UNIVERSIDAD AUSTRAL | INGENIERÍA

# MAESTRÍA EN EXPLOTACIÓN DE DATOS Y GESTIÓN DEL CONOCIMIENTO

## LABORATORIO III

## Trabajo Final


AUTORES:

- Leonel Ruppel
- Pablo Bazzano
- Hector Ariel Baez

# 1. Introducción

El presente trabajo se enmarca en el área de consumo masivo en una empresa multinacional con un fuerte presencia de mercado y para lo cual se aborda el desafío de predecir la cantidad de toneladas vendidas por producto (tn) para 780 artículos de distintas categorías y características, con el uso datos históricos comprendidos desde enero 2017 a diciembre 2019 siendo el propósito predecir el mes +2 es decir febrero 2020. Teniendo en cuenta esto el principal objetivo  fue generar un archivo con 780 predicciones, una por cada “product_id” requerido, en un formato compatible con una competencia tipo Kaggle. La consigna del proyecto implicó no solo construir modelos predictivos competitivos en términos de precisión, sino también navegar las particularidades de un entorno económico argentino volátil, caracterizado por quiebres estructurales y estacionalidades difusas asi como permitir generar un modelo predictivo escalable. La competencia en Kaggle sirvió como marco comparativo y medidor final, empleando el error relativo sobre el total vendido como métrica.

# 2. Hipótesis Experimental

La construcción del modelo final se basó  principalmente en varias etapas que ayudaron a una comprensión más profunda en integral del contexto y datos disponibles.

- Creación de perfil de compañía y caso de estudio.
- Definición de estrategia.

### Creación de perfil compañía y caso de estudio

Se delimitó la compañía de estudio como una compañía con poca maduración de consumo de datos y aplicación de técnicas propias de ciencia de datos por lo que la calidad de los datos se supone se verá comprometida, basado en la información de las entrevistas con los representantes de la compañía el contexto la información provista en los archivos tb_stocks, tb_productos, sell-in y productos a predecir son todos los disponibles para el estudio requerido.

**Definición de estrategia**

Aunado a esto se procedio al analisis exploratorio de datos donde se logro definir la hipótesis principal la cual se centró en que la transformación logarítmica de la variable objetivo y el uso de un modelo como LightGBM con objetivo Tweedie serían cruciales para manejar la distribución sesgada de las ventas y los productos con valores de cero o bajos. Además, se exploró si la regresión directa (sin log) podría ser más efectiva para el WMAPE.

## 3. Diseño Experimental

El diseño experimental siguió una lógica progresiva, fundamentada en la comprensión de la estructura del problema y en una exploración empírica de las técnicas más adecuadas para este tipo de series. El proceso completo se estructuró en las siguientes etapas:

1. **Análisis Exploratorio de Datos (EDA)**: Se llevó a cabo un estudio de tendencias generales, comportamiento de productos, identificación de anomalías en el tiempo y estructura de faltantes para ayudar a diseñar variables que luego pudieran ser *output* del modelo y mejoraran el entendimiento del mismo de las series con las que se estaba trabajando.
2. **Preprocesamiento de datos:** posterior a la exploración preliminar de los datos se cargaron los datasets de ventas (sell-in.txt.gz), stock (tb_stocks.txt), y productos (tb_productos.txt). Se realizó un merge de los datos de ventas con la información de productos para incorporar categorías y tamaño de SKU. Luego, se agruparon las ventas por período, product_id y características del producto, sumando cust_request_tn y tn.
3. **Filtrado de productos:** Se realizó un inner join con una lista de product_id a predecir, reduciendo el dataset a productos relevantes para la competencia.
4. **Fechas y antiguedad:** La columna periodo se convirtió a formato datetime y se calculó la antiguedad_meses para cada producto desde su primera aparición.
5. **Inclusión de stock:** Se realizó un merge con los datos de stock (stock_t0) de la tabla tb_stocks, asegurando que el stock correspondiera al inicio del mes de predicción.

6. **Feature Engineering (FE):** Se implementó una función feature_pipeline para generar una amplia gama de características, incluyendo:
    - Variables de calendario: month, quarter, month_sin, month_cos.
    - Lags de ventas (tn): Hasta 24 lags (lag_1 a lag_24).
    - Delta-lags de ventas: Diferencias entre lags consecutivos (delta_lag_1 a delta_lag_23).
    - Medias móviles de ventas: Promedios de tn para ventanas de 2, 3, 4, 6 y 10 meses (tn_media_2, etc.).
    - Features de stock: flag_stock_disp (indicador de disponibilidad de stock) y ratio_tn_stock (relación entre lag_1 y stock_t0).
    - Features interanuales (YoY): delta_yoy y ratio_yoy (diferencia y ratio de lag_1 respecto a lag_13).

7. **Codificación de categóricas:** Las características categóricas (cat1, cat2, cat3, brand, month, quarter) se tiparon como category en LightGBM. Para la predicción, se utilizó una codificación a enteros (int32) basada en los niveles observados en el conjunto de entrenamiento.
8. **Preparación de splits:** El dataset se dividió en conjuntos de entrenamiento (hasta octubre 2019), validación (noviembre 2019) y prueba (diciembre 2019).
9. **Ensamble final:** La predicción final por producto se obtuvo combinando las predicciones de los modelos.

## 4. Modelos Probados y Resultados

Se probaron diversos modelos, desde enfoques estadísticos clásicos hasta algoritmos de Gradient Boosting, evaluando su rendimiento utilizando MAE, MAPE y WMAPE sobre el conjunto de pruebas de diciembre de 2019.

### 4.1 Enfoques Iniciales y Modelos Base:

- **Modelos Estadísticos Clásicos (ARIMA, Auto ARIMA, SARIMAX):** Se realizaron pruebas con modelos estadísticos clásicos de series de tiempo como ARIMA, Auto ARIMA y SARIMAX. Los resultados obtenidos con estos modelos no fueron satisfactorios, mostrando limitaciones para adaptarse a la discontinuidad de muchas series y a la alta proporción de productos con ventas esporádicas.
- **LightGBM Base (tn_log):** Un modelo estándar con hiperparámetros básicos.
    - Resultados en Diciembre-19: MAE: 8.60, WMAPE: 26.67%.

- **LightGBM Tuned (tn_log):** Un modelo con hiperparámetros ajustados manualmente.
    - Resultados en Diciembre-19: MAE: 8.45, WMAPE: 26.20%.

**4.2 Modelos con y sin Transformación Logarítmica:**

Se realizaron experimentos clave con el objetivo tweedie en LightGBM, explorando diferentes potencias (p) para el término de varianza, y comparando los resultados con y sin la transformación logarítmica del target.

- **LightGBM Tweedie con tn_log:** Se probó p entre 1.1 y 1.5. El mejor p encontrado fue 1.2.
    - Resultados en Diciembre-19: MAE: 8.00, WMAPE: 24.80%.

- **LightGBM Optuna con tn_log:** Se realizó una búsqueda de hiperparámetros con Optuna, optimizando el WMAPE.
    - Mejores hiperparámetros: p=1.097, lr=0.048, num_leaves=96, min_child_samples=130, feature_fraction=0.77, bagging_fraction=0.84, lambda_l2=0.10.
    - Resultados en Diciembre-19: MAE: 8.19, WMAPE: 25.40%.

- **LightGBM Base sin log (tn directamente):** Se entrenó un modelo sin la transformación logarítmica.
    - Resultados en Diciembre-19: MAE: 10.22, WMAPE: 31.69%.

- **LightGBM Tuned sin log (tn directamente):** Se entrenó un modelo con hiperparámetros ajustados manualmente, sin la transformación logarítmica.
    - Resultados en Diciembre-19: MAE: 10.26, WMAPE: 31.84%.

- **LightGBM Tweedie sin log (tn directamente):** Se probó p entre 1.1 y 1.5. El mejor p encontrado fue 1.4.
    - Resultados en Diciembre-19: MAE: 8.51, WMAPE: 26.39%.

- **LightGBM Optuna sin log (tn directamente):** Se realizó una búsqueda de hiperparámetros con Optuna, optimizando el WMAPE.
    - Mejores hiperparámetros: p=1.25, lr=0.034, num_leaves=88, min_child_samples=120, feature_fraction=0.83, bagging_fraction=0.77, lambda_l2=0.71.
    - Resultados en Diciembre-19: MAE: 8.49, WMAPE: 26.33%.

### 4.3 Evaluación Comparativa Final de Modelos en Diciembre-19:

Tras un ajuste final en la evaluación para asegurar la consistencia en el preprocesamiento de las características y la aplicación de las transformaciones de escala, se obtuvieron los siguientes resultados:

| Modelo | MAE | MAPE | WMAPE |
|---|---|---|---|
| BASE | 7.79 | 211.39% | 24.15% |
| TUNED | 8.27 | 222.78% | 25.65% |
| TWEED | 7.61 | 210.64% | 23.60% |
| FINAL | 8.63 | 183.18% | 26.76% |
| Base Sin log | 29.22 | 7480.32% | 90.65% |
| TUNED sin log | 27.39 | 5876.63% | 84.96% |
| TWEED sin log | 8.26 | 386.38% | 25.63% |

## 5.Lo que no funcionó

- **Modelos Estadísticos Clásicos (ARIMA, Auto ARIMA, SARIMAX):** Estos modelos no lograron captar la complejidad y la discontinuidad de las series de ventas, arrojando un mal desempeño en comparación con los modelos de machine learning.
- **Modelos de regresión directa (sin log1p) sin Tweedie:** Los modelos que no utilizaron la transformación logarítmica de la variable objetivo y no emplearon una función de pérdida tweedie tuvieron un rendimiento significativamente inferior en la métrica WMAPE. Esto sugiere que la naturaleza sesgada de los datos de ventas se maneja mejor con transformaciones o funciones de pérdida especializadas.
- **Algunas configuraciones de hiperparámetros en Optuna:** A pesar de la optimización automática, algunas combinaciones de hiperparámetros exploradas por Optuna no resultaron en mejoras significativas o incluso llevaron a un rendimiento subóptimo, lo que es inherente a los procesos de búsqueda.

## 6. Modelo Elegido como Final en Kaggle

El modelo final para la predicción de febrero de 2020 fue un Ensamble Selectivo, que combina las predicciones de los modelos individuales generados. En lugar de una ponderación fija, este ensamble operó eligiendo la mejor predicción para cada product_id a partir de un conjunto de modelos candidatos. Esto implicó una etapa post-modelado donde, para cada producto, se compararon las predicciones de los modelos y se seleccionó aquella que históricamente (o en la fase de validación) demostró el menor error para ese producto o segmento.

**Razones de la elección:**

1. **Optimización por Producto:** El enfoque de ensamble selectivo nos permitió abordar la heterogeneidad inherente a los productos de consumo masivo. No todos los productos se comportan igual, y un único modelo o una ponderación fija puede no ser óptima para todos. Al seleccionar la "mejor" predicción por producto, se maximiza la precisión a nivel individual, lo que repercute directamente en una mejora del WMAPE general.
2. **WMAPE Mejorado:** Al comparar los resultados de los modelos individuales y del ensamble, se observa que el enfoque selectivo permite capitalizar las fortalezas de cada modelo para diferentes productos, resultando en un error total más bajo. La capacidad de elegir la predicción más precisa para cada SKU contribuye directamente a minimizar el Error Relativo sobre el Total Vendido, métrica clave de la competencia.

3. **Flexibilidad y Adaptabilidad:** Este tipo de ensamble es intrínsecamente más flexible, ya que no se compromete con un único algoritmo o una combinación estática. Permite que el sistema se adapte a las particularidades de cada serie de tiempo de ventas.
4. **Aprovechamiento del Portafolio de Modelos:** Al haber generado diversos modelos (LightGBM con y sin log, Tweedie, Optuna), este ensamble saca el máximo provecho de todo el esfuerzo de modelado. Cada modelo, aunque no sea el "mejor" globalmente, puede tener un rendimiento superior para un subconjunto específico de productos.

## 7. Lecciones Aprendidas

- **La Heterogeneidad Demanda Enfoques Diferenciados:** La lección más importante fue que un "talle único" no funciona para la predicción de ventas en un portafolio de productos diverso. Los modelos simples (como la regresión lineal o promedios) funcionaron mejor para algunos productos, mientras que los modelos más complejos (LightGBM) o con diferentes transformaciones (logarítmica vs. sin log, Tweedie) destacaron en otros. La clave residió en la capacidad de identificar y aplicar la mejor estrategia a cada segmento o incluso a cada producto individualmente.

- **El Feature Engineering es Crucial, no la Complejidad Bruta:** La creación de variables significativas que capturan la estacionalidad, las tendencias y la interacción con el stock resultó ser más determinante que la elección de algoritmos extremadamente complejos. Añadir más características de forma indiscriminada a modelos simples a menudo resultaba en un deterioro del rendimiento, reforzando la idea de que la calidad de las features supera la cantidad.

- **La Transformación del Target y Funciones de Pérdida Adecuadas son Fundamentales:** La distribución sesgada de las ventas hizo que la transformación logarítmica (log1p) y la función de pérdida tweedie fueran herramientas esenciales. Los modelos que no las utilizaron tuvieron dificultades significativas para optimizar la métrica WMAPE, demostrando la importancia de adaptar el modelo a la naturaleza estadística de la variable objetivo.

- **Los Modelos Estadísticos Clásicos Tuvieron Limitaciones:** A pesar de su utilidad en otros contextos, modelos como ARIMA, Auto ARIMA y SARIMAX no se adaptaron bien a la discontinuidad y volatilidad de muchas de las series de ventas, resultando en un rendimiento inferior en comparación con los enfoques basados en Gradient Boosting.

## 8. Conclusiones

El enfoque implementado demuestra ser prometedor para resolver un problema de forecasting complejo en un entorno realista de consumo masivo. A través de un proceso iterativo que incluyó una robusta ingeniería de características y la experimentación con diversos modelos de LightGBM, se logró una predicción competitiva. La implementación de un Ensamble Selectivo, que permitió capitalizar las fortalezas de los modelos individuales al elegir la mejor predicción para cada product_id. Este enfoque supone una adaptación más eficaz a la heterogeneidad de las series de ventas. Se reafirma así la importancia de un Feature Engineering de calidad y la selección de herramientas de modelado flexibles que puedan manejar las particularidades de los datos reales.