

TP2 Organización de Datos

Alexis Güttlein Gareis - 104431

24/06/2024

Índice

1. Introducción	2
2. Análisis Exploratorio	3
2.1. Fechas con mayor volumen de ventas	3
2.2. Artículos más vendidos	4
2.3. Histórico de categorías más vendidas	6
2.4. Proyección de ventas	6
3. Baseline	8
4. Modelos	9
4.1. Separando train de validation	9
4.2. Análisis de Features	9
4.3. Modelo: Random Forest	10
4.4. Modelo: XGBoost	11
4.5. Competencia Kaggle	12
5. Conclusión	13
6. Corrección	14
7. Anexo	15

1. Introducción

El propósito del siguiente informe es lograr predecir ventas futuras, utilizando un conjunto de datos proporcionado por una empresa rusa, dentro del marco de una competencia de Kaggle.

Los registros de ventas proporcionados van desde enero de 2013 hasta octubre de 2015 y se buscará predecir, utilizando diferentes modelos, el total de productos vendidos en cada comercio para noviembre 2015.

El listado de comercios y productos varía ligeramente cada mes, por lo que parte del desafío es crear un modelo robusto que pueda manejar este tipo de situaciones. Esto puede ser de gran valor para que las empresas lo utilicen para acciones como gestionar sus recursos de manera más eficiente, implementar campañas de marketing, etc.

Para evaluar la eficacia de los modelos predictivos, se utilizará la métrica del Error Cuadrático Medio de Raíz (RMSE, por sus siglas en inglés). El RMSE es una métrica ampliamente utilizada en problemas de regresión, que mide la diferencia entre los valores predichos por el modelo y los valores reales observados. Un RMSE más bajo indica un modelo más preciso, que es capaz de realizar predicciones más cercanas a las ventas reales.

Este informe explicará el proceso de análisis y modelado de los datos. Inicialmente se deberá explorar y limpiar los datos para luego poder construir y validar los modelos utilizados.

2. Análisis Exploratorio

Se comenzaron a analizar los datos mediante diferentes gráficos para tener un panorama general de las ventas de la empresa.

2.1. Fechas con mayor volumen de ventas

Se analizaron las fechas en donde se realizaban más ventas para determinar si existe algún patrón que pudiera ayudar a mejorar las predicciones.

Se puede apreciar que las fechas en donde más artículos se venden son entre fines de Diciembre y principios de Enero. Vale la pena aclarar que, a diferencia de muchos países, en Rusia, la Navidad se celebra el 07 de Enero.

La única fecha que no aparece en este rango es la del 22 de Febrero. En este país se celebra el 'Día de los Defensores de la Patria', por lo que es muy común dar obsequios a quienes sirvieron en las fuerzas armadas.

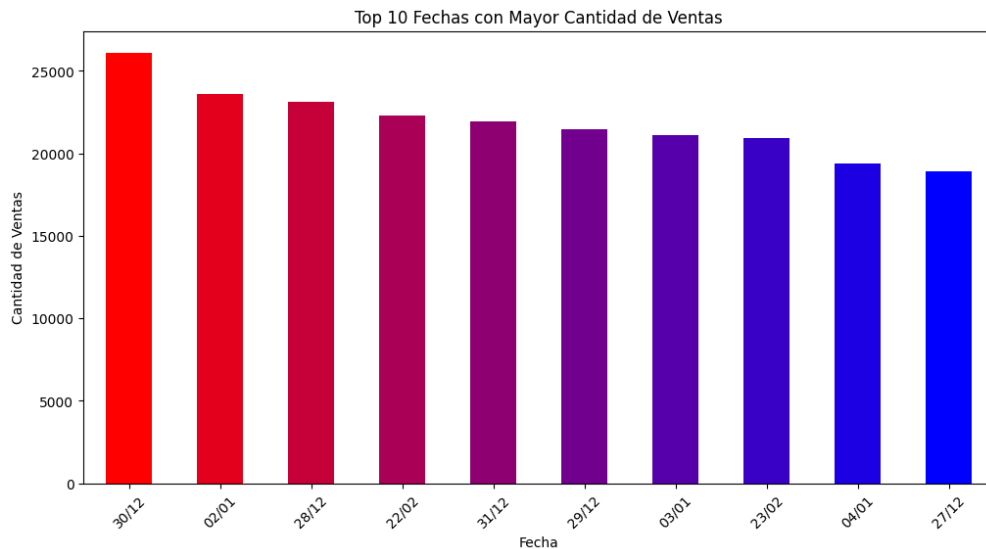


Figura 1: Como en la mayoría de los países, las fechas cercanas a Navidad es donde se realizan mayor cantidad de compras.

También se analizaron las ventas mensuales de cada comercio en donde, luego de Diciembre y Enero, se puede ver que Marzo es un mes con una cantidad considerable de ventas. Sorpresivamente la mediana de Marzo está por encima de la de Enero.

Se estima que la aparición de este último podría deberse a que es el mes en que se celebra el día de la mujer.

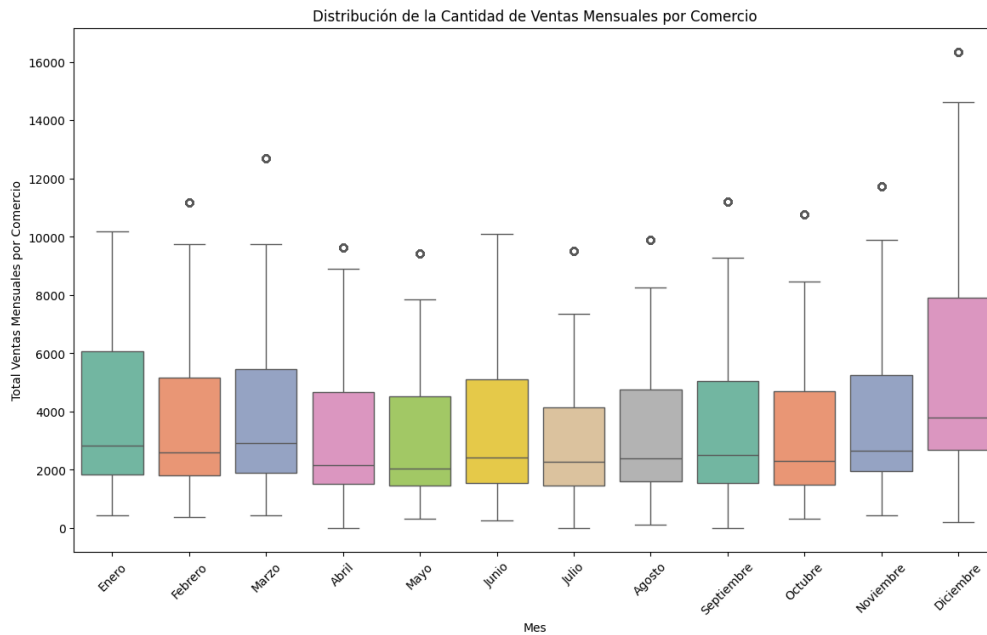


Figura 2: Diciembre y Enero son los meses con más ventas, seguido por Marzo.

2.2. Artículos más vendidos

En el histórico de los años analizados se detectó que las categorías más vendidas estuvieron relacionadas con DVD, PC y PlayStation. Tomando en cuenta que la Navidad es el momento de mayor ventas, tiene mucho sentido la elección de artículos para una empresa que comercializa artículos electrónicos.

Como los ítems relacionados con DVD fueron los más vendidos, se decidió analizar si existía algún momento específico de la semana en el que se vendían más. Con esto se podría detectar si existen días en los que generalmente la empresa otorga descuentos para incentivar las ventas.

Lamentablemente no se detectó lo planteado anteriormente pero sí se pudo ver que, sin importar el día de la semana, las ventas se concentraban en artículos de rangos específicos de precio.

La barrera idiomática dificulta determinar los artículos específicos de ese rango de precios pero podría suponerse que los ítems que tienen un precio de alrededor de 400 podrían corresponder a reproductores de DVD.

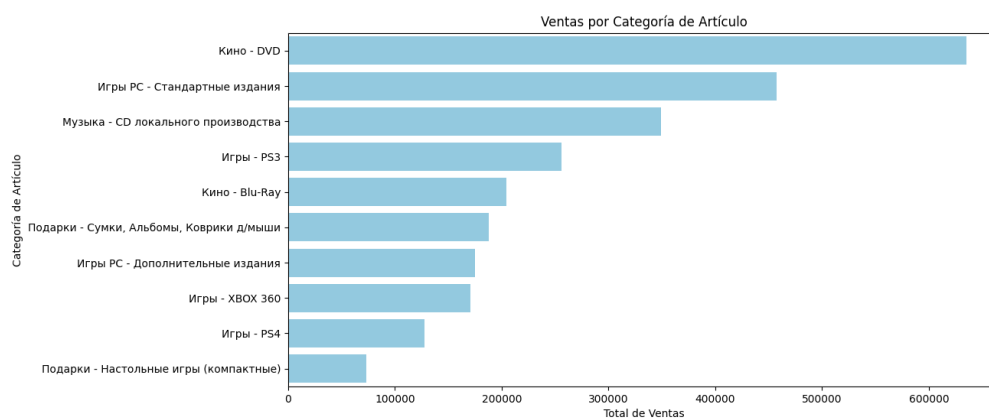


Figura 3: Los artículos relacionados con DVD fueron los más vendidos.

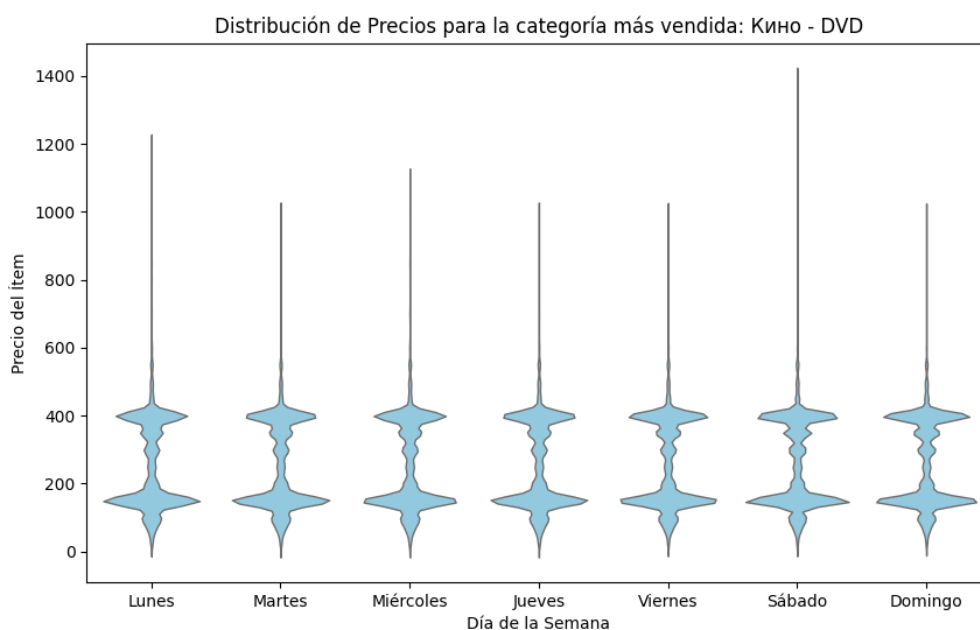


Figura 4: No se aprecia diferenciación de precios según el día de la semana.

2.3. Histórico de categorías más vendidas

Para dar un último análisis a las categorías más vendidas se generó un mapa de calor con las ventas que tuvieron a lo largo de los meses.

De aquí se puede extraer que los artículos relacionados con DVDs cada vez tienen menos relevancia aunque no pierden vigencia. Se puede suponer que sea cuestión de tiempo para que sean finalmente reemplazados por otras tecnologías.

Lentamente se puede ver cómo fueron ganando terreno los artículos relacionados con PS4 que, para la fecha actual, seguramente siguieron teniendo más relevancia.

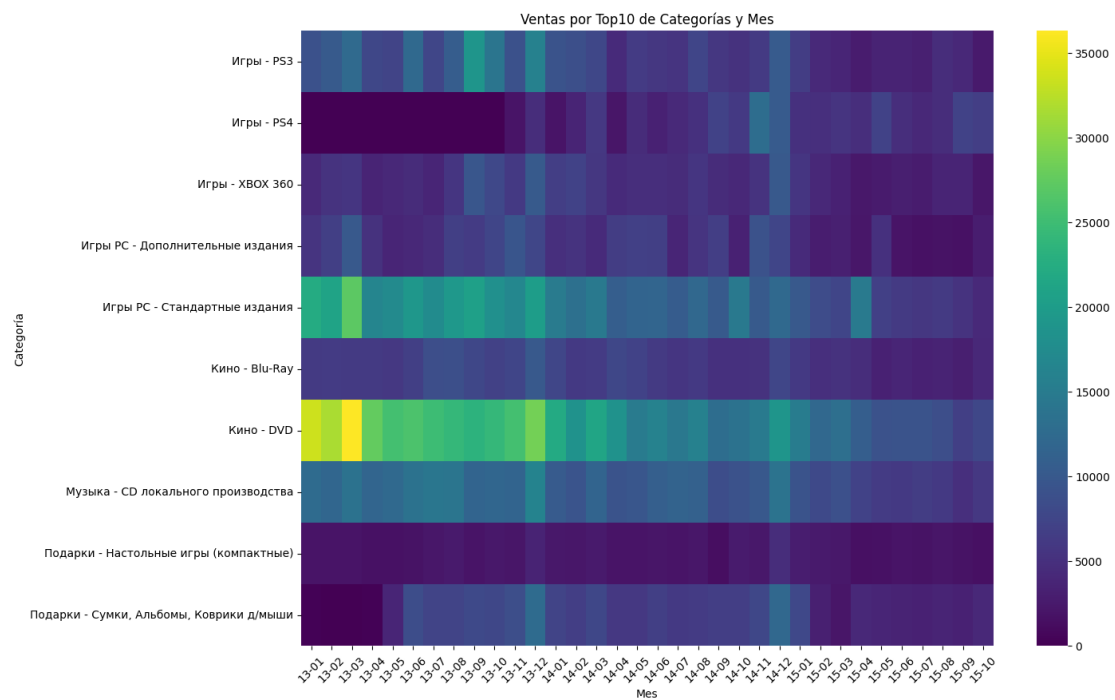


Figura 5: Las ventas de DVD van desapareciendo y lentamente dejan su lugar a nuevas tecnologías.

2.4. Proyección de ventas

Para concluir el análisis, se realizó una línea de tiempo con la cantidad de ventas mensuales.

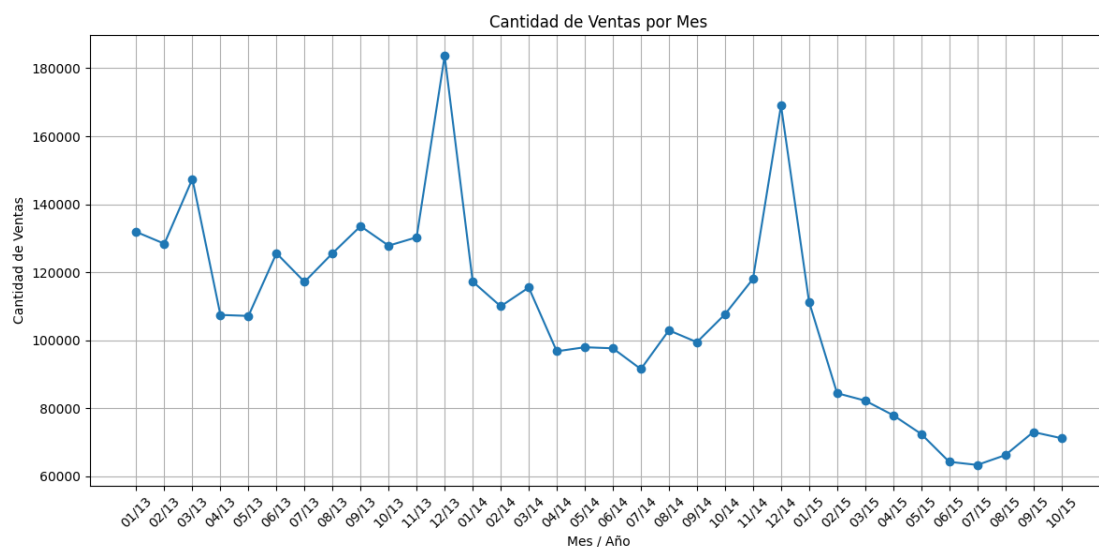


Figura 6: Las ventas tuvieron una importante baja en los últimos meses.

Se confirma que las ventas tienen su pico en Diciembre, pero lo más importante que se puede sacar de este gráfico es la tendencia que tiene esta línea. Se puede ver con claridad cómo bajaron considerablemente las ventas a lo largo de los últimos meses. Algo muy importante a tener en cuenta para realizar predicciones futuras.

3. Baseline

Como punto de partida para tener de referencia se realizó un Regresor Lineal y se utilizó para realizar una primer predicción.

Como era de esperarse para un modelo tan simple, el MSER obtenido fue de 4.67. Este tipo de modelo puede ser preciso si existe una relación lineal entre las variables.

En un caso como el analizado, donde los precios y volumen de ventas pueden variar constantemente de un comercio a otro y/o entre los diferentes meses y temporadas, es muy poco probable que un modelo como este sea suficiente para realizar predicciones.

De todas formas, de este primer intento se pudo rescatar información que será valiosa para los modelos no lineales que se implementarán luego:

- Se determinó que la variable ideal para separar el set de entrenamiento del de validación es "date_block_num", la cual se compone de valores secuenciales entre meses.
- El mes de 'corte' para separar los sets será el mes número 25, cuyos registros hasta este punto representan aproximadamente el 80 % del total de los datasets, dejando un 20 % para validación.

La ventaja de utilizar este modelo linear como prueba es que la ejecución del mismo es muy rápida, por lo que se pudieron hacer ajustes constantes hasta obtener el mejor resultado posible.

4. Modelos

Se decidió utilizar un modelo de Random Forest y otra de XGBoost para realizar las predicciones finales.

Antes de comenzar se combinaron todos los datasets que se disponían y se realizó una limpieza del dataset final en donde se decidieron eliminar unos pocos registros que contenían mayormente valores nulos y no hubieran realizado un aporte significativo a las predicciones. Así como también, luego de algunos análisis se eliminaron series cuyos datos no eran numéricos y no se logró encontrarles una cualidad que pudiese mejorar los resultados.

4.1. Separando train de validation

Una decisión muy importante para poder comenzar a entrenar modelos es la de seleccionar la manera de separar los datos de entrenamiento de los de validación.

Inicialmente se había optado por una separación arbitraria donde el 20% iba a ser para testear/validar resultados. El gran problema de esto es que, al separar registros al azar, podría suceder que los modelos se entrenen con datos futuros y 'predigan' datos pasados. Es por esto que se disidió hacer una separación estricta.

Afortunadamente no se necesitó generar un nuevo feature para esto ya que uno de los campos del dataset se compone de valores numéricos sucesivos que aumentan entre meses (donde 0 corresponde a enero 2013, 1 a febrero 2013,..., 33 a octubre 2015).

4.2. Análisis de Features

Features que no fueron tenidos en cuenta para los modelos:

- **date:** sólo se consideró útil extraer el mes de este campo considerando que podría llegar a ser de utilidad.
- **item_name:** cada item tiene un id, por lo que no se encontró sentido a utilizar este campo.
- **category_name:** similar al anterior, también se cuenta con un id para categorías. Sin embargo este campo fue importante para generar visualizaciones y entender las ventas.
- **shop_name:** como con los anteriores, tambien cuenta con un id asociado.

Nuevos Features creados para mejorar predicciones:

- **total_vtas_shop_mes:** determina el total de las ventas mensuales por comercio. resultó importante para el modelo XGBoost.
- **total_vtas_item_mes:** determina el total de las ventas mensuales de cada ítem. También fue relevante para las predicciones de XGBoost.

- **cambio_precio:** muestra si existió una diferencia de precio de un artículo de un mes a otro.
- **precio_prom_item_mes:** determina el precio de venta promedio mensual de cada tipo de artículo.
- **total_vtas_item_shop_mes:** determina el total de ventas mensuales por ítem, por comercio. Feature muy importante ya que las predicciones se deben hacer en base a la combinación ítem-comercio.
- **total_vtas_cat_mes:** determina el total de ventas mensuales según la categoría de un artículo.
- **cant_vtas_mes:** cantidad de ventas totales mensuales. Importante para determinar que las mismas estan bajando.

Los datos tenidos en cuenta para los diferentes modelos fueron normalizados antes de ser entrenados para poder así mejorar su rendimiento y precisión.

El set a predecir contenía nuevos valores para ítems y comercios, algo para lo que el modelo debía estar preparado, por lo que, al momento de predecir, se detectaron muchos valores nulos. Se optó por rellenarlos, en la mayor parte de los casos, por el valor más común para una categoría o un ítem, o por la media de los mismos.

4.3. Modelo: Random Forest

Con este modelo se logró obtener el menor MSER, aunque las predicciones obtenidas no fueron las mejores para la competencia.

Como es un modelo que requiere mayor cantidad de tiempo para entrenar y el dataset tiene un volumen importante de registros, se priorizó mantener la menor cantidad de árboles posible para que no afecte el rendimiento.

El mejor resultado se obtuvo utilizando, para entrenar, 10 árboles y un valor de 30 de profundidad máxima de hoja, entre otros hiper-parámetros que pueden encontrarse en el notebook.

En cuanto a la importancia de features, el campo '**total_vtas_item_shop_mes**' fue el de más peso, seguido de '**item_price**'. El resto de los features no tuvieron tanta relevancia en el resultado obtenido.

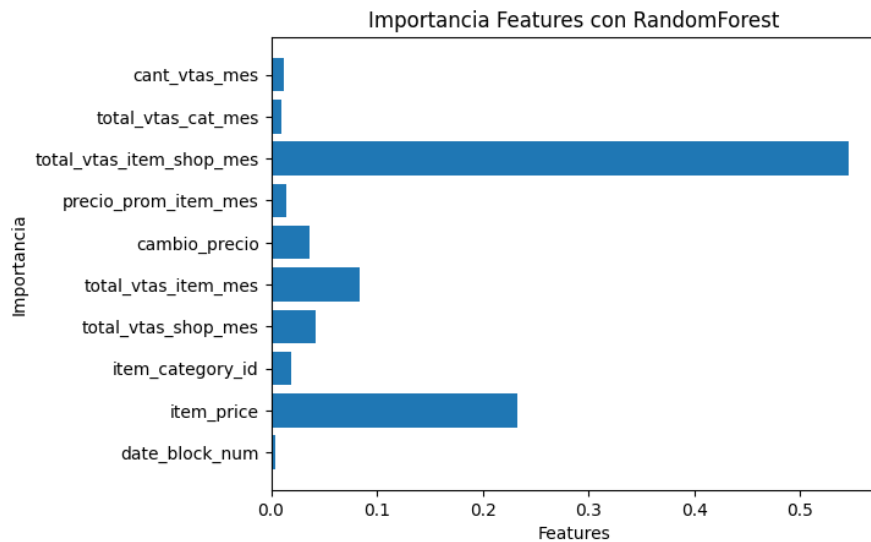


Figura 7: Importancia de features para el modelo de Random Forest.

4.4. Modelo: XGBoost

Este fue el modelo con el que se obtuvo mejor resultado en la competencia.

Al tener un procesamiento más veloz que otros modelos, como el de Random Forest, se pudo ajustar libremente la cantidad de árboles a utilizar hasta encontrar el número ideal, sin tener que considerar tanto el rendimiento.

Se detectó que el mejor resultado se obtenía utilizando 100 árboles y un valor de 5 de profundidad máxima de hoja, entre otros hiper-parámetros que pueden encontrarse en el notebook.

Nuevamente el feature más importante resultó ser '**total_vtas_item_shop_mes**'. A diferencia del Random Forest, el modelo tuvo una alta consideración para otros campos adicionales como 'total_vtas_item_mes' y 'total_vtas_shop_mes'.

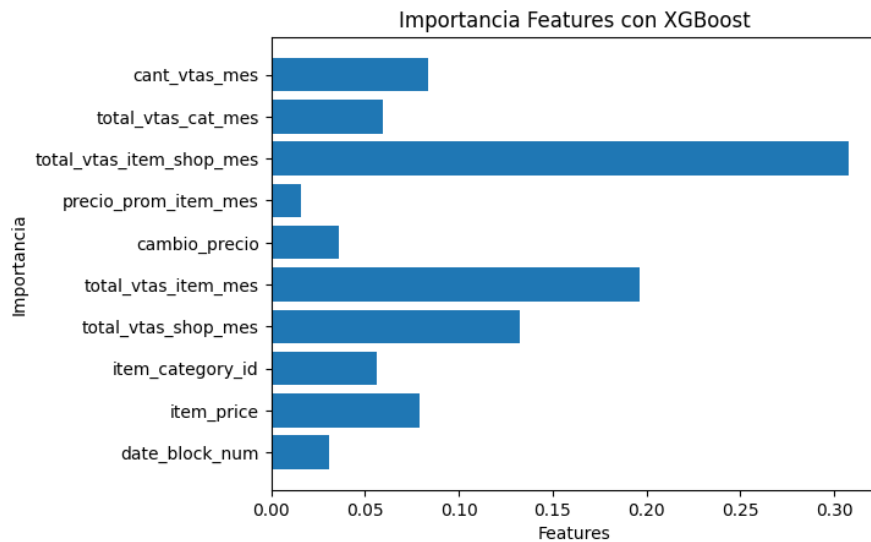


Figura 8: Importancia de features para el modelo de XGBoost.

4.5. Competencia Kaggle

Los resultados obtenidos en la competencia estuvieron muy lejos de ser los ideales, pero de todas formas se puede destacar el avance que se logró de un modelo a otro.

Iniciando por el **Regresor Lineal** que había obtenido un score de 162.444,77 (muy lejos del 0,73 obtenido por el líder). Era de esperar un mal resultado para este modelo en este tipo de dataset.

El siguiente submitido fue el **Random Forest** con un score de 1.090,53. Una mejora importante en cuanto a su predecesor, pero todavía un valor demasiado alto para poder considerarlo fiable al momento de predecir ventas futuras.

El mejor resultado se logró con el modelo de **XGBoost** con el cual se logró un score de 79,33. Si se compara con los mejores resultados de la competencia sigue estando lejos de ser eficiente, pero se puede considerar de forma positiva al ver que tuvo un gran avance en cuanto a las predicciones anteriores.

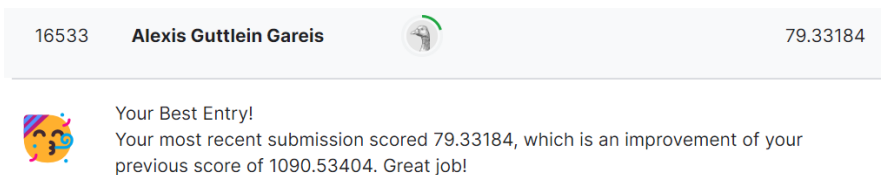


Figura 9: Mejor resultado obtenido en la competencia.

5. Conclusión

Está claro que los modelos elegidos no son los ideales o, en todo caso, se deben realizar ajustes en los mismos para mejorarlos.

Considerando que ambos modelos se basaron principalmente en nuevos features creados a partir de los existentes, se puede considerar que las mejoras podrían lograrse a partir de nuevos features y/o es posible que los que se agregaron puedan optimizarse de alguna manera.

También se debe analizar si la manera de rellenar los valores de los datos nulos, tanto del set original como del set a predecir, fue la mejor o si se debería modificar la estrategia elegida.

6. Corrección

Buscando mejorar los resultados se realizaron las siguientes modificaciones:

- Los set de train y validation se separaron antes de agregar nuevos features. Anteriormente esto se hacía luego del feature engineering lo que podía provocar overfitting.
- La variable 'month' fue incluida en el entrenamiento pero esta vez se realizó one hot encoding sobre la misma.
- Se eliminaron features que incluían el target y se agregaron otros que relacionaban el resto de los campos.
- Se agregó feature que relaciona los campos que posee el dataset de test.
- En el modelo de XGBoost se utilizó RandomizedSearch y cross validation para mejorar la selección de hiperparámetros.
- Se intentó utilizar el mismo método anterior en el Random Forest pero los tiempos de procesamiento eran demasiado altos, por lo que se optó por ajustar manualmente los hiperparámetros con el riesgo de no encontrar los óptimos.
- Se logró mejorar notablemente el resultado en la competencia aunque aún está lejos de lo esperado.


Alexis Guttlein Gareis		29.79000
-------------------------------	---	----------

Figura 10: Mejor resultado obtenido en la competencia.

7. Anexo

Link al repositorio: [Github](#)

Link al notebook: [Google Colab](#)

Link a los Dataset de la competencia: [Kaggle](#)

Link al [Video](#) explicativo