INFORME DE PREDICCIÓN

Especialización AutoML [Grupo 1]: Amazon Forecast, Darts, AutoGluon

Laboratorio de Implementación III - Año 2024

Maestría en explotación de datos y gestión del conocimiento

Integrantes:

García, Matías

Martignoni, Franco

Rodríguez Saá, Milagros

Contenido

1. Introducción

En el presente documento se detalla el análisis y los experimentos llevados a cabo para resolver un problema de forecasting utilizando herramientas de Auto Machine Learning. En concreto, se emplearon tres plataformas/librerias distintas: DARTS , Amazon Forecast y AutoGluon.

Primero se realizó un análisis exploratorio de los datos para entender la magnitud del problema de forecasting, consistente en una gran cantidad de productos disimiles en cuanto a volumen general y comportamiento de la serie, acompañado de una marcada concentración de volumen en sólo algunos clientes grandes.

A continuación, se realizó un proceso de ingeniería de características para enriquecer o ajustar cada serie principal. Cabe destacar que este proceso se fue iterando conforme se realizaban nuevas pruebas, de forma de ir adaptando los datos inputs a los modelos.

Debido a que al utilizar autoML hay poco margen de ajuste en el modelo en sí, el mayor trabajo estuvo en la preparación de los datos (limpiezas, estandarizaciones, enriquecimiento con información adicional, etc.) y el enfoque para atacar el problema (de forma general por producto, en clusters de producto, por producto y cliente, por producto y cluster de cliente, etc.), a lo que se sumaron algunas pruebas menores de mejores métricas a emplear y optimizaciones de hiper parámetros cuando fue posible.

En cuanto a la etapa de modelado, se acudió en primer lugar a la librería DARTS. Sin embargo, se encontró que no brinda una herramienta de autoML propiamente dicha, sino que permite trabajar con las series de forma matricial, facilitando algunas operaciones. Es por ello que sólo fue utilizada para responder a los requerimientos de modelos durante las clases y no se considera relevante explayarse en este tema.

En segundo lugar, se hizo uso de la plataforma de AWS para utilizar el servicio de Forecast, el cual permite seleccionar un modelo puntual o hacer uso de autoML. De esta forma, fue necesario crear cuentas gratuitas para poder acceder a los servicios, configurar roles y permisos mediante IAM y almacenar la información en buckets de S3, para finalmente hacer uso de la herramienta de Forecast. Las pruebas fueron limitadas debido al costo por el uso de los servicios (cada cuenta gratuita sólo permitió entrenar dos predictores gratis) y el largo tiempo de entrenamiento respecto a las otras herramientas.

Finalmente, AutoGluon fue la librería mayormente utilizada para los diversos experimentos. En el proceso de entrenamiento pueden especificarse modelos puntuales o bien recorrer una amplia gama de alternativas propuestas por default, que van desde Seasonal Naive hasta DeepAR. También se realiza automáticamente un ensamblado con los mejores predictores. Esta librería fue la más usada por su facilidad de uso, rapidez y cantidad de modelos disponibles (+10 respecto a AWS Forecast).

Este documento está estructurado de forma de repasar cada una de las etapas previamente descriptas. En la sección 2 se resume el análisis exploratorio de los datos;

mientras que la sección 3 sintetiza el proceso de ingeniería de características y generación de datasets utilizados como inputs en los modelos. La sección 4 detalla los experimentos realizados mediante AWS Forecast, DARTS y AutoGluon, poniendo foco en esta última herramienta. Mientras tanto, la sección 5 especifica el modelo escogido para la competencia de Kaggle y el procedimiento para su reproducibilidad.

2. Análisis exploratorio de los datos

En esta instancia se investigaron en detalle los datasets sell in, tb_productos y tb_stocks proporcionados, en búsqueda de descubrir patrones existentes a nivel producto y cliente, determinar productos y clientes más importantes, analizar tendencias y comportamientos de las series, interrelaciones existentes entre toneladas vendidas, pedidas y stock disponible, entre otros objetivos.

A continuación, se enumeran los principales hallazgos encontrados en el proceso de exploración de datos, haciendo énfasis normalmente en el último año de historia (2019).

- El tamaño de los clientes es muy variado. Existen clientes sumamente grandes y otros sumamente pequeños. Los 54 clientes más grandes concentran el 80% de las toneladas anuales de la empresa en 2019. Esto sugiere que hay que darle más preponderancia e incluso un modelado especial a estos clientes.
- Existe una alta concentración de ventas en un pequeño grupo de productos, donde los 50 productos más importantes representan el 50.25% del total de toneladas anuales de 2019.
- La categoría más importante en volumen en 2019 es HC, le siguen FOODS y PC, pero representan alrededor de un 30% del volumen de la primera. Si bien PC es la categoría de mayor cantidad de productos (+700, contra 200-300 de HC y FOODS), el volumen en toneladas no es demasiado grande. REF es una categoría minoritaria tanto en productos (13 productos) como en volumen
- A nivel grupo de productos, el grupo más grande en toneladas de 2019 es ROPA LAVADO (HC) que contiene 99 productos, seguido por ADEREZOS (FOODS) que contiene 69 productos y CABELLO (PC) con 352 productos. El primer grupo más que duplica las toneladas de los que le siguen. El grupo con menor cantidad de toneladas es TE (REF); también resaltan grupos de PC (vinculados a piel y dental) y FOODS (vinculados a caldos).
- Los productos son muy variados en cuanto a tonelada promedio. Un producto con gran cantidad de toneladas comercializadas no necesariamente tiene más toneladas porque se compra más cantidad de producto, sino que es un producto naturalmente grande en cuanto a empaque (ejemplo jabones líquidos de empaque grande contra una crema facial pequeña).
- Dentro de cada categoría e incluso dentro de cada grupo de producto las toneladas promedio por producto son muy variadas. Por ejemplo, dentro de ROPA LAVADO, el tipo de producto con menor cantidad de toneladas es 3.2% del producto con mayor cantidad de toneladas. Lo mismo ocurre dentro de ADEREZOS y CABELLO.

- El porcentaje de productos (a nivel general) en plan de precios cuidados varía mucho por año y mes, pero nunca superó el 3%, además no parece presentar estacionalidad. Particularmente los productos de REF para el primer semestre de 2019 presentaron +10% de productos en el plan. El resto de categorías generales nunca superó el 2.5%
- En búsqueda de descubrir interrelaciones entre las series numéricas proporcionadas se calcularon índices de correlación. Las toneladas solicitadas (cust_request_tn) y vendidas (tn) tienen 99% de correlación, mientras que el número de solicitudes (cust_request_qty) tiene sólo 25% de correlación con las variables anteriores. El stock final por mes tiene muy poca correlación con tn solicitadas y vendidas (15%). Como el objetivo del modelo es predecir toneladas vendidas, únicamente se considera esta serie para el armado de los datasets inputs para el modelo.
- No se ve un comportamiento estacional general para las categorías o grupos de productos. La categoría FOODS presenta mayor cantidad de grupos de productos con estacionalidad relativamente marcada, entre ellos destaca grupo ADEREZOS (picos en verano) y SOPAS Y CALDOS (picos en invierno). También se ve estacionalidad para grupo DEOS de categoría general PC.

De los puntos detallados se puede inferir que el problema de pronóstico es complejo en diversas aristas. Primero, debido a la concentración de toneladas totales en ciertos clientes y productos. Segundo, dada la variabilidad en el tamaño y volumen de productos y categorías. Tercero, a raíz de la diversidad en el comportamiento de las series y diferentes estacionalidades de los productos, sin existir un patrón genérico de evolución de las toneladas en el tiempo por producto.

Lo anterior sugiere que un modelo muy genérico puede no ser adecuado, ya que no captaría las particularidades y especificidades por producto (y cliente o clúster de cliente) necesarias para realizar predicciones precisas. En este sentido, la métrica que utiliza la empresa para medir la bondad de la predicción, pesa con mayor importancia los productos grandes y, luego, tener un modelo que no discrime por tipo de producto y la preponderancia que tienen algunos en cuanto a volumen puede no ser el mejor enfoque.

3. Ingeniería de características y generación de datasets inputs para modelos

Los modelos utilizados admiten tres tipos de dataset diferentes. En primer lugar, el dataset target, que contiene la variable objetivo a predecir, en este caso la variable de toneladas vendidas, bajo la granularidad que se desea realizar la predicción (ya sea a nivel producto, a nivel producto y cliente, etc.). En segundo lugar, un dataset de producto en donde se añade toda la meta data de producto que se crea pueda mejorar la predicción. Por último, un dataset relacionado, que debe tener la misma granularidad que la serie target, y en donde se añaden todos los features adicionales

que se suponga puedan colaborar a explicar el comportamiento de las series del dataset target.

A continuación, se explican las diferentes variantes de cada uno de esos datasets. Como se adelantó en la introducción, el proceso de experimentación exigió ir adecuando cada uno de estos datasets conforme se realizaban pruebas. Para mayor orden del presente documento, aquí se intentarán resumir todas las variantes de datasets estipuladas, para pasar en la parte del modelado a explicar cuáles de estos datasets fueron escogidos según experimento puntual.

3.1. Dataset Target

A grandes rasgos, se generaron cuatro dataset target principales (algunos con ciertas variantes):

- Con apertura a nivel producto – cluster de cliente. Se definieron 10 clusters de clientes según concentración de volumen.
- Con apertura a nivel producto.
- Con apertura a nivel producto acotado; es decir procesando para acotar outliers por cada serie de producto. Por *acotar* se entiende a llevar el valor outlier al valor del percentil utilizado para detectarlo.
    - Variante 1: acotar outliers inferiores (menores a p0.05 de la serie) y superiores (mayores a p0.95 de la serie).
    - Variante 2: acotar outliers superiores (mayores a p0.95 de la serie).
    - Variante 3: acotar outliers superiores (mayores a p0.9 de la serie).

3.2. Dataset de producto

Se generaron dos datasets de producto posibles:

- Versión base. Considerando product_id, cat1, cat2, cat3, brand, sku_size
- Versión full. Considerando variables anteriores y adicionando 10 features números nuevas que recogen correlación de cada producto puntual con los 10 productos más importantes en volumen.

3.3. Dataset relacionado

Se generaron cinco datasets relacionados posibles (notar su vinculo con dataset target, dado que se exija posean la misma granularidad)

- Con apertura a nivel producto – cluster de cliente. Contiene como features adicionales: month (mes del año) , quarter (trimestre del año), days_in_month (días que contiene el mes), sundays (cantidad de domingos del mes), saturdays (cantidad de sábados del mes), tn_m3 (tn de tres meses antes), tn_m6 (tn de seis meses antes), tn_m12 (tn de doce meses antes).
- Con apertura a nivel producto.
    - Versión base. Contiene como features adicionales: month, quarter, days_in_month, sundays, saturdays, tn_m3, tn_m6, tn_m12.

  - Versión full. Contiene como features adicionales: month, quarter, days_in_month, sundays, saturdays, lags (orden 1 a 12), deltas (orden 1 a 12), monthly_percentage (porcentaje de tn anuales que corresponden al mes del registro), lowerextremepoint (dummy para recoger valor extremo inferior), upperextremepoint (dummy para recoger valor extremo superior)
- Con apertura a nivel producto, procesando para acotar outliers [Variante 2]. Contiene como features adicionales: month, quarter, days_in_month, sundays, saturdays, lags (orden 1 a 12), deltas (orden 1 a 12), monthly_percentage, ipc (serie de tiempo del indice de inflación), dólar (serie de tiempo de valor del dólar blue).

4. Experimentos: pruebas de modelado

Los experimentos giraron en torno a: 1) pruebas de diversas métricas de optimización; 2) pruebas de división del problema en clusters de series de productos homogéneos; 3) pruebas de división de problema en clusters de clientes; 4) acotar valores extremos por serie; 5) enriquecer con información externa de productos y features relacionados; 6) estandarización de los datos; 7) optimización de hiper parámetros.

La mayor parte de pruebas se realizaron con AutoGluon. Sin embargo, también se ejecutaron algunos modelos con AWS Forecast. A continuación, se detalla qué se probó con cada herramienta. Cabe desatacar que la guía más detallada de experimentos se encontrará en el apartado AutoGluon.

4.1. AWS Forecast

En Amazon Forecast, los proyectos de previsión implican tres etapas:

1) Importación de conjuntos de datos. Se generan tres conjuntos de datos, que siguen lo detallado en el apartado 3 (target, producto, relacionado).
2) Entrenar predictores. Aquí puede escogerse la opción de autoML (auto_predictor) o entrenar un predictor con uno o varios modelos especificados y personalizar el entrenamiento (que la opción de autoML no lo permite).
3) Generación de inferencias futuras. Una vez entrenado el predictor, se generan las inferencias futuras.

A continuación, se detallan los experimentos realizados en AWS Forecast. Notar que los nombres de los datasets se corresponden con lo explicado en el apartado anterior.

Experimento 1:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12

- Validación: Automática (la validación en AWS Forecast no puede customizarse bajo la selección de autoML)
- Dataset target: con apertura a nivel producto
- Dataset productos: si; versión base
- Dataset relacionado: si; con apertura a nivel producto (versión base)
- Descripción: versión simple con uso de autoML
- Puntaje Kaggle: 0.253

Experimento 2:

- Apertura: product_id, cluster_client_id
- Historia desde-hasta: 2017-01 / 2019-12
- Validación: automática
- Dataset target: con apertura a nivel producto – cluster de cliente
- Dataset productos: si; versión base
- Dataset relacionado: si; con apertura a nivel producto – cluster de cliente
- Descripción: versión simple con uso de autoML
- Kaggle: no se cargó la salida al obtener malos resultados al validar

Experimento 3:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Validación: Automática
- Dataset target: con apertura a nivel producto
- Dataset productos: si; versión base
- Dataset relacionado: si; con apertura a nivel producto (versión base)
- Descripción: prueba auto_predictor
- Kaggle: no se cargó la salida al obtener malos resultados al validar

Experimento 4:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Validación: Automática
- Dataset target: con apertura a nivel producto [3 conjuntos de series diferentes según cluster de producto]
- Dataset productos: si; versión base [3 conjuntos de metadata de ítems diferentes según cluster de producto]
- Dataset relacionado: si; con apertura a nivel producto (versión base) [3 conjuntos de series relacionadas diferentes según cluster de producto]
- Descripción: versión segmentada por clusters de productos con uso de autoML. Se generaron 3 modelos diferentes según clúster de serie de tiempo de producto.
- Kaggle: 0.269

Proceso de clustering de las series de productos realizado mediante TimeSeriesKMeans, luego de estandarizar cada una de las series a través de TimeSeriesScalerMeanVariance, ambos pertenecientes a librería tslearn. Cabe destacar que si bien se estandarizaron los valores de cada serie de producto para la segmentación, al modelo de AWS Forecast no ingresaron las series estandarizadas. Se generan únicamente 3 clusters de productos (posiblemente muy pocos). Este proceso vuelve a repetirse para algunos experimentos de AutoGluon que se explicarán más adelante.

Experimento 5:

- Apertura: product_id
- Historia desde-hasta: 2018-01 / 2019-12
- Validación: Automática
- Dataset target: con apertura a nivel producto
- Dataset productos: si; versión base
- Dataset relacionado: si; con apertura a nivel producto (versión base)
- Descripción: versión con estandarización de cada una de las series de producto (por media y desvio) y uso de autoML
- Kaggle: 0.317

4.2. AutoGluon

A continuación, se detallan las pruebas realizadas con AutoGluon. La operatoria es muy similar a la de AWS Forecast, dado que a cada serie de tiempo target puede adicionársele metadata de productos y un dataset relacionado.

Experimento 1:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. A modo de prueba de la librería
- Kaggle: 0.249

Experimento 2:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No

- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación
- Kaggle: 0.249

Experimento 2_1:

- Apertura: product_id
- Historia desde-hasta: 2018-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Se prueba acotar 1 año, de modo que se quita 2017 de la historia
- Kaggle: 0.261

Experimento 2_2:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Prueba de métrica customizada para la optimización, según la función de error que utiliza la multinacional
- Kaggle: no se cargó la salida al obtener malos resultados al validar

Experimento 2_3:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Prueba de métrica customizada para la optimización, según la función de error que utiliza la multinacional
- Kaggle: 0.251

Experimento 2_4:

- Apertura: product_id

- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 1]
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Prueba utilizando serie target acotada en valores extremos inferiores y superiores. Valores superiores al percentil 95 de la serie se llevan a ese valor y valores inferiores al percentil 5 de la serie se llevan a ese valor
- Kaggle: 0.249

Experimento 2_5:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Prueba utilizando serie target acotada en valores extremos inferiores y superiores. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior)
- Kaggle: 0.247

Experimento 2_5_1:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 6 periodos de ventana de validación. Prueba utilizando serie target acotada en valores extremos inferiores y superiores. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior).
- Kaggle: 0.255

Experimento 2_6:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 3]
- Dataset productos: No
- Dataset relacionado: No

- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Prueba utilizando serie target acotada en valores extremos inferiores y superiores. Valores superiores al percentil 90 de la serie se llevan a ese valor (no se acota valor inferior)
- Kaggle: 0.256

Experimento 3:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Se incorpora optimización de hiperparámetros
- Kaggle: 0.277

Experimento 4:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: Básico
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo con data de productos (versión básica, sólo features de maestro de productos proporcionado por la multinacional). Un periodo de ventana de validación
- Kaggle: 0.263

Experimento 4_1:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: Full
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo con data de productos (versión full, que incluye correlaciones con items más importantes). Se agregan 3 periodos de ventana de validación. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior)
- Kaggle: 0.263

Experimento 5:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: Básico
- Dataset relacionado: Con apertura a nivel producto. Básico
- AutoML: Default
- Descripción: Modelo con data de productos (versión básica) y data relacionada (versión básica). Se agregan 3 periodos de ventana de validación
- Kaggle: 0.27

Experimento 6:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: Básico
- Dataset relacionado: Con apertura a nivel producto. Full
- AutoML: Default
- Descripción: Modelo con data de productos (versión básica) y data relacionada (versión full). Se agregan 3 periodos de ventana de validación. Existencia de gran sobreajuste
- Kaggle: 0.321

Experimento 7:

- Apertura: product_id, cluster_client_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto, cluster de cliente
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Prueba de predicción a nivel producto y cluster de cliente (10 clusters de clientes). Modelo básico. No incluye data de productos ni relacionada. Un periodo de ventana de validación
- Kaggle: 0.26

Experimento 8:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default

- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se estandariza cada una de las series de productos por media y desvío. Un periodo de ventana de validación
- Kaggle: 0.5

Experimento 9:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No. Chronos
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Un periodo de ventana de validación. Se usa el modelo Chronos PRE-ENTRENADO
- Kaggle: 0.29

Experimento 10:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Un periodo de ventana de validación. Se entrenan 7 modelos diferentes según 7 clusters de productos obtenidos mediante TimeSeriesKMeans (similar a lo que se explicó en el apartado de AWS Forecast)
- Kaggle: 0.29

Experimento 11:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Un periodo de ventana de validación. Se entrenan tantos modelos como categorías que conforman cat_2, entendiendo que cada uno de los grupos se corresponden con productos medianamente similares.
- Kaggle: 0.263

Experimento 12:

- Apertura: product_id

- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Un periodo de ventana de validación. Se realiza un procedimiento de clusterización en dos etapas. Etapa 1: clustering de las series mediante TimeSeriesKMeans (10 clusters). Etapa 2: clustering de los productos en función de los features existentes en el maestro de productos proporcionado por la multinacional (cat1, cat2, cat3, brand, sku_size) y el cluster_id de la serie obtenido en la etapa 1, mediante KModes al tratarse de todas variables categóricas (se acota a 5 clusters)
- Kaggle: 0.268

Experimento 13:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Un periodo de ventana de validación. Se realiza un procedimiento de clusterización en dos etapas. Etapa 1: clustering de las series mediante TimeSeriesKMeans (10 clusters). Etapa 2: clustering de los productos en función de los features existentes en el maestro de productos proporcionado por la multinacional (cat1, cat2, cat3, brand, sku_size) y el cluster_id de la serie obtenido en la etapa 1, mediante KModes al tratarse de todas variables categóricas (10 clusters)
- Kaggle: 0.257

Experimento 13_1:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Se entrenan 10 modelos diferentes según 10 clusters de productos obtenidos mediante TimeSeriesKMeans
- Kaggle: 0.285

Experimento 13_2:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico. No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación. Se realiza un procedimiento de clusterización en dos etapas. Etapa 1: clustering de las series mediante TimeSeriesKMeans (20 clusters). Etapa 2: clustering de los productos en función de los features existentes en el maestro de productos proporcionado por la multinacional (cat1, cat2, cat3, brand, sku_size) y el cluster_id de la serie obtenido en la etapa 1, mediante KModes al tratarse de todas variables categóricas (30 clusters)
- Kaggle: 0.273

Experimento 14:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: Full
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo completo (incluye data de productos y relacionada) con target acotado. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior). Se agregan 3 periodos de ventana de validación.
- Kaggle: 0.314

Experimento 14_1:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: Full
- Dataset relacionado: No. Temporal Fusion Transformer
- AutoML: Modelo completo (incluye data de productos y relacionada) con target acotado. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior). Se agregan 3 periodos de ventana de validación. Optimización de hiperparámetros de Temporal Fusion Transformer (el algoritmo que mejor se desenvuelve cuando coexisten dataset target, de productos y relacionado)
- Kaggle: 0.364

Experimento 15:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: Full
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo completo (incluye data de productos y relacionada) con target acotado. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior). Se agregan 3 periodos de ventana de validación. Se realiza clustering de series de productos por medio de TimeSeriesKMeans (10 clusters)
- Kaggle: 0.307

Experimento 16:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico con target acotado. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior). No incluye data de productos ni relacionada. Un periodo de ventana de validación. Se realiza un procedimiento de clusterización en dos etapas. Etapa 1: clustering de las series mediante TimeSeriesKMeans (10 clusters). Etapa 2: clustering de los productos en función de los features existentes en el maestro de productos proporcionado por la multinacional (cat1, cat2, cat3, brand, sku_size) y el cluster_id de la serie obtenido en la etapa 1, mediante KModes al tratarse de todas variables categóricas (10 clusters)
- Kaggle: 0.27

Experimento 17:

- Apertura: product_id
- Historia desde-hasta: 2017-01 / 2019-12
- Dataset target: Con apertura a nivel producto acotado [variante 2]
- Dataset productos: No
- Dataset relacionado: No
- AutoML: Default
- Descripción: Modelo básico con target acotado. Valores superiores al percentil 95 de la serie se llevan a ese valor (no se acota valor inferior). No incluye data de productos ni relacionada. Se agregan 3 periodos de ventana de validación.

Se entrenan 25 modelos diferentes según 25 clusters de productos obtenidos mediante TimeSeriesKMeans

- Kaggle: 0.283

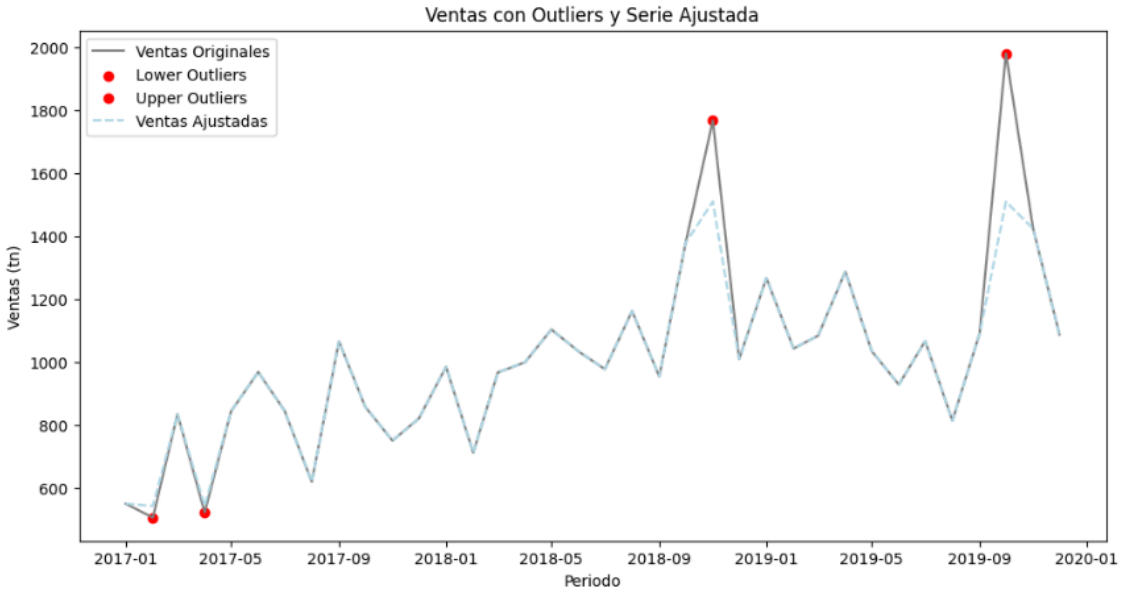
5. Modelo final elegido

El modelo final escogido es el modelo 2_5, cuyas características son:

- Apertura: product_id
- Periodo de Historia: Desde enero de 2017 hasta diciembre de 2019
- Dataset Target: Con apertura a nivel producto, acotando outliers superiores [variante 2]
- Dataset Productos: No incluye datos de productos adicionales
- Dataset Relacionado: No incluye datos relacionados adicionales
- AutoML: Configuración predeterminada (Default)
- Ventanas de Validación: 3 periodos de ventana de validación

Este modelo básico no incluye datos adicionales de productos ni datos relacionados. La principal característica es que la serie temporal objetivo se ha acotado en sus valores extremos superiores. Los valores superiores al percentil 95 de la serie se ajustan a ese valor, mientras que no se aplica ningún acotamiento a los valores inferiores.

Si bien este modelo no es el de mejor puntaje en el Leaderboard Publico, se escogió debido a que en general los modelos acotando outliers dieron mejores resultados. La imagen muestra la forma de acotar outliers de una serie particular (superiores e inferiores). En el caso de este experimento, sólo se acotaron los outliers superiores, no inferiores.



La gran conclusión que obtuvo el equipo esque, agregar datos relacionados y desagregar el problema (en clusters, por ejemplo), no mejora el resultado. El modelo básico siempre da mejores resultados, hablando de lo poderosa que es la herramienta elegida de autoML.

¿Cómo reproducir el experimento? Siguiendo estructura de repositorio:

1. Obtener datos inputs al modelo. Ejecutar notebook "4. Mejorando datasets nivel producto v2 [data target, cut outliers].ipynb" de la carpeta "1. EDA y Armado de Datasets"
2. Entrenar modelo y obtener salidas. Ejecutar notebook "4. AutoGluon/Exp2_v5 Modelo básico + Ventana validación - Acotar outliers sup.ipynb" de la carpeta "4. AutoGluon"
3. Buscar la salida "prediccion_autogluon_fill_upper_outliers.csv" en la carpeta "Outputs"

Link repo: https://github.com/milagrosrsaa/mcd_labo3_autoML1/tree/main

Bibliografía y links

AutoGluon Time Series - Forecasting Quick Start
https://auto.gluon.ai/stable/tutorials/timeseries/forecasting-quick-start.html

What Is Amazon Forecast?
https://docs.aws.amazon.com/forecast/latest/dg/what-is-forecast.html

Darts: Time Series Made Easy in Python
https://unit8.com/resources/darts-time-series-made-easy-in-python/

Introduction to Amazon Forecast for Predictive Analytics | by Christopher Adamson | Medium
https://medium.com/@christopheradamson253/introduction-to-amazon-forecast-for-predictive-analytics-8b87617535dd

Darts Time Series TFM Forecasting | by Mark W Kiehl | Medium
https://medium.com/@markwkiehl/darts-time-series-tfm-forecasting-8275ccc93a43

Time Series Forecasting using AutoGluon. | by Rakesh M K | Medium
https://medium.com/@mkk.rakesh/time-series-forecasting-using-autogluon-34f8780b8214

AutoGluon-TimeSeries : Creating Powerful Ensemble Forecasts - Complete Tutorial
https://aihorizonforecast.substack.com/p/autogluon-timeseries-creating-powerful

AutoGluon–TimeSeries: AutoML for Probabilistic Time Series Forecasting
https://arxiv.org/pdf/2308.05566