

## **Laboratorio de implementación III**

### **TP FINAL**

## **“Desafío Forecasting Multinacional”**

**Alumnos:**

**Mariano Santos**

**Nicolás Wicky**

## 1. Introducción al desafío

En este Trabajo Práctico abordaremos el desafío de forecasting de ventas de una multinacional de consumo masivo, que comercializa cerca de 1233 SKUs en las categorías de alimentación, limpieza y cuidado personal. Nuestro objetivo es predecir, cada mes, las toneladas vendidas de cada SKU para el mes de Febrero 2020 (datos disponibles hasta Diciembre 2019, ergo Mes+2), empleando como métrica de evaluación el **TotalForecastError**:

$$\text{TotalForecastError} = \frac{\sum_{i=1}^{780} |\text{venta real}_i - \text{predicción}_i|}{\sum_{i=1}^{780} \text{venta real}_i}$$

El proceso operativo exige generar y enviar la predicción por correo antes de las 9 h del día 2 luego del cierre de cada mes. Este ritmo debe ajustarse al ciclo productivo-logístico de la compañía (60 días de stock in trade y vencimiento a 6 meses) y al esquema de facturación al salir el camión de planta (pagos a 90 días).

Aunque contamos con un market share dominante (50 %-85 %), la heterogeneidad de SKUs (desde decenas hasta miles de toneladas mensuales) y la volatilidad de eventos puntuales (promociones, rupturas de stock, cambios de precio o contextos macroeconómicos) plantean un entorno de alta complejidad.

## 2. Resumen Ejecutivo inicial.

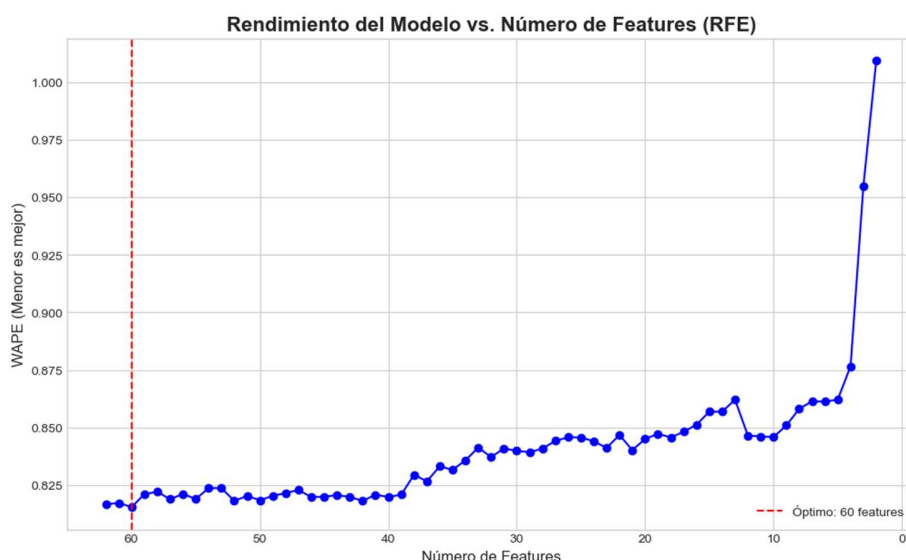
El mejor resultado en el leaderboard público de Kaggle fue 0.274, con un modelo LightGBM con una función objetivo Tweedie, optimizado a través de un proceso jerárquico de varias fases para asegurar la simplicidad, robustez y poder predictivo del modelo final. (Ver más detalles en punto 5).

## 3. Descripción del equipo y estrategias

Equipo compuesto por dos integrantes. Inicialmente dividimos las estrategias: uno de los integrantes optó por utilizar el agrupado SKU (780 predicciones), mientras que el otro integrante optó por hacer predicciones SKU/Clientes, generando una compleja y voluminosa cantidad de predicciones y optimizaciones.

Este segundo abordaje es el que ha brindado la propuesta final, luego de experimentar con más de 10 estrategias distintas (con varias-sub estrategias y ajustes de parámetros en cada caso), y con cerca de 160 intentos en Kaggle entre ambos participantes. Elegimos la predicción detalla en el punto anterior por dos motivos:

- Ha brindado uno de los mejores resultados que obtuvimos en el público de Kaggle.
- Lo consideramos una propuesta robusta ya que se ha generado a partir de un modelo LGBM con optimización de parámetros utilizando Optuna, y re-optimizado sobre los 40 parámetros que brindaron el mejor balance WAPE/#feautures (ver gráfica) después una eliminación recursiva de características (RFE).



En el anexo I se hace una breve descripción de los modelos con los que experimentamos, pero que finalmente descartamos.

Adicional a los datos nativos se compilaron e incluyeron las siguientes variables exógenas:

- Inflación.
- Inflación mes anterior.
- Cotización dólar blue promedio del mes.
- Cotización dólar blue fin de mes.
- Cotización dólar oficial promedio del mes.
- Cotización dólar oficial fin de mes.
- Brecha cambiaria.

Estos parámetros no demostraron poder predictivo al realizar un Feature importance, por lo que se descartaron.

#### 4. Preparación del dataset.

El Script adjunto es pipeline de ingeniería de características. Su objetivo principal es tomar datos de ventas crudos y transformarlos en un archivo (data\_258.csv) enriquecido con decenas de variables informativas, listo para ser usado en el entrenamiento de un modelo de machine learning para predicción de demanda.

El proceso se puede resumir en:

1. Cargar y unificar los datos de ventas, productos e indicadores.
2. Crear una grilla completa de producto-cliente-mes para asegurar que no falten períodos.
3. Filtrar para quedarse solo con los clientes más importantes de cada producto.
4. Generar features temporales, de calendario y de comportamiento.
5. Guardar el dataset final.

#### Listados Bases (Archivos de Entrada)

- sell-in.csv: Es el archivo principal que contiene el historial de transacciones. Incluye product\_id, customer\_id, la fecha (periodo o date) y la cantidad vendida (tn).

- `product_id_apredcir201912.csv`: Una lista que especifica qué productos son relevantes para el problema, es decir, sobre cuáles se quiere hacer una predicción. El script filtra el `sell-in.csv` para trabajar únicamente con estos productos.
- `indicador.csv`: Contiene variables exógenas o indicadores económicos (ej: inflación, tipo de cambio, etc.) a nivel mensual. Se usa para añadir contexto macroeconómico a los datos.

## El Filtrado (Pareto 95%)

Esta es una de las partes más importantes del script. Su objetivo es reducir el ruido y el tamaño del dataset enfocándose en los clientes que generan la mayor parte del volumen de ventas para cada producto, siguiendo el principio de Pareto (el 80% de los efectos provienen del 20% de las causas).

El proceso consiste en:

1. Calcular el volumen total (tn) vendido a cada cliente para cada producto durante todo el año 2019.
2. Para cada producto, ordenar sus clientes de mayor a menor según el volumen de ventas.
3. Calcular el porcentaje acumulado de ventas que representa cada cliente en esa lista ordenada.
4. Seleccionar y conservar únicamente a los clientes que, en conjunto, representan el 95% superior del volumen de ventas (`THRESHOLD = 0.95`).

El resultado es un dataset que ignora a los clientes esporádicos o de muy bajo volumen, permitiendo que el modelo se concentre en las relaciones producto-cliente más significativas.

## 5. Descripción del modelo predictivo seleccionado:

### 1. Ingeniería de Características (Feature Engineering)

El proceso comienza leyendo los datos crudos (`sell-in.csv`, etc.) y realizando una ingeniería de características exhaustiva.

- **Creación de una Grilla Continua:** Genera una base de datos con todos los meses para cada par producto-cliente, rellenando los meses sin ventas con cero para tener una serie de tiempo continua.
- **Filtro Anti-Fugas:** Identifica a los clientes que representan el 95% del volumen de ventas basándose únicamente en datos de 2019. Luego, aplica este filtro a todo el historial (2017-2019). Esta es una técnica muy robusta para evitar usar información futura (data leakage).
- **Creación de Múltiples Features:** Añade una gran cantidad de características, incluyendo:
  - **Temporales:** Lags, deltas y medias móviles (`tn_lag_1`, `tn_roll_mean_3`, etc.).
  - **Macroeconómicas:** Datos de inflación y cotización del dólar.
  - **Estacionales:** Banderas para mes, año, trimestre y estaciones.
  - **Por Quintil:** Características complejas que suman las ventas de los clientes agrupados por quintiles de importancia.
- **Resultado:** Guarda este set de datos enriquecido en `data_258.csv`, que sirve como base para el modelo principal.

## 2. Optimización de Hiperparámetros y Features

Con los datos listos, el notebook realiza dos procesos de optimización clave:

- **Optimización de Hiperparámetros (Optuna):** Ejecuta 50 pruebas con Optuna para encontrar la mejor combinación de hiperparámetros para un modelo LightGBM, minimizando el error **WAPE**. Los mejores parámetros se guardan en `best_gbm_params_tweedie.pkl`.
- **Selección de Características (RFE):** Realiza una Eliminación Recursiva de Características, probando el rendimiento del modelo con un número decreciente de variables (desde 62 hasta 20). Este proceso, que se guarda en `rfe_checkpoint.pkl`, permite identificar cuál es el número óptimo de features. El

análisis concluye que el número óptimo es **60**, pero luego se decide usar **40** como un balance entre rendimiento y simplicidad.

### 3. Modelo Principal (95% del Volumen)

Esta es la parte central del pronóstico:

- **Selección Final de 40 Features:** Carga el historial del RFE (rfe\_checkpoint.pkl) y selecciona programáticamente la lista exacta de las 40 características más importantes.
- **Entrenamiento y Predicción:** Entrena un modelo LightGBM final usando las 40 features seleccionadas y todos los datos históricos disponibles. Luego, genera la predicción para el período futuro (mes 38) y la guarda en kaggle\_lgbm\_submission\_40\_features.csv.

### 4. Modelo para la "Cola" (5% Residual)

Para el 5% de datos de bajo volumen que se filtró inicialmente, se sigue una estrategia diferente y más simple:

- **Aislamiento de la Cola:** Se identifican los pares producto-cliente que no pertenecen al 95% principal.
- **Micro-Modelo LGBM:** Se entrena un **modelo LightGBM separado y más simple** con muy pocas características (mes, tn\_lag\_1, etc.) usando únicamente los datos de la cola.
- **Predicción de la Cola:** Este micro-modelo genera un pronóstico específico para este segmento y lo guarda en prediccion\_cola\_micromodelo\_lgbm\_v2.csv.

### 5. Ensamblaje Final

El último paso es unir los dos resultados:

- **Suma de Predicciones:** El script carga las predicciones del modelo principal y las del modelo de la cola.
- **Agregación:** Para cada producto, suma las dos predicciones (tn del 95% + tn del 5%) para obtener el pronóstico total.
- **Resultado Final:** Guarda el archivo consolidado en kaggle\_lgbm\_final.csv, listo para la submission.

#### Se adjuntan:

Archivos

Modelo-Santos-Wicky\_LGBM.ipynb

indicador.csv (datos macroeconómicos para correr el script anterior).

Adicional: Modelo-Santos-Wicky\_TimeFM.ipynb (sometido por uno de los integrantes en Kaggle).

Anexo I: Detalle de otros modelos ensayados.

Anexo II : Análisis adicional sobre el poder predictivo del promedio

## 1. Anexo I: Análisis adicional sobre el poder predictivo del promedio

Esto son algunos de los modelos que se probaron utilizando los 780 productos (sin desglosar por cliente), pero que se han descartado:

### A. Línea base: Media móvil ajustada

$$\hat{y}_{p,dic19} = \alpha \times \frac{1}{n} \sum_{i=1}^n tn_{p,t-i}$$

- Ventanas probadas:  $n \in \{6, 12, 36\}$  Meses.
- Factor de ajuste:  $\alpha \in [0.6, 0.8]$  M

### B. Modelos globales de boosting.

#### XGBoost + Optuna

$$XGBRegressor(\theta^*), \theta^* = \frac{\arg \min MAE}{\theta}$$

- Hiperparámetros:  $\{n\_estimators, max\_depth, \backslash learning\_rate, \dots\}$
- Early stopping sobre validación sep-oct 2019

#### LightGBM sklearn API

LGBMRegressor(params) con verbose = -1, early\_stopping\_rounds = 20

- Parámetros estándar:  $\backslash learning\_rate = 0.05, \backslash num\_leaves = 31$

### C. Modelos por producto (p)

Para cada p:

$$f_p = \text{LGBMRegressor}_p(X_{p, \leq \text{oct19}}, tn_{lead2m}) \Rightarrow \hat{y}_{p,dic19} = f_p(X_{p, \text{oct19}})$$

- Captura comportamientos individuales
- Guardado/recarga de 780 modelos .pkl

### D. AutoGluon (ensembles automáticos)

Se hicieron varias corridas del script compartido en clase, y se sometieron individualmente y en ensemble de 4 predicciones.

### E. Señales e ingeniería de características

- *Trigonométricas:*  
 $\sin\left(2\pi \frac{mes}{12}\right), \cos\left(2\pi \frac{mes}{12}\right)$
- *Tendencia lineal:*  
time\_idx=cumcount() por p.
- *Cambios interanuales:*

$$\Delta_{YOY} = \frac{tn - tn_{t-12}}{tn_{t-12}}.$$

- *Rolling statistics:* medias, desviaciones, máximos, mínimos sobre 3, 6, 12, ... 36 meses.
- Lags puros:  $tn_{t-1}, tn_{t-2}, \dots, tn_{t-12}$

Entre los modelos con los que experimentamos usando la combinación Cliente/producto, podemos nombrar:

#### ***F. Modelo TimesFM con Fine-Tuning***

Para generar pronósticos de series temporales, se implementó una metodología basada en transferencia de aprendizaje utilizando el modelo preentrenado TimesFM-2.0-500M de Google. Este modelo, entrenado previamente en cientos de millones de series temporales, ya posee un conocimiento profundo sobre patrones comunes como estacionalidades y tendencias, lo que permite partir desde una base experta en análisis temporal en lugar de entrenar desde cero.

Antes de especializar el modelo, se realizó un preprocesamiento clave del conjunto de datos `data_chronos.csv`, que contiene información de ventas de 780 productos. Para evitar que las diferencias de magnitud influyan en el aprendizaje, cada serie fue escalada individualmente mediante `StandardScaler`, asegurando que cada producto tuviera media 0 y desviación estándar 1. Estos escaladores se guardaron para revertir luego las predicciones a su escala original.

Luego, se ejecutó un proceso de fine-tuning: se continuó el entrenamiento del modelo preentrenado, usando únicamente los datos escalados del problema específico. Tras 5 épocas de ajuste, el modelo se especializó en las particularidades de las series de estos productos. Para la predicción final, se usaron los últimos 15 datos de cada serie, se generaron pronósticos escalados, y luego se aplicó el `inverse_transform` correspondiente para devolverlos a su escala real.

Uno de los integrantes del equipo seleccionó en Kaggle este resultado 0.27 para la competencia.

#### ***G. Chronos Bolt (Zero Shot, 205M parámetros, modelo “base”)***

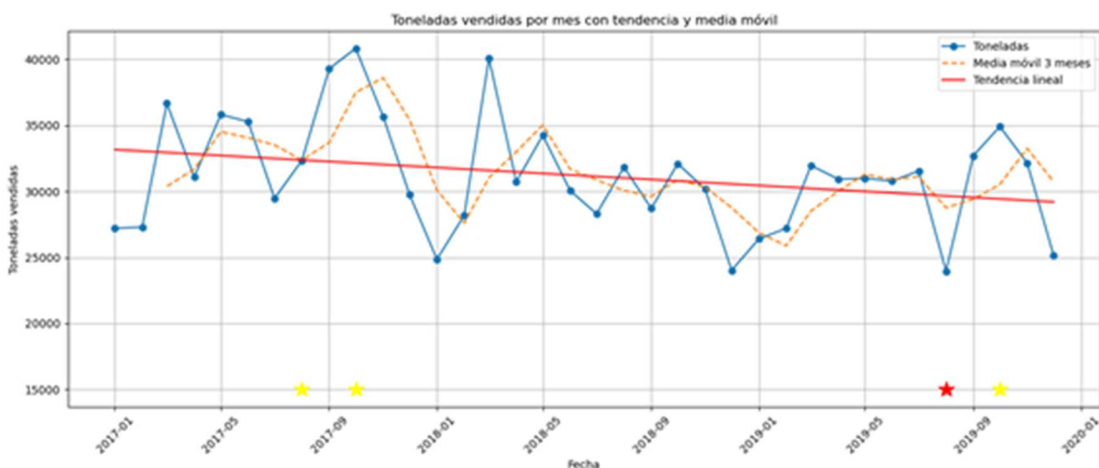
Se utilizó Chronos Bolt, un modelo de series temporales desarrollado por Amazon con 205 millones de parámetros, para predecir ventas mensuales de 780 productos (a nivel de `product_id`, agregando `customer_id`). El modelo se aplicó en modo zero-shot, lo que significa que genera pronósticos sin haber sido ajustado con los datos específicos del problema (sin fine-tuning). Utiliza pesos preentrenados y no requiere entrenamiento adicional. En la implementación no se incorporaron co-variables y se usaron los parámetros por defecto del modelo.

El desempeño en la competencia de Kaggle fue consistente: tanto la versión original como la variante con `num_val_window: 2` lograron un WAPE de 0.276. Actualmente, los pesos del modelo no están disponibles para realizar fine-tuning, lo que limita su adaptación a contextos particulares.

## 2. Anexo II: Análisis adicional sobre el poder predictivo del promedio

A modo de ejemplo, sólo se utilizaron los valores totales, pero podría aplicarse al comportamiento general de varios productos (no todos, debido a sus características de madurez, decisión estratégica y demanda).

En el gráfico que considera los 36 meses se observa (independiente de la situación electoral atípica marcada con una estrella roja) Periodos de mayor regresión a la media (05-18 a 09-18//03-19 a 07-19) y periodos de picos y valles alejados de la media(10-17; 12-18).



También hemos visto con ejemplos que dentro de los comportamiento de demanda de clientes para un productos existe varios comportamientos:

Demanda estable y constante.

Demanda errática, pero manteniendo cierta centralidad (varios valores de demanda mensuales, con picos y valles que se asemejan a esa demanda si se promedian) (potenciales sobre-stockeo para cubrir una demanda constante).

Demanda completamente errática.

**Por qué el promedio es un buen predictor para Feb-2020? Posiblemente por:**

- Circunstancial demanda de febrero con una gran regresión a la media (ver gráfico siguiente).





Los cambios históricos vs el mes anterior no han sido mayores al 15% vs enero:

fecha	tn	variacion_pct vs Enero
01/01/2017	2.719.898.239	NaN
01/02/2017	2.730.419.010	0,387%
01/01/2018	2.485.668.239	-16,586%
01/02/2018	2.818.520.277	13,391%
01/01/2019	2.640.614.976	9,896%
01/02/2019	2.719.965.581	3,005%

Y si consideramos vs diciembre (2 casos calculables) no solo no cambian, sino que hasta se suavizan:

fecha	Variación vs enero	Variación vs dic
01/02/2017	0,387%	N/A
01/02/2018	13%	-5%
01/02/2019	3,005%	13,197%

Comportamiento de demanda aleatorio de los distintos cliente que compensa picos de uno con valles de otros, desfases entre consumo de stock de unos y sobrestockeo de otros.

**Por qué a largo plazo el promedio solo no representa un buen modelo predictivo?:**

- Porque por sí solo no captura la estacionalidad mensual.
- Porque para progresivamente las compensaciones realizadas entre las demandas de los distintos clientes (picos y valles; consumo de stock vs sobrestockeo) se van a ir desfasando y perdiendo la compensación entre ellos se podrían disparar los errores de predicción.

**Se debería descartar totalmente el promedio como herramienta predictiva?.**

- No. Para productos con demanda constante y robusta, el promedio puede ser una herramienta simple y efectiva, si con algunos ajustes podemos moldearla a esos productos sin estacionalidad, ni picos de demanda.
- También puede ser de mucha utilidad cuando queremos modelar la oferta y planificar una estrategia de discontinuación de producto, racionalización para suavizar el abastecimiento anticipando futuras rupturas de stock y cualquier situación en la que la empresa tiene control sobre las ventas.