los datos de años anteriores. A pesar de esto, podemos observar una pequeña correlación del precio de producto y cantidad de productos vendidos con el mes de análisis.

2.4. Las 10 categorías que mas productos venden

A continuación, realizamos el análisis de las ventas por las diez principales categorías de productos con el siguiente plot:

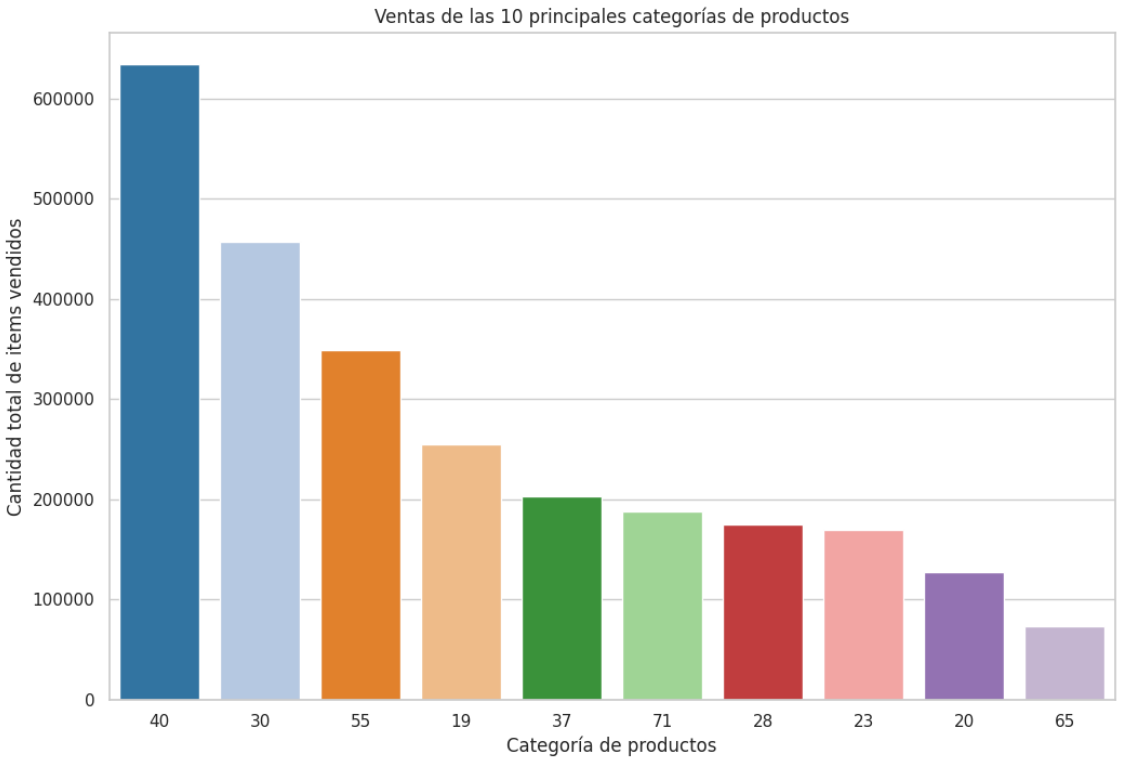

Figura 5: Bar Plot.

Este análisis proporciona una visualización sobre qué categoría de productos alcanza una mayor cantidad de ventas.

Para este gráfico, fusionamos los datos de ventas con los de ítems, agrupamos las ventas por categoría y por último tomamos las diez categorías de productos principales con mayor cantidad de ventas.
De esta manera podemos ver de forma descendente la cantidad total de ítems vendidos en cada una de estas categorías.

2.5. Outliers

Para buscar probables outliers en el set de datos optamos por el siguiente tipo de plot:

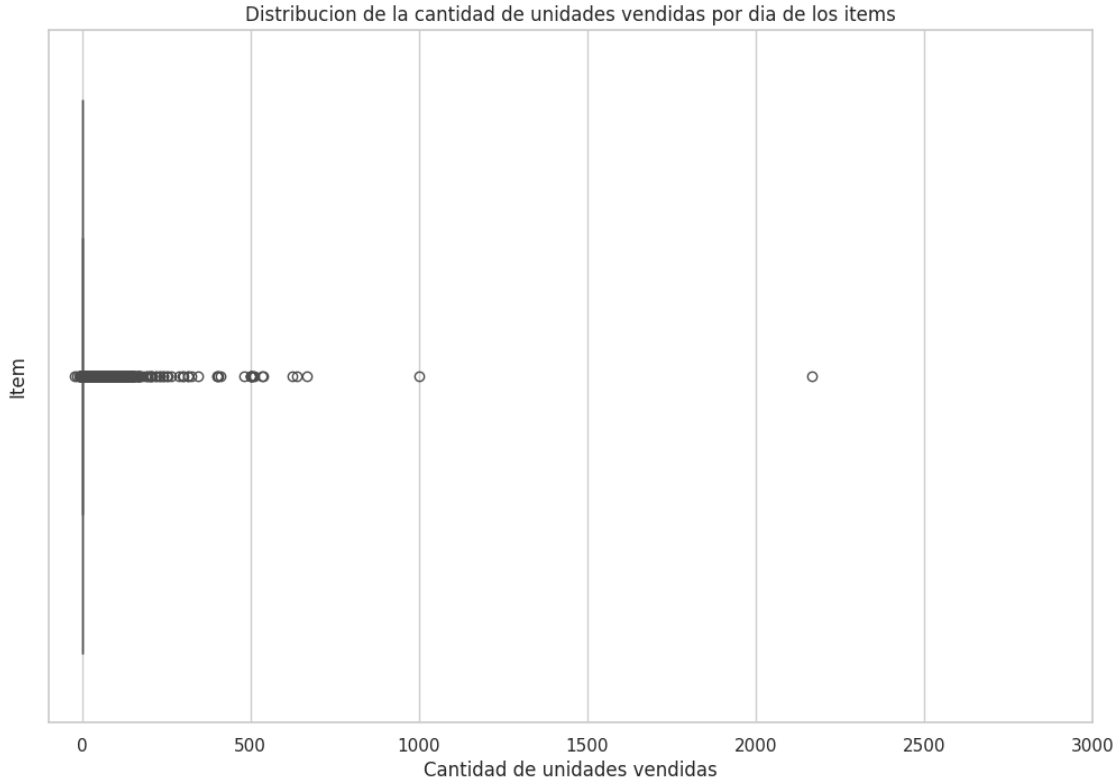

Figura 6: Box Plot.

En este primer caso, analizamos las cantidades vendidas por día de cada ítem. En el gráfico resultante, se observa que solo hay un ítem con un valor superior a 1000. Decidimos examinar este ítem en detalle, ya que 2500 unidades vendidas en un solo día es una cantidad extremadamente alta en comparación con las ventas de los demás ítems.
Luego de identificar el ítem correspondiente a ese valor, descubrimos que pertenecía a una empresa rusa dedicada a la entrega de paquetes. Procedimos a investigar si existían otros valores asociados a este mismo ítem. Al analizar y comparar las apariciones en diferentes días, concluimos que la cantidad de ventas no guardaba relación con los valores de este ítem específico. Por lo tanto, decidimos eliminarlo del conjunto de datos.

Asimismo, el gráfico muestra ítems con valores negativos. Tras revisar los comentarios de los participantes de la competición que trabajaron con el conjunto de datos, inferimos que los productos con valores negativos podrían corresponder a devoluciones.

Para el segundo gráfico, continuamos con el análisis de los ítems, pero en este caso modelamos la distribución de sus precios:

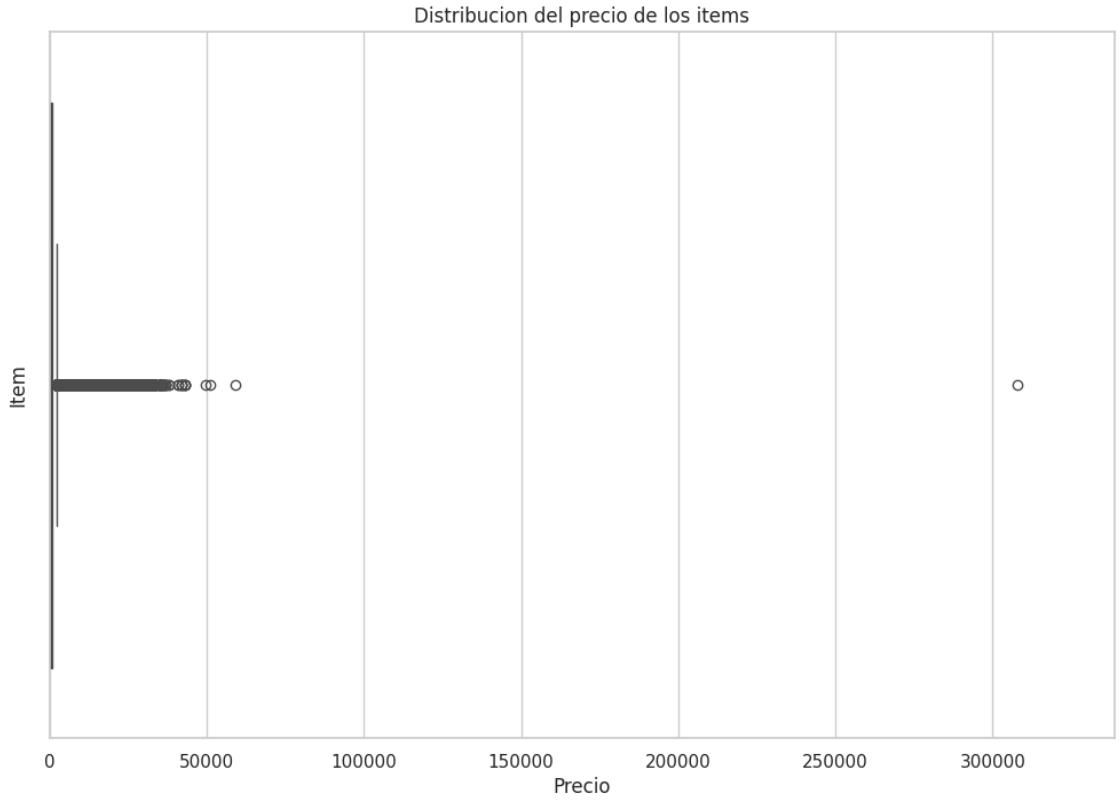

Figura 7: Box Plot.

Al examinar la distribución de precios de los ítems, encontramos nuevamente un valor extremo: un ítem con un precio superior a 300,000. Al traducirlo, descubrimos que correspondía a una empresa de software de escritorio cuyo nombre incluía un número (522) seguido de la palabra "personas". Intuimos que este valor podría representar el costo de instalación para esa cantidad de personas. Dado que no estaba relacionado con un servicio proporcionado por la compañía y no representaba al resto de los ítems, decidimos filtrar los datos eliminando este valor.

Por último, localizamos un ítem con un precio negativo. Investigamos si había otras ocurrencias de este ítem en el mismo mes, año (*date block num*) y tienda (*shop id*). Al encontrar dos ocurrencias de este mismo ítem, calculamos la mediana de los precios y la reemplazamos en el precio del ítem con valor negativo.

3. Modelos de Machine Learning

Antes del desarrollo de los modelos de Machine Learning, que luego explicaremos en detalle, realizamos lo siguiente:

- Preprocesamiento del set de datos

  Primero, como explicamos detalladamente en la sección del análisis exploratorio, filtramos los outliers y modificamos algunos valores que podrían causar problemas al realizar la predicción de la variable con el modelo.

  Luego, agrupamos cada par (*date block num*, *shop id* e *item id*) para obtener la cantidad de unidades vendidas de cada elemento en una tienda determinada durante un mes específico, ya que esta es la variable que debemos predecir.

- Separamos la variable a predecir: ***item cnt month***

- División del set de datos en set de entrenamiento y set de validación

  Para el de entrenamiento utilizamos un 80 por ciento del set de datos y para el de validación un 20 por ciento. Al ser un set de datos donde existe una linea temporal, en este caso, los datos de la compañía durante meses continuos, tuvimos que realizar la división del set de datos manualmente, para evitar el *'time travelling'*. En otras palabras, el set de entrenamiento tenia que tener los datos "mas viejos". Para esto calculamos el largo del set de datos y lo multiplicamos por 0.8, desde el comienzo hasta este valor se extendía el set de entrenamiento y desde donde terminaba el set de entrenamiento hasta el final del set de datos se extendía el set de validación. El set de testeo no hizo falta separarlo ya que tuvimos que utilizar uno de los archivos .csv que brindaba la pagina.

- Feature Engineering

  Realizamos la creación de variables de los set de datos que contenían información acerca de los productos y tiendas de la compañía.

  Primero analizamos el set de datos que contenía información acerca de las tiendas. Traduciendo algunos nombres intuimos que los nombres de las tiendas seguían un mismo patron: nombre de la ciudad de origen de la tienda + descripción de la tienda + nombre de la tienda. Realizamos un split para así quedarnos con el nombre de la ciudad y a partir de esto les realizamos un Label Encoding obteniendo una nueva variable para cada ciudad. Luego a partir de los nombres de las tiendas, encontramos que varias de estas se encontraban duplicadas, por lo que removimos una de estas tiendas de los sets de datos. Nos terminamos quedando con el id y la ciudad encodeada de las tiendas.

  Del set de datos que contenía información acerca de la categoría de los items obtuvimos la siguiente información. Nuevamente traduciendo los nombres de las categorías de los items encontramos el siguiente patron, la categoría de cada ítem estaba formada por: tipo ítem + '-' + subtipo ítem. Realizando un strip nos quedamos con el tipo del ítem y le aplicamos nuevamente un Label Encoding a cada uno de estos para así crear una nueva variable con el tipo del ítem encodeado. En el análisis del subtipo nos dimos cuenta de lo siguiente: algunos items no tenían subtipo. Por lo que decidimos que en caso de que no tuviese, le asignaríamos el nombre del tipo y los que tuviesen el subtipo. Luego de obtener los nombres de los subtipos definidos para cada ítem, aplicamos Label Encoding, encodeando cada uno de estos. Nos terminamos quedando con el id y el tipo y subtipos encodeados de las categorías.

Para finalizar con el análisis de los set de datos, en el que contenía información acerca de los items optamos por remover el nombre ya que no encontramos información relevante para utilizar dentro de nuestros modelos.

Luego de esto fuimos integrando cada uno de estos set de datos con las nuevas variables con el set de entrenamiento, el de validación y el de test.

A partir de la columna *date block num*, agregamos una nueva variable 'month' que contenia el numero de mes calendario al que hacia referencia esa columna. Esta nueva variable la creamos en el set de entrenamiento y el de validación.

Por ultimo en el set de datos de test removimos la columna de los id de cada par ya que no aportaba información útil. También, le agregamos una columna *date block num* con el numero del nuevo mes el cual realizamos la predicción de la variable (34). A partir de esta, como le aplicamos al set de entrenamiento y validación, le agregamos la columna 'month', en este caso, 11 (Noviembre).

3.1. Regresión Lineal

Para nuestro baseline, implementamos una Regresión Lineal, importándolo de la librería de *sklearn*, debido a su simplicidad y capacidad de ofrecer una referencia inicial sólida para la predicción de ventas. La regresión lineal intenta ajustar una línea recta que minimiza la suma de los errores al cuadrado entre las predicciones y los valores reales. Las métricas que utilizamos para la evaluación de modelo fueron el Error Medio Absoluto (MAE) y el Error Cuadrático Medio (MSE).

Luego de realizar el entrenamiento del modelo y la posterior prediccion con el set de validacion, a partir de este modelo obtuvimos los siguientes resultados:

- Error Absoluto Medio = 1.6995691002301792
- Error Cuadrático Medio = 84.9768197311301

Después de estos primeros resultados, probamos este modelo con el set de testeo. Posteriormente, integramos sus predicciones con el set de sample y las subimos a la plataforma Kaggle, donde se lleva a cabo la competición. La predicción del modelo obtuvo una puntuación de 2.42. Este valor se estableció como punto de partida, con el objetivo de reducirlo mediante la búsqueda de hiperparámetros en modelos posteriores.

3.2. K-Nearest Neighbours

Para el modelo de KNN (K-Nearest Neighbors), primero realizamos un escalado de datos utilizando 'StandardScaler' para asegurar que todas las características tengan una escala similar. Esto es importante en modelos de este tipo ya que el algoritmo calcula las distancias entre los features para determinar la cercanía de los vecinos.

Luego hicimos una búsqueda de hiperparámetros utilizando 'GridSearchCV' para poder encontrar los mejores parámetros para dicho modelo. Estos son:

- n neighbors: Cantidad de vecinos a tener en cuenta en la predicción. Este parámetro afecta la suavidad de las predicciones.
- weights: Esquema de ponderación de los vecinos. En este modelo utilizamos 'Uniform', donde todos los vecinos tienen el mismo peso, y 'Distance', donde los vecinos más cercanos tienen mayor influencia que los más lejanos.
- metric: Métrica de distancia para medir la cercanía entre los puntos. La que utilizamos las métricas que utilizamos fueron 'euclidean', 'manhattan' y 'mikowski'.

La optimización de hiperparámetros en KNN es fundamental para obtener el mejor rendimiento del modelo.
A partir de esto, probamos todas las combinaciones de hiperparámetros especificadas y usamos la validación cruzada para evaluar cada combinación.
Después de completar esta búsqueda, obtuvimos el mejor modelo encontrado (basado en la métrica RMSE).

Finalmente, realizamos las predicciones y evaluamos el rendimiento del modelo de KNN. El mejor modelo encontrado tuvo el siguiente resultado en el conjunto de validación:

- RMSE = 9.524705.

Por ultimo, probamos este modelo con el set de testeo. Integramos sus predicciones con el set de sample y las subimos a la plataforma Kaggle. La mejor predicción del modelo obtuvo una puntuación de 1.74, obteniendo una gran mejora en comparación al resultado obtenido en el baseline, a diferencia de esta ultima hecha con búsqueda de hiperparametros que obtuvo un puntaje de 3.41.

3.3. LightGBM

Para el segundo modelo implementamos LightGBM de la librería *lightgbm*. Este es un algoritmo de boosting basado en árboles de decisión que es eficiente en términos de tiempo y memoria, y que es capaz de manejar grandes volúmenes de datos con alta velocidad y precisión.

Para comenzar realizamos la búsqueda de hiperparametros. En este modelo utilizamos el enfoque Random-Search de la librería de sklearn. Pasándole los hiperparametros, el numero de iteraciones y el numero de divisiones en la validación cruzada por parametro, este enfoque se basa en probar k combinaciones aleatorias y elegir la mejor métrica de ese grupo. Algunos de los hiperparametros mas importantes que utilizamos fueron los siguientes:

- Numero maximo de hojas (*num leaves*): 200

  Este parametro es muy importante para captar las relaciones mas complejas dentro del set de datos. Sin embargo, si se utiliza un numero muy alto se puede generar un 'overfitting', es decir, un sobre ajuste de los datos.

- Fracciones de datos y caracteristicas (*bagging/feature fraction*): 0.5

  Estos parámetros fueron muy importantes para evitar el problema anteriormente planteado de sobre ajuste de datos. La fracción de datos es fundamental para reducir la varianza del modelo y mejorar la generalización. La fracción de características se encarga de controlar la complejidad del modelo al limitar la cantidad de características utilizadas en cada árbol.

Luego, realizamos el entrenamiento del modelo con los sets de entrenamiento y su posterior evaluacion de las predicciones del set de validación. En este caso los resultados obtenidos fueron los siguientes:

- RMSE: 9.220303

Para visualizar como se comportan y cuales fueron las features mas influyentes en el desarrollo del algoritmo de predicción modelamos lo siguiente:

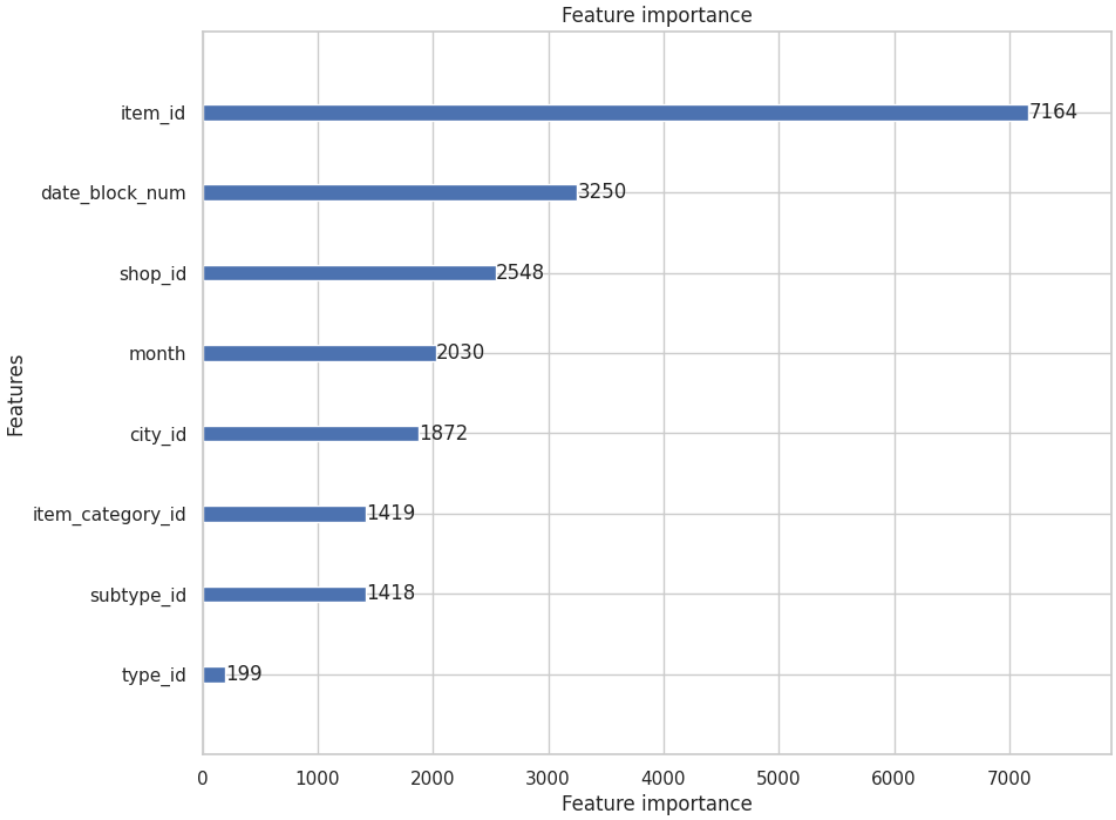

Figura 8: Importancia de las features.

Este gráfico muestra la importancia de las diferentes features en nuestro modelo de predicción. Las features están listadas en el eje Y y la importancia relativa de cada una está representada por la longitud de las barras en el eje X. De las nuevas features que agregamos, se puede apreciar como 'month' fue la que mas importancia tuvo. En el caso de la que menos relevancia tuvo fue 'type id' y se puede observar la gran diferencia de esta con 'subtype id' que obtuvo un mayor valor de importancia.

Para finalizar, probamos este modelo con el set de testeo. Integramos sus predicciones con el set de sample y las subimos a la plataforma Kaggle. La predicción del modelo obtuvo una puntuación de 2.38, obteniendo una mejora no tan significativa en comparación al baseline.

4. Referencias

- Dejamos el link del notebook donde estuvimos analizando el set de datos y donde se ubica la lógica de los gráficos y los modelos de Machine Learning implementados.
- Dejamos el link al vídeo que explica algunos de los detalles mas importantes del trabajo practico. También se encuentran algunos de los archivos csv que subimos a la competencia de Kaggle con las respectivas predicciones de los modelos utilizados en el trabajo practico.