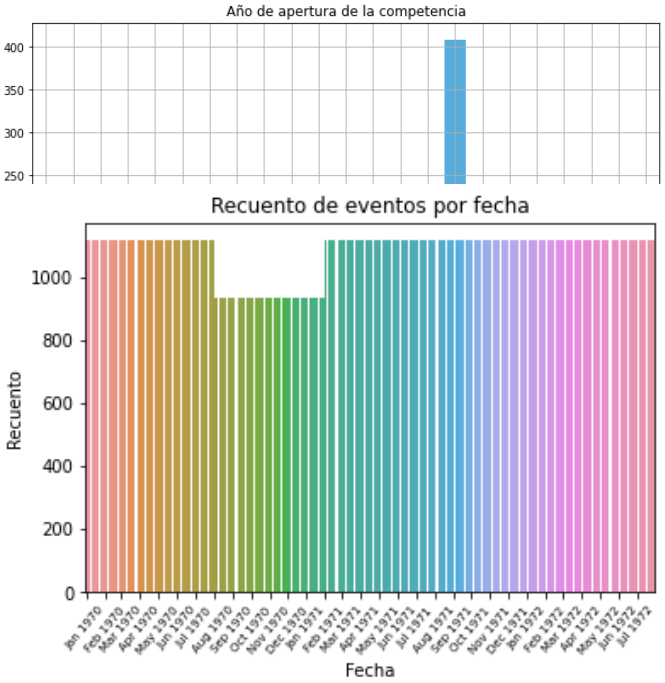
Introducción

En este trabajo, detallo el análisis que realicé de la base de datos Rossman tomada del desafío de Kaggle "Rossman store sales" en 2015. Se trata de una cadena de droguerías con más de 3000 sedes y con presencia en 7 países europeos. En esta competencia busqué predecir las ventas de la empresa en Alemania minimizando el error cuadrático medio porcentual (RMSPE) en base al historial de ventas de sus sedes entre 2013 y 2015. Mi objetivo en este trabajo es resolver el desafío y luego plantear y resolver un modelo donde busco clasificar correctamente si una tienda está participando de una promoción o no en un día dado, utilizando las variables disponibles en los conjuntos de datos proporcionados.

Análisis exploratorio

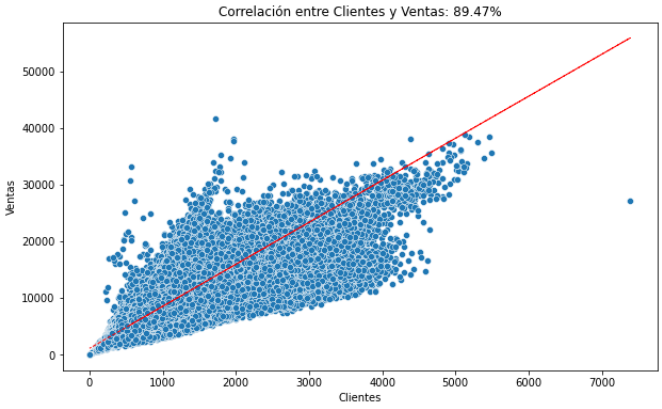
En primer lugar, realicé una exploración exhaustiva de las bases de datos de train, test y store. Comencé observando la cantidad y estructura de los datos que se encuentran en el dataset store. Observé las variables categóricas y sus subniveles presentes en el dataset: 'Storetype', 'Promo2', 'PromoInterval' y 'Assortment'. En particular, encontré una mayor presencia del tipo de tienda 'a' y 'd' en el dataset.

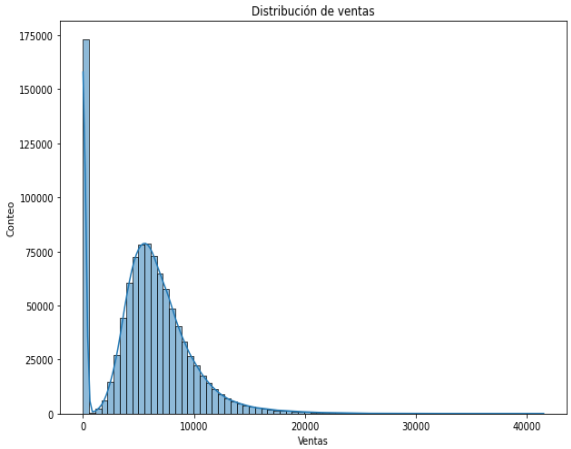
Además, exploré la variable 'PromoInterval', comprendida por 3 subniveles con los meses en los que la promoción se encuentra activa para cualquier año dado. La categoría que cuenta con mayor cantidad promociones es 'Jan, Apr, Jul, Oct'. Mientras tanto, analizando la variable 'Assortment' vi una menor presencia del assortment extra ('b').



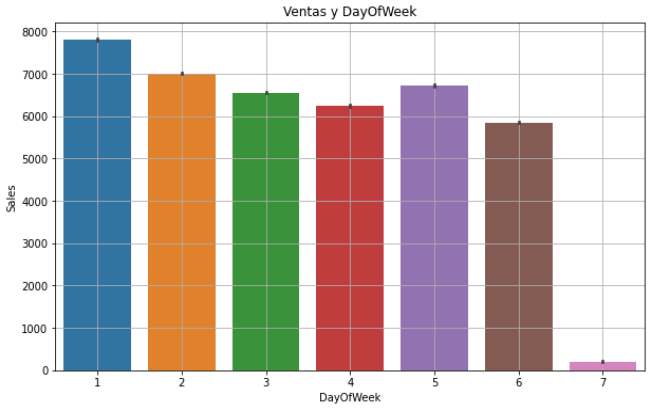
Asimismo, analicé en qué años abrió la competencia de las 1115 stores en el dataset. El primer año observado es en el 1900 y el año que presenta mayor frecuencia de apertura es el 2008. Continué observando la cantidad y estructura de los datos en el conjunto de train. Realicé un recuento de eventos presentes en función de su fecha y observé la falta de información para una serie de meses. Investigando en profundidad el tema, descubrí que el host de la competencia comentó que había sido un error de carga de su parte y que no era significativo a la hora del análisis. Decidí igualmente tenerlo en cuenta a la hora de realizar featuring engineering. La variable categórica 'State Holiday' me indica si la observación fue tomada en un día que corresponde a un feriado estatal. En su mayoría las observaciones fueron tomadas en días no-feriado y, entre aquellos datos que corresponden a un feriado Observo una mayor presencia del subnivel 'a', 'public holiday'. Al realizar la exploración de las variables numéricas, empecé observando la correlación entre la cantidad de clientes y las ventas. Observé, como era esperado, una correlación alta, de 89%. En este mismo gráfico observé que la cantidad de clientes oscila entre 0 y 5000 y las ventas entre $0 y $40000.

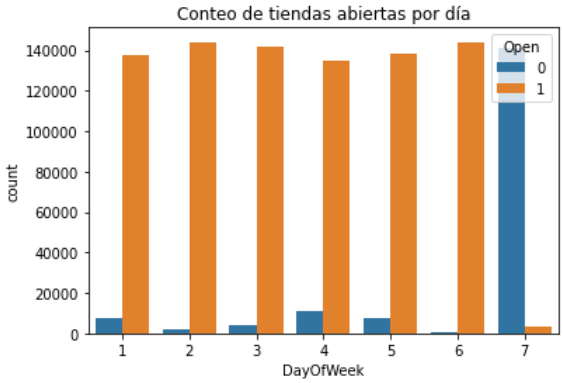
Luego, grafiqué las ventas para comprender mejor su distribución, observé la presencia de aproximadamente 175 mil valores de no-venta ($0) y una forma que se asemeja a una distribución normal con la mayoría de las observaciones correspondiendo a valores por debajo de $10000.





Por otro lado, analizando la variable 'Dayofweek' observé que el día con mayor conteo de ventas es el lunes y el de menor es el domingo, que coincide con lo que Observo en el conteo de tiendas abiertas por día: el domingo muchas tiendas se encuentran cerradas, me llamó la atención estas tiendas que abren los domingos y me motivó a analizar un poco más de cerca cuál era la causa. Lo más relevante resultó ser,





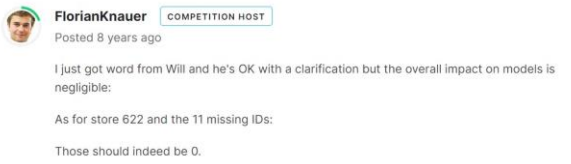
lo descubierto y analizado por otro participante de la competencia: dependiendo el estado donde estas se encuentran en Alemania (varían por región los feriados, y por ende las aperturas), sí figuraban cómo abiertas o no. Esto me dio el hincapié a incorporar un dataset detallando de qué estado es cada una, resultando en cierta medida relevante para algume modelos utilizados y detallados más adelante.

A su vez, como es de esperar, observé que no había ventas registradas para datos en los que la tienda se marcó como cerrada (Open = 0). Asimismo, se observa que se registran valores de venta en días de vacaciones de colegios.

Antes de continuar hacia la ingeniería de atributos analizo a través de gráficos de tipo box-plot los valores y dispersión que toman las variables numéricas presentes en el dataset. Particularmente, observo la presencia de valores más allá del rango intercuartil tanto para 'Customers' como para 'Sales', con la mayoría de sus valores encontrándose por debajo de 1.500 y 15.000 respectivamente.

Feature engineering

En esta etapa del análisis decidí abordar cada dataset por separado, haciendo el featuring en caso de ser necesario para cada variable presente. Cominezo por transformar las variables en Store, unificando el formato de la variable ‘StateHoliday’, descubierto durante el análisis exploratorio. Además, se imputan los valores faltantes en las variables ‘Open’ por 0, siguiendo las recomendaciones del host de la competencia en Kaggle (discutidas bajo un foro, adjunto imagen). También completo con 0 y ‘No Promo’ los valores faltantes en las variables ‘PromoInterval’, 'Promo2SinceYear' y ‘PromoInterval’ dado a que se debían a la ausencia de promociones para dichas observaciones, para que quede prolijo al luego definir One Hot Encoding.



A la hora de extraer información de las columnas existentes en el dataset creo a partir de la variable ‘date’, tomando la información de día, mes año, quarter, día de semana y la incluyo tanto al train como al test set con el fin de mejorar el poder predictivo del modelo.

También, me pareció interesante incorporar información de fuentes externas en busca de mejorar la precisión del modelo y para proporcionar nuevas variables útiles para la predicción. Analizo las opciones disponibles y decido incorporar un dataset de Clima (Weather) y Estados de Alemania (State).

La información de State fue sencilla de incorporar haciendo un join en “store”. En cambio, para incorporar weather requerí realizar varias transformaciones previas, entre estas, una función para convertir los nombres completos de los estados alemanes a sus abreviaturas y transformaciones en las columnas de temperatura, humedad y velocidad del viento para normalizar los valores e imputar valores faltantes. Para poder implementar lo más rápidamente posible los nuevos datasets me guié por una discusión que encontré en el foro de Kaggle.

A partir del entrenamiento del modelo podré observar y eliminar aquellas variables que aporten un menor poder de predicción para obtener un modelo más parsimonioso.

Sistema de validación del modelo

Antes de plantear el sistema de validación del modelo se plantean dos formas de dividir el conjunto de entrenamiento de modo tal que continua respetando la temporalidad de los datos. En primer lugar, se plantea una validación mediante holdout set que requirió reordenar los datos en función de año y mes y luego separar un 20% como conjunto de validación. Luego, investigando alternativas, decidí incorporar la validación a través de usar validación cruzada de series de tiempo a través de la función Time Series Split de la biblioteca de Scikit-Learn para realizar una validación que permita generalizar mejor a datos futuros. Time Series Split me asegura que la data utilizada para validar siempre es posterior a la que se utiliza para entrenar, creando folds con los que entrenar y validar, a medida que avanza el entrenamiento cada fold incluye aquellos índices con los que se entrenó previamente.

Model selection & selección de hiperparámetros

Entreno modelos de Random Forest y de gradient boosting a través de XG Boost y Light GBM. Se realizan varios entrenamientos, en los que analizo los resultados obtenidos para luego entrenar un nuevo modelo con propuestas de mejora.

La siguiente tabla resume los resultados obtenidos:

-	Modelo	Cross - Validation Hold out set (80/20)	Time Series Split	Data Externa	RMSPE	FEATURE IMPORTANCE	Comentarios
1	XG Boost	x			0.0078	-	Modelo base para comparación
2	XG Boost	x		x	0.0085	StateHoliday DayOfWeek Customers Open Assortment	Incorporación de data externa
3	XG Boost		x	x	0.0087	StateHoliday 0.690438 DayOfWeek 0.218509 Customers 0.068934 StoreType 0.015820	Implementación de Time Series Split
4	Light GBM		x	x	0.0235	CompetitionDistance Store CompetitionOpenSinceMonth CompetitionOpenSinceYear DayOfWeek Open StoreType	Se observa que “Customers” se encuentra con una gran correlación con la variable a predecir por lo que se dropea esta columna, y se observa un resultado más razonable en el RMSPE
5	Random Forest		x	x	0.0193	Open DayOfWeek Weekday StateHoliday Promo	
6	Random Forest (parámetros modificados)		x	x	0.029	Open DayOfWeek Weekday StateHoliday Promo	Modificación de parámetros
6.5	Random Forest c/ OOB Score		x	x	OOB Score: 0.033	-	Modificación de columnas eliminando filas de stores cerradas. Se bbserva que no mejora el score.
7	Light GBM (iteraciones = 300)		x	x	Score Kaggle: 0.59	Variables to keep: ["Store", "DayOfWeek", "Open", "CompetitionDistance", "CompetitionOpenSinceMonth", "CompetitionOpenSinceYear", "StoreType", "Day", "Promo2SinceWeek", "Promo2SinceYear", "Month", "Assortment", "State_NW", "PromoInterval", "State_SH", "Max_TemperatureC", "Weekday"]	Se utilizan únicamente las variables importantes.
	Light GBM (Iteraciones = 1000)		x	x	Score Kaggle: 0.47		Se utilizan únicamente las variables importantes. Aumento el número de iteraciones
8	Light GBM (Iteraciones = 5000)		x	x	Private Score Kaggle: 0.17 Public Score Kaggle: 016	Variables to keep: ["Store", "DayOfWeek", "Open", "StoreType" , "Day", "Month", "Assortment", "State_NW", "PromoInterval", "State_SH", "Max_TemperatureC", "Weekday", "Year", "Promo", "Mean_TemperatureC"]	Modifico las variables elegidas. Supongo que las variables de Store tienen poca capacidad para generalizar. Se priorizan las variables externas.
9	Random Forest (Iteraciones = 1000)		x	x	Private/Public Score Kaggle: 0.21	["Open", "DayOfWeek", "Weekday", "StateHoliday", "Promo", "CompetitionDistance", "Store" , "Day", "CompetitionOpenSinceYear", "CompetitionOpenSinceMonth", "Month", "StoreType", "Promo2SinceYear", "Assortment", "Promo2SinceWeek", "State_NW", "PromoInterval"]	Se ultilizan únicamente las variables importantes.
10	Random Forest (Iteraciones = 1000)		x	x	Private Score Kaggle: 0.23 Public Score Kaggle: 0.21	["Year","Open","DayOfWeek", "Weekday", "Promo", "CompetitionDistance", "Store" , "Day", "Month", "StoreType", "Promo2SinceYear", "Assortment", "Min_TemperatureC"]	Modifico las variables elegidas, de manera similar a la mejor corrida de Light GBM

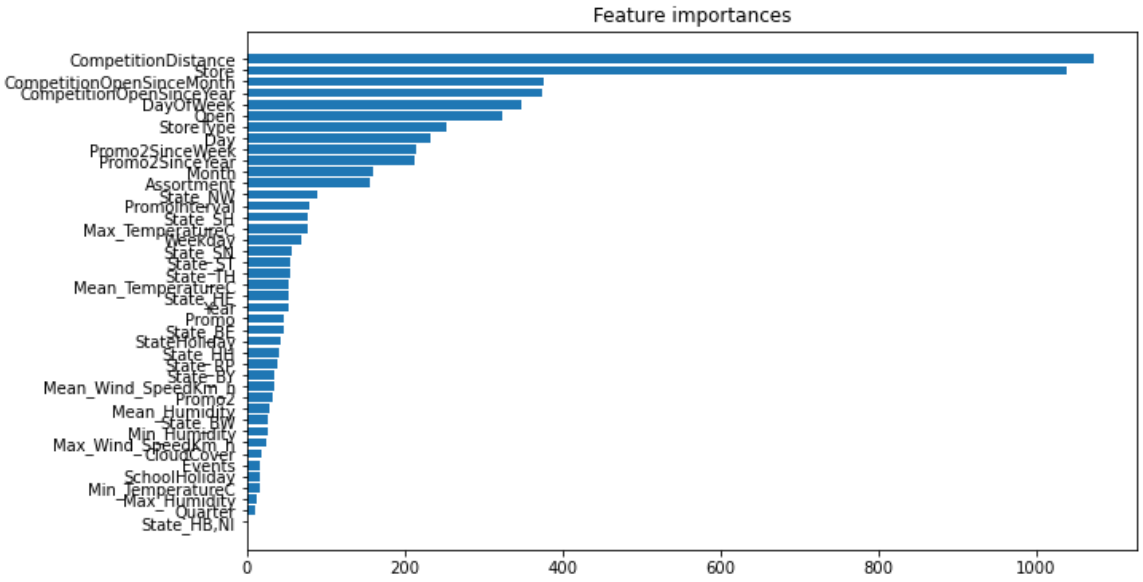
En una primera instancia realizo una primera corrida del modelo XG Boost donde no se busca evaluar los hiperparametros óptimos ni la importancia adjudicada a las variables, unicamente se utiliza para determinar un benchmark del cual partir para ir mejorando. En las posteriores evaluaciones de performance de los modelos se incluye un análisis de la importancia de las variables según el modelo entrenado.

Luego, con el objetivo de evaluar si la incorporación de datos externe presentaba una mejora en la performance del modelo corro un segundo modelo **XG Boost** con igual configuración que el anterior incorporando "**Weather**" y "**State**". Obtengo un RMSPE 0.07% mayor, pero decido mantener el uso de features externas dado que todavía me encuentro realizando modificaciones al modelo y aun falta evaluar la performance de esta data en los distintos modelos seleccionados.

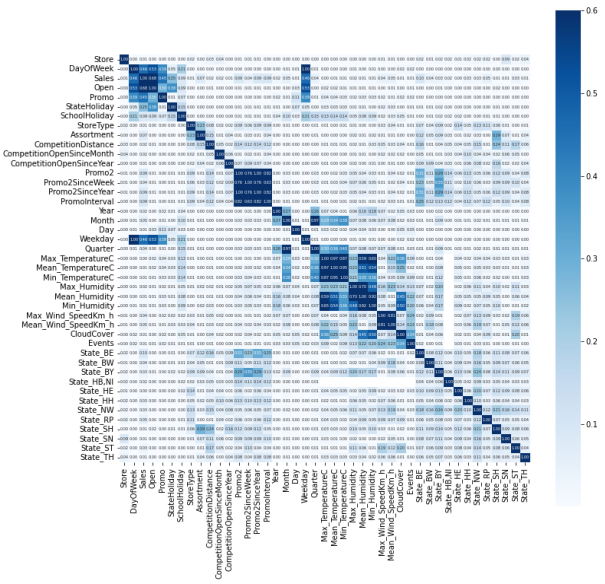
Para el tercer entrenamiento de **XG Boost**, uso **Time Series Split** para hacer validación cruzada a través del tiempo manteniendo las features externas de "Weather" y "State", se obtiene un RMSPE aproximadamente 10% mayor al anterior. Aún así, considero que esta alternativa tendrá mejor performance en producción debido a que la validación cruzada con series temporales dividida en 5 series (folds) puede generalizar mejor. Observo que la importancia asignada a las variables difería del modelo con XG Boost entrenado con holdout set con la presencia de 'StoreType'. No obstante en ambos modelos entrenados con XG Boost y data externa se destacó la importancia de 'StateHoliday' y 'Dayofweek'.

Considerando la utilización de **Light GBM**, otro modelo de gradient boosting, debido al tiempo de duración de entrenamiento del modelo. Este tiende a ser más rápido que XG Boost, utiliza meme memoria y en térmime de precisión ambos algoritmos tienden a dar resultados similares. Por otro lado, me llamaba la atención el bajo valor de RMSPE que obtengo en las corridas anteriores y al analizar las features más importantes de los modelos noto que "Customers" tiene una gran correlación con la variable a predecir y no contaríamos con esta información en producción. Por lo que lo correcto era eliminarla. Gracias a ello obtenemos un resultado más razonable: 0.0235.

En este punto, comenzamos a testear los modelos en Kaggle con el objetivo de evaluar si el error bajo se mantenía en producción también, utilizo *feature importance* para dejar únicamente las variables más importantes y un *N grande variable*. El gráfico obtenido en la cuarta corrida (**Light GBM**) para seleccionar las variables, había resultado en:



Manteniendo las primeras 17 variables (hasta 'Weekday'). Hice dos corridas una con 300 iteraciones, en la cual obtengo un Score Kaggle de 0.59 y otra con 1000 en la que obtengo 0.47. Claramente el N es relevante, pero hay algo que aún no es correcto porque ese valor de Score es alto.



Decido realizar una nueva iteración debido a que los errores en producción eran altos. En función de ello comenzamos a considerar la opción de que la variables de Store, que son en su mayoría las que mantuvimos para la "x" el modelo. Es por ello que decido realizar una *matriz de correlación* y descubrimos no sorprendentemente luego de haber obtenido estos resultados, que las variables en cuestión tienen poca capacidad para generalizar datos desconocidos. Priorizamos las variables externas para evaluar si seleccionando otras variables el error disminuye y elimino las variables 'Promo2SinceWeek', 'Promo2SinceYear', 'CompetitionDistance', 'CompetitionOpenSinceYear', 'CompetitionOpenSinceMonth'. Asimismo, aumentamos el número de N ya que por la cantidad de datos que manejamos resulta razonable. Efectivamente Observo un Score Kaggle Public mejorado de 0.16.

En simultáneo, considero la utilización de **Random Forest** con Time Serie Split. En esta corrida también elimino 'Customers' al tratarse de una variable que se encuentra fuertemente correlacionada con aquella que buscamos predecir y considero que no es información que tendríamos en el tiempo a predecir. De esta manera evalúo la diferencia de performance con Light GBM y obtengo un RMSPE de 0.0193, mejor al de Light GBM pero la velocidad del modelo fue considerablemente mejor (menor tiempo de cómputo del modelo).

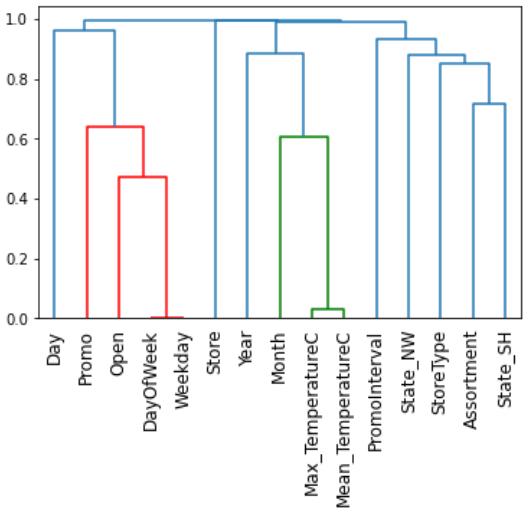
A continuación, utilizo el mismo modelo anterior modificando algume parámetros: 'n_estimators',

'max_depth', 'min_samples_split' y 'min_samples_leaf'. Al notar que los mejores parámetros del entrenamiento anterior se encontraban en los valores límites de los parámetros posibles. El error aumentó 0.029, lo cual tiene sentido dado que la modificación de parámetros tendía a árboles meme profundos y complejos.

Luego, entreno un **Random Forest** calculando el **OOB** score. En este modelo, se realizan N iteraciones en las que se selecciona una muestra bootstrap para entrenar los árboles de decisión, y las observaciones no seleccionadas - también conocidas como "Out of Bag" se usan para calcular el error RMSPE. Debido a que no tiene en cuenta la temporalidad incorporo validación cruzada utilizando Time Series Split. Este modelo tuvo en cuenta los parámetros con los modificados en la versión anterior y también elimino las filas con datos en los que la variable 'Open' fuera igual a 0. Sin embargo, dado que obtengo un OOB score de 0.033 y el error aumentó decido entrenar el modelo final sin OOB.

Finalmente, entreno 2 últimos modelos de Random Forest y lo subo a Kaggle para evaluar su performance frente a datos desconocidos. Ambas corridas constaron de 1000 iteraciones y los mejores parámetros obtenidos. En la primera corrida utilizo las variables más importantes obtenidas en el análisis de feature importance. En la segunda, elegí variables similares a la mejor corrida de Light GBM. En ambas obtengo un Kaggle Score de aproximadamente 0.21, esto se debe a que el resultado de random forest está fuertemente condicionado por 4 variables, las cuales estaban incluidas en ambas corridas, como veremos debajo en un contribution plot.

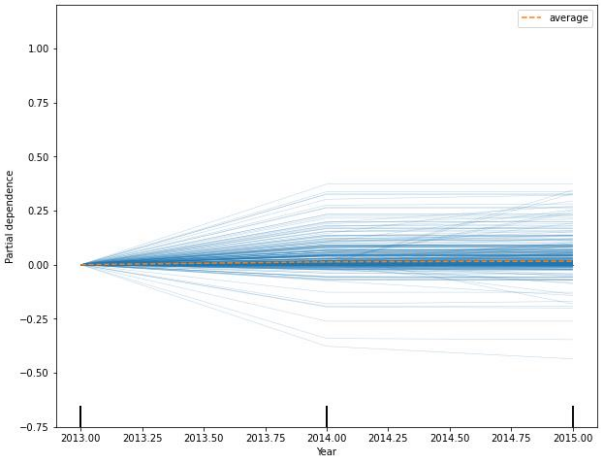
Interpretación del modelo



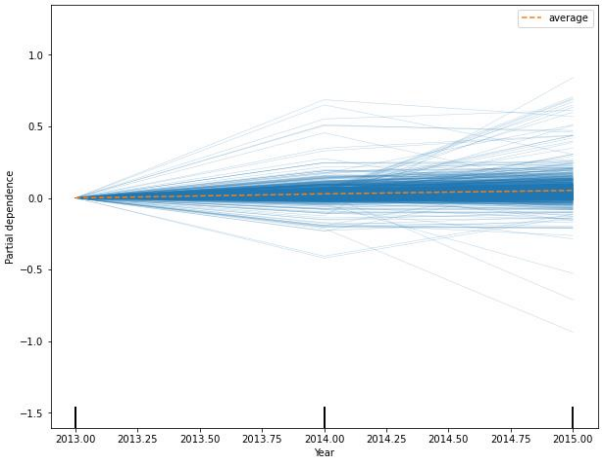
Como parte del análisis de los modelos con mejor Score en Kaggle, considero relevante hacer un análisis de sus componentes principales. En primer lugar para el modelo Light GBM y Random Forest realizo un *clustering jerárquico.* Incluyo únicamente el gráfico obtenido a partir de Light GBM dado que sus resultados fueron similares:
En el dendrograma graficado se puede observar no solo la relevancia de las variables que se utilizaron (cómo lo propuesto y realizado anteriormente en *feature engineering*, sino también la correlación entre las variables). Noto 4 bloques en el siguiente gráfico, donde las variables de los dataset Store y State Store/Weather tienen correlación aparente, lo cual es de esperar. Por otro lado, las variables cronológicas cómo mes y año con la temperatura, y luego las promociones, días de la semana y abierto en un último grupo. Esto último me despertó un interés particular para plantear el segundo modelo. Además, estos gráficos me indican un camino de mejoría para continuar realizando ingeniería de atributos, es decir manteniendo una variable de cada cluster, por ejemplo 'Max_TemperatureC' o 'Mean_TemperatureC'.

Por otro lado, grafico la dependencia del logaritmo de ventas en función del transcurso de los años del dataset utilizando un *ICE plot* para Random Forest y Light GBM. Observo que Partial Dependence en logaritmo tiene un valor promedio de 0 en ambos modelos, significando que, en promedio, no se espera un cambio proporcional en las ventas en relación con la variable 'Year'. Sin embargo, al comparar las

oscilaciones de las observaciones de cada modelo, noto que Light GBM presenta oscilaciones ascendentes y descendentes que pueden indicar patrones estacionales o cíclicos que podrían indicar una leve dependencia sobre el valor del logaritmo de ventas. Mientras que Random Forest tiene predicciones más bien constantes a lo largo de los años.

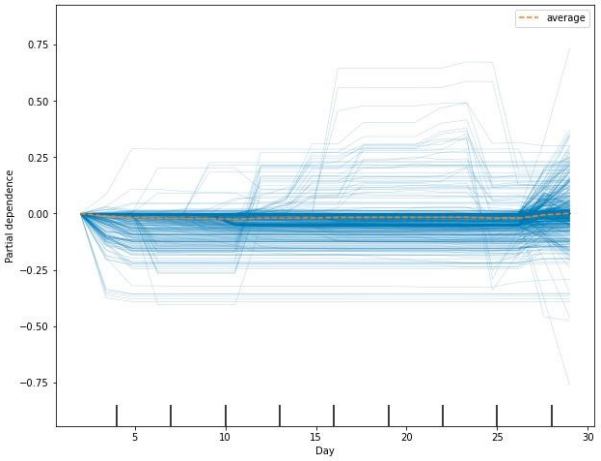


Random Forest - Year

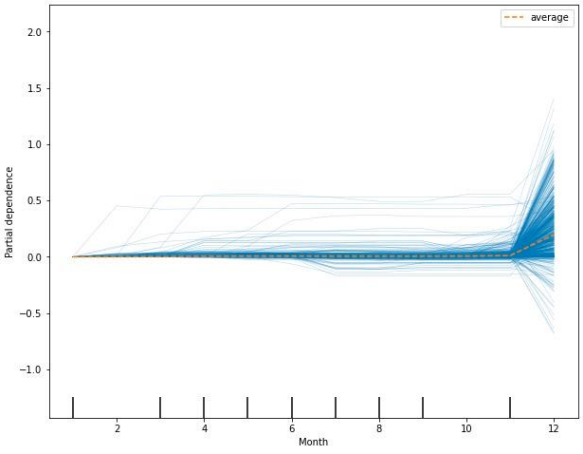


Light BM- Year

Asimismo, considero que podría resultar interesante evaluar 'Day' y 'Month'. En el ICE Plot de 'Month' puedo observar que las oscilaciones de las predicciones del modelo Random Forest son mayores y crecientes a partir del día 15 del mes. Sin embargo, Partial Dependance tiene un valor promedio de 0 hasta el 26/27 del mes donde recién presenta un aumento en el valor promedio, lo que sugiere que se puede esperar un cambio proporcional leve en las ventas en estos días. Algo similar ocurre al analizar el Partial Dependance promedio en función del mes, en donde veo un valor promedio constante hasta el mes de Noviembre donde comienza un valor creciente positivo, por lo que se espera un cambio proporcional mayor en las ventas en Diciembre.

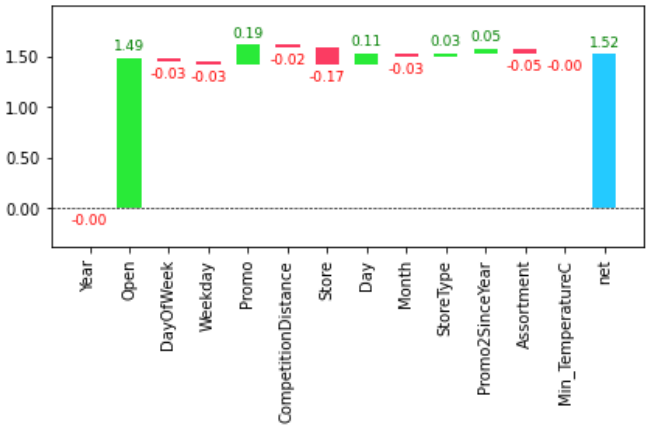


Random Forest - Day



Random Forest - Month

Únicamente para Random Forest, realizo también un análisis de contribuciones. En este gráfico puedo analizar las contribuciones que tiene cada columna para llegar al valor final "net". Es notorio que 'Open', 'Promo', 'Store' y 'Day' son las features con mayor contribución al valor final. La baja contribución del resto de las variables (<= 0.05) generó que el Kaggle Score de las dos corridas de Random Forest con distintas variables sea casi constante.



<u>Segundo problema de Machine Learning para realizar un analisis con foco estadistico.</u>

El objetivo planteado en este segundo problema es el de predecir si una tienda tendrá una promoción activa o no. Se trata de un problema de clasificación binaria, donde utilizo un conjunto de variables (Store, Train y Test) para entrenar un modelo de Random Forest Classifier. El análisis de la variable 'Promo' permite comprender qué factores o características influyen en la decisión de una tienda de participar en una promoción y cómo se pueden utilizar para realizar predicciones más precisas.

Considero que el fin de esta predicción puede ser útil tanto para planificar y gestionar una estrategia de marketing de manera óptima, como para la planificación de inventario ya que la participación en promociones puede influir en la demanda de productos en las tiendas. Al predecir qué tiendas participarán en promociones se puede por ejemplo ajustar la planificación de inventario en consecuencia y, de esta manera, las tiendas se aseguran que cuentan con suficiente stock de productos.

Las variables utilizadas son todas aquellas presentes en los conjuntos de datos Store, Train y Test así como en la data externa utilizada, 'state_store', ya que me interesaba analizar si resultaban relevantes y le sumaba al modelo. Luego de analizar la Feature Importance noto que aunque no son de las que más peso tienen, hay algun estados que le resultan útiles al modelo para predecir.

El conjunto de datos se divide en 'x_train', 'y_train', 'x_test' e 'y_test' dropeando donde corresponde Sales, Date y Promo. Luego, entreno el modelo Random Forest Classifier Out of the Box y se realizan predicciones en el conjunto de prueba.

La precisión del modelo se evalúa utilizando la métrica de accuracy_score y se presenta la matriz de confusión para evaluar el rendimiento del modelo en térmime de falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. Además, se muestra el informe de clasificación que proporciona métricas como precision, recall, f1-score y support para cada clase. Observo que los resultados son positivos ya que, haciendo un desglose para

```
Accuracy: 0.8610361710631494
Confusion Matrix:
[[21256  2930]
 [ 2633 13213]]
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.88      0.88     24186
           1       0.82      0.83      0.83     15846

    accuracy                           0.86     40032
   macro avg       0.85      0.86      0.86     40032
weighted avg       0.86      0.86      0.86     40032
```

cada clase (0 o 1), obtenemos que:

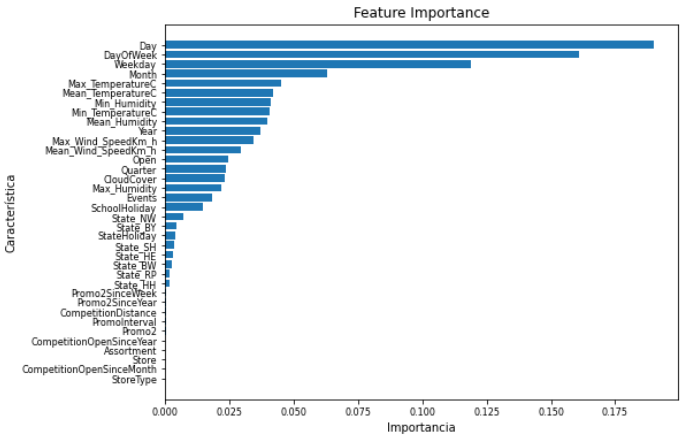
- *Para la clase 0:*

El modelo clasifica correctamente el 89% de las muestras que son realmente de la clase 0. El modelo identifica correctamente el 88% de todas las muestras de la clase 0. La medida F1 es del 88%, que es una buena combinación de precisión y recall.

- *Para la clase 1:*

El modelo clasifica correctamente el 82% de las muestras que son realmente de la clase 1. El modelo identifica correctamente el 83% de todas las muestras de la clase 1. La medida F1 es del 83%, que es una buena combinación de precisión y recall. Además, observando el accuracy el modelo clasifica correctamente el 86% de todas las muestras.

En cuanto a relevancia de las variables que utiliza el modelo, 'Day', 'Day of Week', 'Weekday', 'Month' y la temperatura son relevantes para predecir la variable objetivo 'Promo'. Por otro lado, las variables con valores de importancia más bajos, como 'StoreType', 'CompetitionOpenSinceMonth' y 'Assortment', tienen una influencia mínima en las predicciones del modelo y en caso de buscar hacer una predicción con mayor calidad, sería ideal eliminarlas del modelo.



Consideraciones finales

A modo de cierre me gustaría mencionar que a lo largo del proceso de análisis e implementación de los modelos fui construyendo sobre el mismo y sus variables enfocándome en la investigación de estas, analizando relaciones entre ellas que consideraba acertadas para el objetivo y aplicándoles las transformaciones a mi parecer pertinentes. Es por ello y con el objetivo de continuar perfeccionando este proceso de investigación que hice sobre los dataset en cuestión que me gustaría plantear a continuación una serie de potenciales mejoras para continuar investigando y mejorando el poder predictivo del modelo.

En primer lugar, me gustaría hacer foco en que creo que continuar realizando ingeniería de atributos en el modelo a entrenar le continuará sumando valor. Es por ello que se podría, por un lado, considerar escalar la variable de competition distance en función de lo comentado previamente en el informe, por otro también filtrar los outliers más allá de 3 desvíos estándar de la media en todas las variables incluídas en el modelo final para asegurarme que las predicciones no sean sesgadas hacia por valores atípicos. Otras mejoras futuras podrían incluir la inclusión de datos referentes a 'Google Trends' e indicadores macroeconómicos sobre los estados de Alemania para evaluar la capacidad predictiva que esta información aportaría al modelo. No menos importante es mencionar que nuevos modelos basados en XG Boost podrían haber sido entrenados teniendo en cuenta las mejoras encontradas a lo largo del desarrollo del trabajo con una menor porción de datos. Otro punto interesante, que excede el alcance de este trabajo, pero merece la pena ser mencionado, es el del poder computacional a disposición, en caso de contar con un poder mayor podría considerarse realizar la validación cruzada respetando la temporalidad en los datos a través del método de "rolling window" y, de esta forma, re-entrenar modelos considerando el OOB Score. Por último, dada la reducción en el error que se observó luego de reentrenar Random Forest con una nueva grilla de hiperparámetros, se podría continuar iterando en búsqueda de obtener una optimización hiperparámetros aún más robusta.