Facultad de Ingeniería

Maestría en Ciencia de Datos

Materia Laboratorio III

Integrantes:
Leandro Boisselier
Nicolás Churrarin
Franco Sonzogni

Objetivo

El objetivo de este proyecto fue desarrollar un modelo predictivo robusto capaz de anticipar las ventas mensuales de productos para el mes de febrero de 2020 (2020-02), a partir del análisis de datos históricos de ventas.
La capacidad de estimar de manera precisa las ventas futuras es crítica para la optimización de la gestión de inventarios, la reducción de quiebres de stock, la mejora en la planificación de compras y el ajuste eficiente de estrategias comerciales.

Proceso Realizado

Análisis exploratorio de datos (EDA): identificación de estacionalidades, atipicidades y patrones relevantes.
Para completar los datos faltantes en los casos de técnicas de regresión tabular y agrupamiento por [customer_id], [product_id] y completamos registros tomando como base la fecha de inicio de un producto.

Ingeniería de features

Variables creadas para técnicas de regresión tabular (No serie de tiempo). Finalmente trabajamos entre 190 y 250 features.

- *Fecha real:* Derivada del campo periodo (formato YYYYMM), transformada a datetime.
- *Variable objetivo (clase):* Toneladas (tn) del mes +2 por cliente y producto. También se probó con Ratios log(1 + tn_2)/log(1 + tn). Se experimentó también con el parámetro [linear_tree] en lightgbm.
- *Lags:* Lags de 1 a 36 meses.
- *Diferencias Lags:* tn - lag_tn.
- *Medias móviles:* Medias móviles de 1 a 36 meses.
- *Diferencias medias móviles:* tn - rolling_mean.
- *Contexto temporal:* Año, mes, quarter, día de semana.
- *Representación cíclica:* seno/coseno del mes.
- *Mínimos y máximos:* Flags (1/0) si el valor actual es el mínimo/máximo de las ventanas 3, 6, 12 meses.

- *Volatilidad e impulso:* Porcentaje de cambio (pct_change) y desviación estándar móvil (rolling_std).
- *Jerárquico:* Promedios por brand, cat1, cat2, cat3 y razón contra promedio (ratio).
- *Cluster DTW:* Cluster asignado como variable categórica y multiplicado por el mes.
- *Factorización:* Transformación de todas las variables categóricas a enteros.
- *Target encoding:* Media histórica de la clase por customer_id y product_id.
- *Contexto económico del país:* Enlace externo por periodo con variables macroeconómicas (por ejemplo, inflación, tipo de cambio, etc.).
- *Features de prophet:* Se suman como features las variables de un modelo prophet por *customer* y *product*.
- *Estandarización:* Estandarización (media 0, desvío 1) por product_id, aplicada antes del FE para evitar fugas de información. Se creó un diccionario (scaler_dict) con los valores de media y desvío por producto para revertir el escalado post-predicción. También exploramos el escalado de la clase mediante la aplicación de logaritmo.
- Lags convertidos a Bins: Pensando los lags como series de tiempo se construyeron features a nivel de decil, octil y cuartil, asociando cada categoría a una letra (0->A, 1->B, etc) y se empaquetaron en strings de diversas longitudes de 4 a 8 representando patrones de comportamiento previos.

Modelos Evaluados

AutoGluon

Probamos dos tipos de granularidad: producto y producto/cliente y con técnicas tabulares, de serie de tiempo y un enfoque híbrido.

Modelos de boosting: CatBoost, LightGBM, XGBoost (cada uno en múltiples variantes y configuraciones). Usamos en algunos experimentos “sample_weight” con los valores de “tn”.

Modelos de redes neuronales: Activados en forma automática cuando detecta una GPU disponible.

Modelos híbridos: Toman para un modelo de regresión tabular el set de datos del top 100 de cliente con mayor promedio de ventas y para el resto de los

datos, los agrupa por producto y usa un modelo de serie de tiempo. Al final hace una suma de predicciones por producto.

Resultados en public: [0.247 - 0.266]

Regresión Lineal

Granularidad: Producto
Estrategia: propuesta por la cátedra, genera variables de rezago (lags) y entrena un modelo de regresión lineal usando un grupo especial de 33 productos con historial completo. El modelo se ajusta para predecir las ventas de febrero 2019 a partir de los datos de diciembre 2018 y sus lags. Luego se predicen las ventas de diciembre 2019 para los 33 productos: si el producto tiene todos los datos de lags, usa la regresión entrenada; si le faltan datos, estima usando el promedio de los valores disponibles.
clase: tn+2
resultados en public: 0.250

Prophet

Granularidad producto y customer - producto
clase: [tn+2]
Resultados en public: La agregación customer y producto lo usamos solo para obtener features que incorporamos al modelo (tendencia, términos aditivos, otros). También se probó un prophet a nivel producto con clase tn+2
resultado en public: 0.900

Arimas

Granularidad: Producto
Estrategia: Se probaron opciones con y sin suavizado, con clipping a 0 de las predicciones
clase: [tn+2]
resultados en public: 0.286

Granularidad: Producto Cliente
clase: [tn+2]
resultados en public: 0.305

XGboost

Granularidad cliente - producto
clase: [tn] de mes + 2
resultados en public: Entre 0.34 y 0.37.

Lightgbm
Granularidad producto
clase: log(1 + tn_2)/(log(1 + tn))
semilleríos: + 100
bayesiana: + 20 horas + 1000 trials
resultado en public: 0.303

Granularidad producto cliente
clase: log(1 + tn_2)/(log(1 + tn))
semilleríos: + 10
bayesiana: + 20 horas + 50 trials
resultado en public: 0.384

Problemas generales encontrados:

Creemos que nos faltó trabajar mas en la identificación de nuevas características que permitan generar variables que ayuden a la explicabilidad de las variaciones de las distintas series de tiempo.
Al usar lightgbm con [linear_tree] fué muy difícil manejar la variabilidad de los resultados.
Con ligthgbm y al usar un ratio en la clase hubo que aplicar logaritmos para evitar generar pronósticos muy altos.
En la agregación por cliente y producto para modelos de regresión tabular experimentamos problemas de capacidad computacional para poder ejecutar el modelado.
No pudimos completar el análisis de DTW a nivel de cliente y producto, para una muestra de los últimos 3 meses de cada combinación. Solo pudimos hacerlo a nivel producto y periodo.
Con Autogluon y en los experimentos en los cuales usamos estandarización por producto, no notamos una mejora importante en los valores de Kaggle, prácticamente obtuvimos los mismos que sin estandarizar.

Elección de la técnica:

Elegimos un Ensemble de 5 archivos. 4 generados por Autogluon y 1 de Regresión Lineal. De Autogluon, 2 son basaddos en modelos tabulares y 1 en redes neuronales. El cuarto, es un híbrido que combina tabular con series de tiempo. En éste último, se consideran el top 100 de los clientes con mayor poder de compra (tabular) y para el remanente por producto (series).
Una vez obtenida la puntuación en el public factorizamos en 1.01 y 0.99 para encontrar un mejor ajuste. Quedó en un score de 0.244 en el Public.