



Laboratorio de Implementación III 2025

Maestría en Explotación de Datos y Gestión del Conocimiento

Barbona, Dylan

Martin, Ariel

Szulacki, Tomas

- **¿Qué probamos y cuáles fueron los resultados de esas pruebas?**

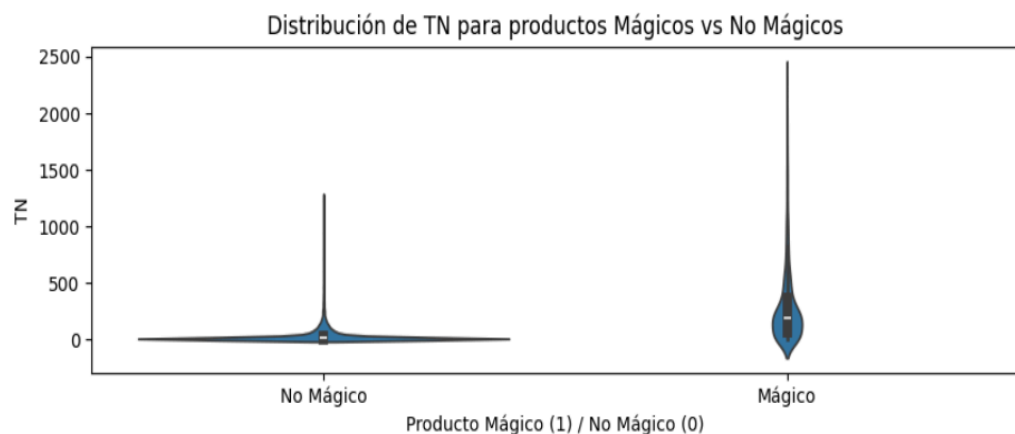
En el transcurso de la cursada, fuimos probando distintos modelos y algoritmos, con el objetivo de entender qué estrategias performaban mejor, y como podíamos obtener una buena predicción frente al objetivo de estimar las toneladas vendidas por la multinacional en febrero de 2020. En ese camino probamos:

- **Modelos simples**, como el promedio de 2019 agrupado por product_id y distintas combinaciones lineales del mismo, que nos sirvió como benchmark base.
- **Modelos de series de tiempo, como ARIMA, SARIMA, Holt-Winters, Prophet** agrupado por product_id.

En el caso de los modelos ARIMA, SARIMA y Prophet, se realizó una división del dataset en entrenamiento y validación. Se entrenó con los datos hasta 2019-11 y se realizó optimización de hiper parámetros, validando en el período 2019-12. Luego, con los mejores hiper parámetros hallados, se volvió a construir el modelo con los datos hasta el período 2019-12, y con dicho modelo se realizó la predicción para el período 2020-02.

- **Regresión lineal**, entrenada exclusivamente sobre los llamados “productos mágicos”, que aumentó notoriamente la performance en el public.

En este caso se analizaron en forma individual las series de tiempo a nivel de product_id, mirando la variabilidad de cada serie. Con ello se agruparon los product_id con menor variabilidad con un indicador de “Mágico”. Como se puede apreciar en el siguiente gráfico, los product_id seleccionados poseen una distribución con mucho menor variabilidad que los product_id identificados como “No mágicos”.



Con los product_id “mágicos” se construyó un dataset con las columnas product_id, periodo (Año-Mes), tn_0 a tn_11 (tn_0 correspondiente a las toneladas vendidas en el mes correspondiente al período (Año-Mes), y tn_1 a tn_11 correspondientes a los lags de los 11 meses anteriores), y una columna clase que corresponde al mes a predecir (Lag + 2).

Luego se seleccionaron los datos correspondientes al periodo 2018-12 para cada uno de los product_id mágicos, y con dicho dataset se construyó una regresión lineal simple con 12 coeficientes y un intercepto a saber:

Intercepto: -0.23325204529917373

	Variable	Coeficiente
0	tn	0.096711
1	tn_1	0.137613
2	tn_2	0.053502
3	tn_3	-0.116880
4	tn_4	-0.192194
5	tn_5	0.107450
6	tn_6	-0.003234
7	tn_7	-0.042620
8	tn_8	0.121842
9	tn_9	-0.062090
10	tn_10	-0.033233
11	tn_11	0.199008

Por último se seleccionó el período 2019-12 para todos los product_id a predecir que tuvieran valores en los 12 meses del año 2019, y se aplicó la regresión lineal anteriormente construida para hallar el valor del período 2020-02 (clase).

En el caso de los product_id a predecir que no tuvieran venta en los 12 meses del año 2019, se tomó en este caso el promedio de venta del año 2019.

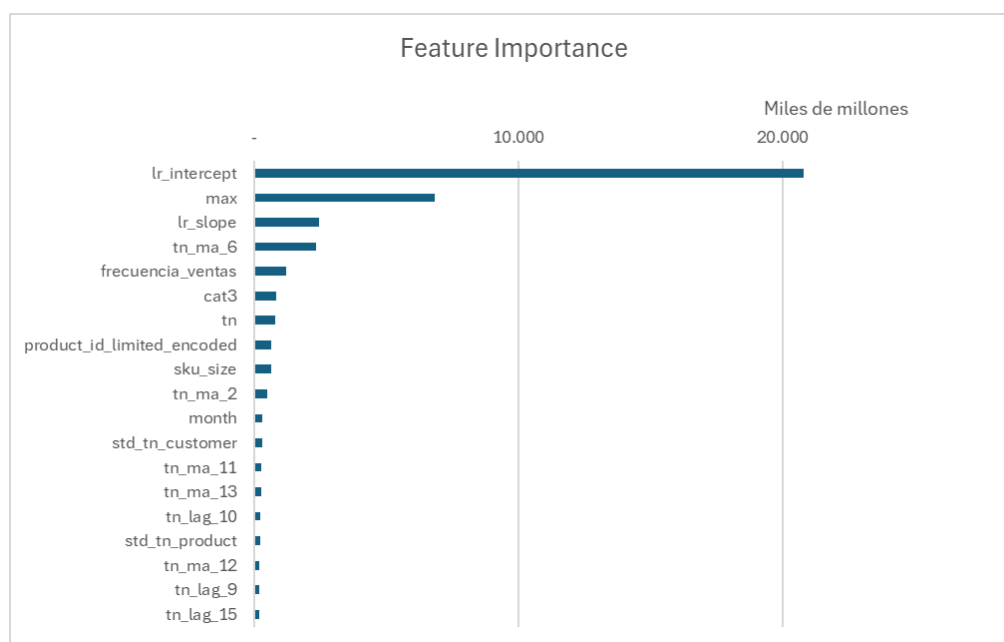
- **LightGBM**, con feature engineering intensivo (lags, medias móviles, tendencias, deltas contra el promedio, deltas contra las medias móviles, variaciones porcentuales contra la media, variaciones porcentuales contra las medias móviles, etc.) y con granularidad a nivel <product_id;client_id>

La generación de features y el pre-procesamiento de los datos fue clave para lograr un modelo con resultados aceptables, sin embargo al trabajar con la granularidad <product_id, client_id>, y al incorporar features al dataset, los requerimientos de recursos de hardware escalaban en forma significativa sin poder lograr resultados significativamente mejores que modelos más simples, como el de regresión lineal.

Se eligió una estrategia de entrenamiento rolling window dividiendo al dataset en 12 períodos para train y 1 período para validación, siempre tomando los

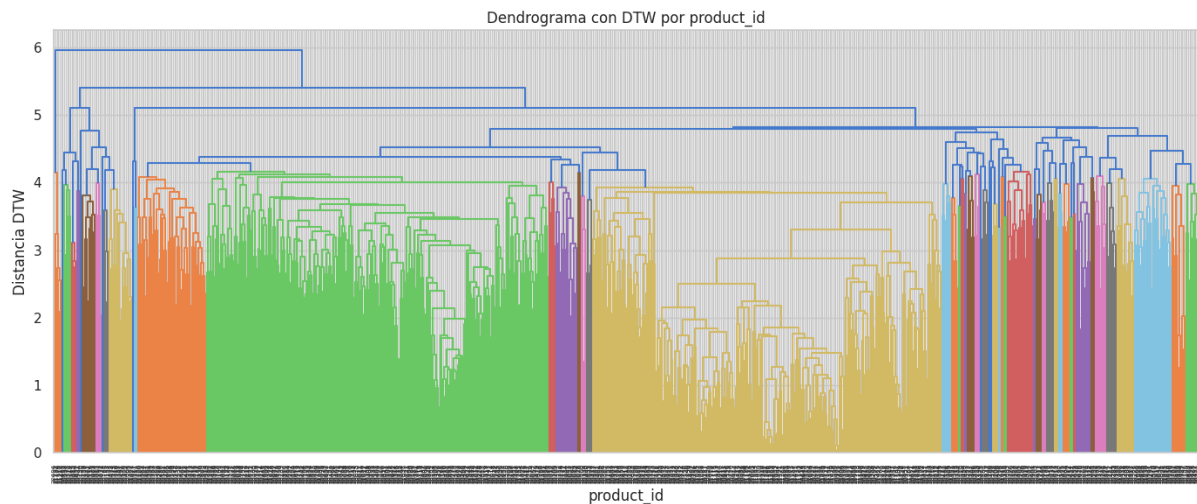
datos hasta el período 2019-09. Para optimización de los hyper parámetros se utilizó Optuna. Luego, con los mejores hyper parámetros se construyó el modelo final y se volvió a entrenar con el mismo método de validación, pero tomando todo el dataset de train y validación completo (hasta el período 2019-09). Se testeó el modelo sobre el período 2019-10, arrojando la predicción para el período 2019-12, que se comparó contra los datos reales de venta de dicho período.

De este entrenamiento surgieron las siguientes features como más relevantes:

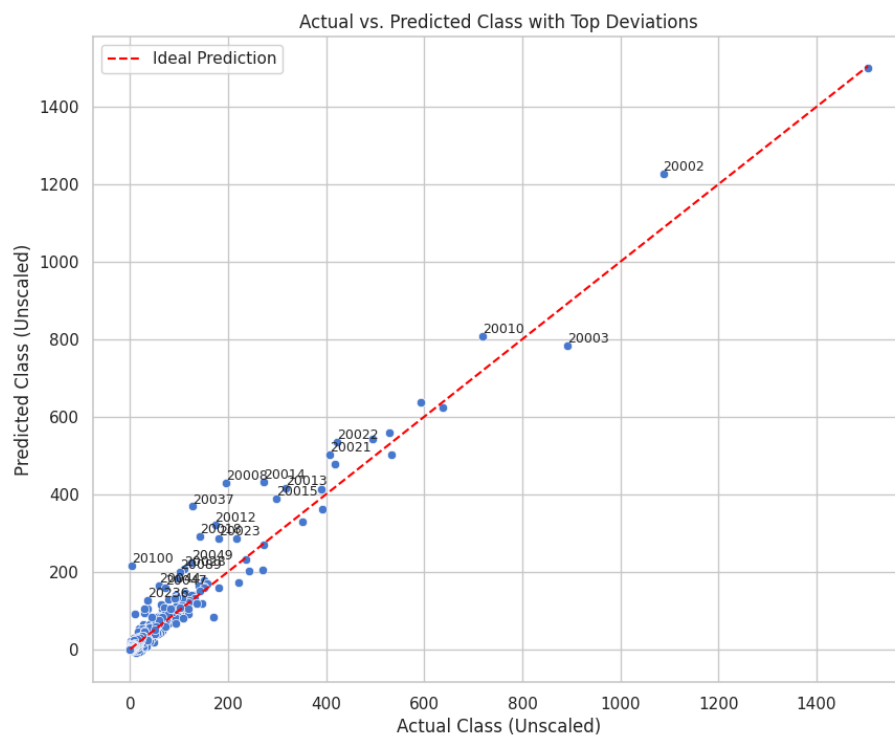


- **Auto Gluon**
- **Clusterización Jerárquica con Dynamic Time Warping** con granularidad por product_id y también con granularidad por <customer_id, product_id>. Luego se generó un modelo de regresión lineal distinto para cada cluster.

Primero se trabajó con una granulometría de los datos a nivel de product_id (para las 780 series de tiempo). Se realizaron los cálculos de distancias con Dinamic Time Warping entre todas las series de tiempo, construyendo una matriz de distancias. Luego se realizó con dichas distancias la clusterización jerárquica. El siguiente gráfico de Dendrograma muestra las distancias calculadas con DTW entre cada serie de tiempo y la agrupación jerárquica de las mismas.



Se realizaron distintos cortes en base al dendrograma anterior (DTW = 2, 3, 3.5, 4, 4.18, 4.2, 4.5, 4.6, 4.8, 5, 6), obteniendo distintas cantidades de clusters y diferentes distribuciones de las series temporales por cluster. Luego, con cada cluster por separado se entrenó un modelo de regresión lineal simple sobre los 12 meses del año 2018. Con dichas regresiones lineales se realizó primero una predicción para el período 2019-12, y dicho valor se comparó con el valor real de venta para dicho período. De esta comparación, se identificaron una serie de product_id que el modelo predecía muy mal, y se decidió quitarlos del conjunto de datos de entrenamiento. Volviendo a entrenar el modelo sin estos product_id, se obtuvo otras regresiones lineales, que se volvieron a testear contra el valor real de venta en el período 2019-12. Allí sí se obtuvo un mejor ajuste del modelo, como se puede observar en el siguiente gráfico:



Con este modelo se realizó la predicción para el periodo 2020-02 para todos los product_id que tuvieran venta en los 12 meses del año 2019. Para los productos que no tuvieran venta en los 12 meses del año 2019 se tomó el promedio de los últimos 12 meses ajustado por un factor de corrección.

No obstante, la predicción final no logró un score en el public de Kaggle mejor que el modelo de Regresión Lineal simple sobre los product_id “Mágicos”.

- **Ensamble multinivel** seleccionando para predecir a nivel de product_id aquel modelo que tuviera menor error en la predicción de Diciembre-2019. Los modelos considerados en este caso fueron Auto Gluon, LightGBM, DTW, Regresión Lineal sobre los product_id “mágicos”.

Modelo	Public Leaderboard
Auto Gluon	0.243
Regresion	0.25
Ensemble	0.265
Promedio	0.273
LightGBM	0.301
Clusterización Jerárquica	0.305
Arima	0.313
Prophet	0.313
Holt- Winters	0.353

- **¿Qué modelos no funcionaron?**

- **Prophet / ARIMA / SARIMA / Holt-Winters:** Son modelos que asumen estacionalidad, tendencias suaves y patrones consistentes en el tiempo. En nuestro caso, muchas series eran cortas, ruidosas y con comportamientos irregulares, lo que hizo que estos modelos no pudieran generalizar bien ni capturar patrones reales.
- **Clusterización jerárquica con DTW:** Aunque útil para detectar series con formas similares, requiere que dentro de cada cluster los productos se comporten de forma coherente. En la práctica, esto no se cumplió: los grupos eran heterogéneos o demasiado pequeños, lo que llevó a poca robustez en las predicciones.
- **LightGBM:** El mejor resultado obtenido fue 0.267. Analizando luego los datos identificamos que el modelo realizaba demasiado sobreajuste, ya que la técnica de

validación estaba introduciendo datos del futuro en dicha validación. El mejor modelo obtenido finalmente sin sobre ajuste obtuvo un resultado de 0.31 en el public, el cual se encontraba por debajo del promedio de los 12 meses. El resultado seguramente sea consecuencia de no poder generalizar correctamente los resultados para todos los <client_id, product_id>.

- **El modelo elegido, ¿En qué consiste y porqué?**

El modelo final elegido no fue un único algoritmo, sino una estrategia de selección personalizada por producto. Para cada product_id, entrenamos varios modelos (AutoGluon, Regresión Lineal con “productos mágicos” y promedio simple) utilizando datos hasta octubre de 2019, y luego evaluamos cuál de ellos predecía mejor el valor real de diciembre 2019. Ese modelo fue el que se usó finalmente para predecir febrero de 2020, bajo el supuesto de que “siendo el modelo que mejor performa para diciembre, logra predecir mejor relativamente para esa serie” .

Si bien esta metodología no generó el mejor puntaje en el public leaderboard, nos pareció la más robusta y adaptable a la heterogeneidad de productos, dado que cada serie presenta patrones diferentes que favorecen modelos distintos. Además, todos los modelos considerados en la selección superan al promedio de referencia en la métrica pública, lo cual reforzaba la validez de la estrategia.

La participación de cada modelo, en el submit final fue la siguiente:

Participacion	
Modelo	modelo final
Regresion	42.82%
Auto Gluon	33.72%
Promedio	23.46%

Siendo la regresión el modelo con mayor participación.

En resumen, nuestro modelo final fue un ensamble selectivo por producto, que replica en febrero 2020 el algoritmo que mejor funcionó para predecir diciembre 2019, priorizando la estabilidad del enfoque por sobre el sobreajuste al leaderboard.

Link al repositorio: <https://github.com/dylanbarbona/labo3-2025v/tree/main>