Informe Técnico – Predicción de Ventas para Kaggle (Febrero 2020)

Autor: Waldo Griffiths

Fecha: 11/07/2025

# 1. Introducción

A comienzos del proyecto nos propusimos predecir la cantidad de toneladas vendidas por producto (tn) para el mes de febrero de 2020, utilizando como base los datos históricos comprendidos entre enero de 2017 y diciembre de 2019. El objetivo era generar un archivo con 780 predicciones, uno por cada product\_id requerido, en el formato compatible con una competencia tipo Kaggle.

**1. Exploración y Construcción del Dataset Inicial**

El punto de partida del proyecto fue una fase de exploración profunda de los datos disponibles. Antes de cualquier modelado, nos enfocamos en entender la naturaleza y estructura del conjunto original. La información provenía de archivos separados que contenían registros de ventas, productos y stocks. En particular, los archivos relevantes fueron sellin.txt, tb\_productos y tb\_stocks. La fusión de estos tres archivos mediante operaciones de merge permitió la construcción del primer dataset integral sobre el cual trabajar.

En esta etapa, se tomaron decisiones clave sobre el formato y las columnas que debían conservarse, las claves de unión más adecuadas, y el tratamiento de valores faltantes o inconsistencias de tipo. Este dataset inicial tenía aproximadamente 17 millones de filas y representaba ventas mensuales para el período completo 2017-2019, cubriendo distintas combinaciones de customer\_id y product\_id.

Una vez consolidado el dataset, realizamos un **Análisis Exploratorio de Datos (EDA)** utilizando diversas herramientas automáticas para acelerar el descubrimiento de patrones:

* **AutoViz**: Permitió obtener una vista rápida de las distribuciones, correlaciones y outliers.
* **DataPrep**: Brindó perfiles detallados por columna, valores nulos, tipos de datos y sugerencias de limpieza.
* **Sweetviz**: Fue particularmente útil para comparar subconjuntos de datos (por ejemplo, productos activos vs. inactivos).

Complementariamente, desarrollamos una app con **Streamlit** que nos permitió visualizar de forma interactiva el comportamiento mensual de clientes y productos. Esta herramienta fue clave para validar visualmente lo que sugerían las métricas: muchos clientes mostraban comportamientos repetitivos, caracterizados por la compra frecuente de los mismos productos o por largos períodos de inactividad.

A partir de esta observación empírica y también a partir de lo que se discutió en clase, tomamos la primera gran decisión técnica del proyecto: **reducir el volumen del dataset**, pero de forma informada y justificada. Se filtraron todos los pares de customer\_id y product\_id cuya variable tn (toneladas) permanecía en cero a lo largo de los 36 períodos considerados. La lógica fue simple: si durante tres años completos no se registraron ventas de un producto por parte de un cliente, la probabilidad de que esa combinación arroje un valor distinto a cero en el futuro era extremadamente baja, salvo por cambios drásticos que el modelo no podría anticipar con evidencia.

Esta decisión permitió reducir el dataset de 17 millones de registros a **aproximadamente 9.5 millones de filas**, sin perder información relevante para la predicción. Este nuevo conjunto fue más liviano, más manejable en memoria, y sentó las bases para un entrenamiento más eficiente de modelos posteriores.

Este recorte selectivo no fue trivial: implicó rehacer algunos índices, asegurar la consistencia temporal del dataset filtrado y ajustar las estrategias de validación para evitar sesgos en el entrenamiento. Sin embargo, sentó un precedente metodológico importante en el proyecto: **las decisiones de depuración de datos deben estar basadas en observaciones empíricas y criterio técnico, no solo en consideraciones computacionales**.

Con esta versión reducida, limpia y estructurada del dataset, se dio por concluida la primera etapa del trabajo, que resultó fundamental para todo lo que vendría luego: la ingeniería de features, la segmentación por deciles y clusters, y el entrenamiento de modelos predictivos.

**2. Generación de Variables y Primeros Modelos**

La siguiente etapa del proyecto consistió en la automatización de la ingeniería de variables utilizando la biblioteca **Featuretools**, que permitió generar nuevas características a partir de relaciones entre entidades, normalizadas por product\_id y customer\_id, con venta\_id como índice y periodo como time\_index. Esta herramienta facilitó la creación de transformaciones y agregaciones, produciendo inicialmente un conjunto amplio de más de 300 variables candidatas.

Luego de un proceso de depuración, seleccionamos un subconjunto de **poco más de 100 variables**. El criterio para esta selección se basó en:

* La ausencia de multicolinealidad grave entre variables.
* La estabilidad temporal de las features.
* Su interpretabilidad y relación directa con las ventas.
* La viabilidad computacional para entrenar modelos con bajo riesgo de overfitting.

Una vez definido este dataset enriquecido, realizamos una primera ronda de experimentación con tres enfoques predictivos distintos:

* **AutoGluon**: Framework automatizado que prueba múltiples modelos de ML de forma eficiente.
* **ARIMA**: Modelo clásico de series temporales, aplicado por producto.
* **NeuralProphet**: Variante del modelo Prophet, con capacidad para capturar estacionalidades múltiples y regresores externos.

A pesar del rigor metodológico, los resultados iniciales no fueron satisfactorios. En el caso de AutoGluon, si bien el entrenamiento fue ágil, las métricas de validación (MAE, R²) fueron bajas. Por su parte, ARIMA mostró limitaciones claras para adaptarse a la discontinuidad de muchas series. En cuanto a NeuralProphet, la alta proporción de productos con ventas esporádicas generó resultados planos o nulos en gran parte de los casos.

Estos primeros resultados, aunque decepcionantes desde el punto de vista de la precisión, sirvieron como diagnóstico clave: dejaron en evidencia la necesidad de segmentar los productos según comportamiento histórico, incorporar features adicionales como tendencias y estacionalidades por producto, y repensar la estructura del entrenamiento para evitar predicciones triviales o planas.

**3. Promedios Triviales y Primer Modelo Competitivo**

A partir de lo observado en clase, decidimos probar una serie de enfoques simples conocidos como **predicciones triviales**, basados únicamente en el cálculo de promedios de ventas (tn) recientes. Desarrollamos las siguientes variables agregadas:

* Promedio de los últimos 3, 6, 9 y 12 meses.
* Medias móviles exponenciales (EMA) para los mismos períodos: 3, 6, 9 y 12 meses.

De forma sorpresiva, estas estrategias arrojaron mejores resultados que los modelos anteriormente probados. En particular, el promedio simple de los últimos 12 meses mostró un desempeño robusto y consistente, convirtiéndose en el nuevo baseline del proyecto.

Motivados por estos resultados, desarrollamos un modelo **LightGBM**, esta vez con un pipeline más sólido y búsqueda de hiperparámetros automatizada mediante **Optuna**. El modelo entrenado con los parámetros óptimos superó levemente al mejor promedio (12 meses), lo que validó tanto la elección de variables como el uso de técnicas más complejas para capturar interacciones no lineales.

Este fue el primer modelo que superó de manera consistente la línea base, y por lo tanto, marcó el inicio de una etapa más madura del proyecto, enfocada en consolidar la segmentación, evaluar el impacto de nuevas features y aplicar estrategias de validación más rigurosas.

**4. Segmentación por Participación en Volumen**

A partir del modelo competitivo inicial, se tomó una decisión estratégica clave: **segmentar el dataset según la participación en volumen de ventas (tn)**, considerando los pares product\_id - customer\_id. El criterio de segmentación se basó en el share acumulado de toneladas vendidas, y se establecieron cuatro grupos diferenciados:

* El primer segmento representaba el **40% del volumen total**, conformado por los pares más significativos.
* El segundo segmento abarcaba el **30% siguiente**.
* El tercero, un **20% adicional**.
* El cuarto y último segmento comprendía el **10% restante**, es decir, los pares de menor peso relativo en el total.

Esta partición tuvo un doble objetivo: por un lado, adaptar la estrategia de modelado a la relevancia de cada segmento; por otro, optimizar tiempos de cómputo y entrenamiento sin sacrificar calidad predictiva.

Como decisión operativa, se resolvió entrenar un **modelo LightGBM específico para cada uno de los tres primeros segmentos**, aplicando distintas optimizaciones de hiperparámetros con **Optuna** en cada caso, para capturar con mayor precisión las dinámicas particulares de esos subconjuntos.

Para el cuarto segmento, que agrupaba las combinaciones menos relevantes, se optó por una solución más eficiente: **utilizar directamente el promedio simple de los últimos 12 meses**, estrategia que había demostrado ser sólida y no alteraba significativamente el resultado final. Esto permitió ganar tiempo sin comprometer el rendimiento global del modelo en la predicción de febrero 2020.

**5. Ensamble y Nuevas Estrategias de Segmentación**

Con los modelos entrenados y validados por segmento, exploramos la posibilidad de mejorar aún más la predicción final a través de un **ensamble de resultados**. La estrategia consistió en combinar las predicciones del modelo LightGBM con un promedio interanual específico de los meses de febrero de 2017, 2018 y 2019. Esta media histórica permitía capturar una posible estacionalidad de ese mes en particular, complementando lo que el modelo general no lograba.

Se probaron diferentes ponderaciones entre ambas fuentes, y se observó que una combinación de **60% del valor predicho por LightGBM y 40% del promedio de febreros** alcanzaba el mejor rendimiento en el leaderboard público de Kaggle, con un score de **0.269**. Si bien este resultado se mantenía lejos del óptimo esperado, marcó un punto de referencia competitivo y replicable.

A partir de allí, decidimos **refinar la segmentación del dataset original**, pasando de cuatro a **diez subconjuntos**, manteniendo el criterio de participación acumulada de toneladas vendidas. Cada uno de estos nuevos segmentos fue tratado como un conjunto independiente, al que se le aplicó nuevamente la lógica de entrenamiento con LightGBM y búsqueda de hiperparámetros con Optuna.

Sin embargo, esta ampliación de la granularidad no resultó en mejoras sustanciales. Si bien algunas predicciones estuvieron cercanas a la mejor marca obtenida, **ninguna superó el rendimiento alcanzado con el esquema original de cuatro segmentos**. A pesar del mayor esfuerzo computacional y de ajustes, los resultados indicaron que el beneficio marginal de esta nueva segmentación no justificaba su complejidad.

Finalmente, se realizaron pruebas adicionales con otros algoritmos como **XGBoost** y una **red neuronal optimizada con Keras y Optuna**, pero los resultados obtenidos no superaron a los alcanzados con el modelo LightGBM original ni con las combinaciones de promedios, reafirmando la robustez del enfoque elegido.

**6. Regresión Lineal, AutoGluon Simple y el Retorno a la Simplicidad**

En este punto, surgió una propuesta por parte del profesor: explorar la **regresión lineal** como alternativa. Aplicando un script base con esta técnica, se logró **la mejor marca hasta ese momento**, superando tanto a los modelos complejos como a las combinaciones anteriores.

**6. Regresión Lineal, AutoGluon Simple y el Retorno a la Simplicidad**

En este punto, surgió una propuesta por parte del profesor: explorar la **regresión lineal** como alternativa. Aplicando un script base con esta técnica, se logró **la mejor marca hasta ese momento**, superando tanto a los modelos complejos como a las combinaciones anteriores.

También se probaron variantes de la regresión lineal, como **Lasso**, **Huber** y **RANSAC**, en busca de mejoras adicionales. Sin embargo, estos modelos no lograron superar el desempeño de la regresión lineal clásica, ni en validación ni en la predicción final.

A partir de este hallazgo, se testeó también un nuevo enfoque simplificado, implementado por los compañeros **Iván Parra y Fernando Raco**, basado en un script de **AutoGluon** con configuración mínima. De forma inesperada, este modelo logró **una mejora adicional**, alcanzando el mejor rendimiento hasta la fecha en el leaderboard público de Kaggle, con un score de **0.244**.

Sin embargo, cuando se intentó complejizar este modelo simple —incorporando nuevas features como lags y deltas—, el rendimiento empeoró, lo cual evidenció una constante que ya veníamos observando: **los modelos simples superaban sistemáticamente a los más sofisticados**.

Ante esta situación, se decidió cambiar de enfoque. Inspirados en el dicho "si la montaña no viene a Mahoma, Mahoma va a la montaña", se optó por abandonar la idea de complejizar los modelos y, en su lugar, **simplificar los datasets**. La nueva hipótesis de trabajo fue que los modelos exitosos requerían una estructura de datos más limpia y menos sobrecargada.

Por lo tanto, se rediseñó el pipeline de predicción, esta vez con un dataset **sin un volumen excesivo de variables**, retomando la lógica de segmentación en **cuatro subconjuntos** según la participación en volumen, y aplicando sobre cada uno de ellos las técnicas que mejores resultados habían demostrado: **LightGBM**, **regresión lineal** y el **AutoGluon simple**.

El paso final fue la combinación (ensamble) de estas predicciones, retomando el espíritu del enfoque anterior pero desde una arquitectura más liviana y precisa, alineada con el nuevo principio que había guiado los últimos descubrimientos: **la simplicidad como estrategia central de modelado**.

A partir de este hallazgo, se testeó también un nuevo enfoque simplificado, implementado por los compañeros **Iván Parra y Fernando Raco**, basado en un script de **AutoGluon** con configuración mínima. De forma inesperada, este modelo logró **una mejora adicional**, alcanzando el mejor rendimiento hasta la fecha en el leaderboard público de Kaggle, con un score de **0.244**.

Sin embargo, cuando se intentó complejizar este modelo simple —incorporando nuevas features como lags y deltas—, el rendimiento empeoró, lo cual evidenció una constante que ya veníamos observando: **los modelos simples superaban sistemáticamente a los más sofisticados**.

Ante esta situación, se decidió cambiar de enfoque. Inspirados en el dicho "si la montaña no viene a Mahoma, Mahoma va a la montaña", se optó por abandonar la idea de complejizar los modelos y, en su lugar, **simplificar los datasets**. La nueva hipótesis de trabajo fue que los modelos exitosos requerían una estructura de datos más limpia y menos sobrecargada.

Por lo tanto, se rediseñó el pipeline de predicción, esta vez con un dataset **sin un volumen excesivo de variables**, retomando la lógica de segmentación en **cuatro subconjuntos** según la participación en volumen, y aplicando sobre cada uno de ellos las técnicas que mejores resultados habían demostrado: **LightGBM**, **regresión lineal** y el **AutoGluon simple**.

El paso final fue la combinación (ensamble) de estas predicciones, retomando el espíritu del enfoque anterior pero desde una arquitectura más liviana y precisa, alineada con el nuevo principio que había guiado los últimos descubrimientos: **la simplicidad como estrategia central de modelado**.