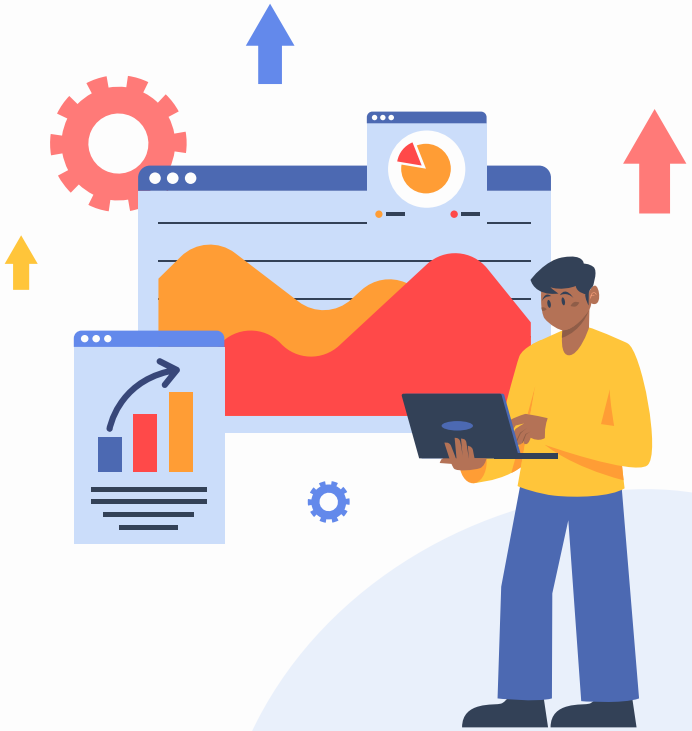
Challenge DS/DE Benefits

Paul Lijtmaer



01

EDA

Objetivo - Problema a resolver

- El objetivo era entender cómo mejorar la efectividad de las ofertas relámpago en términos de ventas, al realizar un EDA de un dataset suministrado.

- Se buscaba identificar los mejores momentos para lanzar ofertas, las categorías de productos más exitosas, y las condiciones logísticas que influyen en las ventas (como el envío gratuito) para optimizar futuras campañas de ofertas relámpago.

Información utilizada

- La fuente de datos principal fue el dataset mismo, donde se evaluó el comportamiento de cada columna, haciendo foco en fechas, horarios, categorías de productos y stock involucrado y remanente

- Luego de observar que la concentración de las ofertas se encontraban en la última semana de julio, se investigó el calendario oficial de Argentina, proporcionado por el gobierno de Buenos Aires. https://www.argentina.gob.ar/educacion/consejofederal/calendario2025

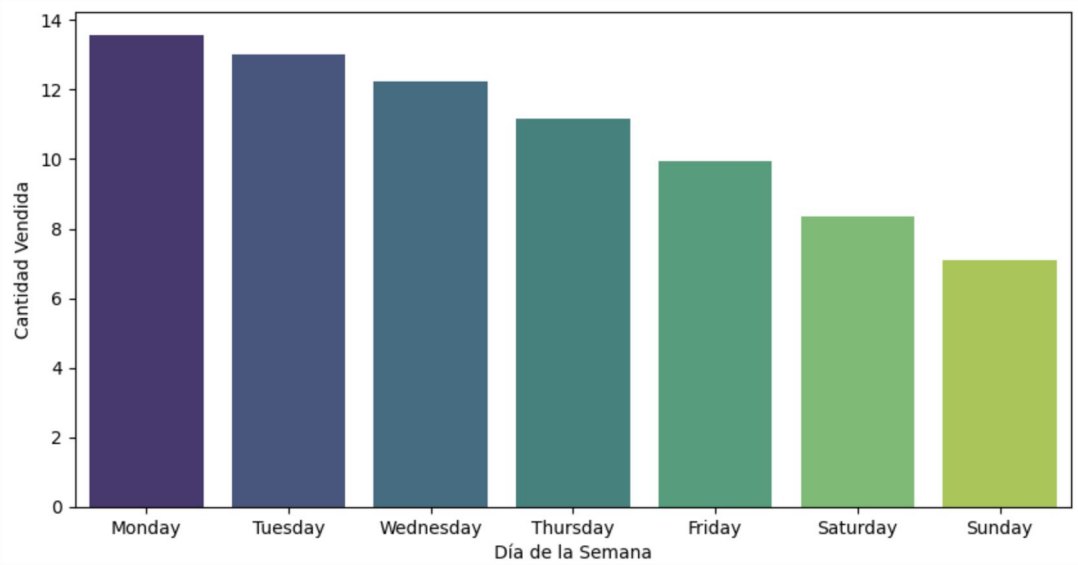
- Para entender el contexto de las ofertas relámpago, se consultó la guía de vendedores de Mercado Libre, para entender la duración de las ofertas relámpago (6 horas). https://vendedores.mercadolibre.com.ar/nota/ofertas-relampago-liquida-tu-stock-en-pocas-horas/

Hipótesis:

- Las ofertas lanzadas durante las vacaciones tendrían mejores ventas.
- Los envíos gratuitos aumentarían significativamente las ventas.
- Las categorías relacionadas con tecnología tendrían un mejor desempeño.

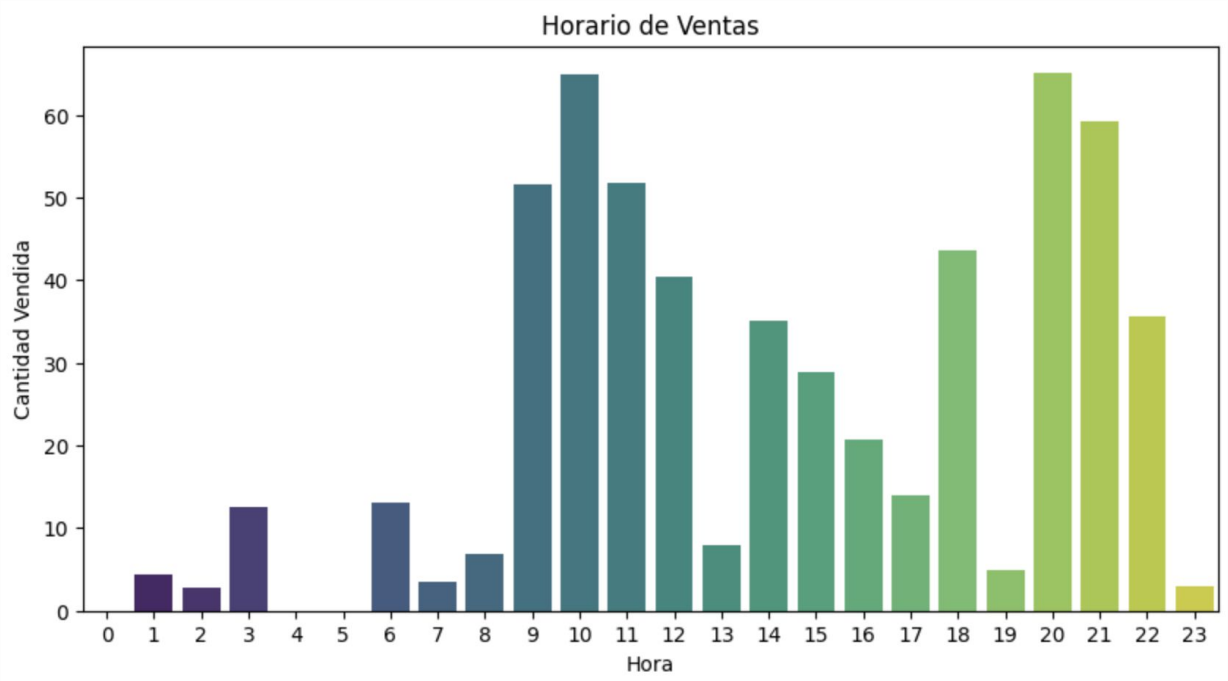
Insights

- El lunes tiene el promedio de ventas más alto (13.57), lo cual puede indicar que las ofertas que comienzan o terminan este día son particularmente atractivas. Esto podría estar asociado al timing de las ofertas relámpago, cuando los usuarios vuelven a sus rutinas y tienen más probabilidades de comprar. Los promedios disminuyen progresivamente desde el lunes hasta el domingo, mostrando una tendencia a la baja en los días del fin de semana. Es posible que los usuarios compren menos durante el fin de semana debido a actividades al aire libre o simplemente a una menor actividad online.



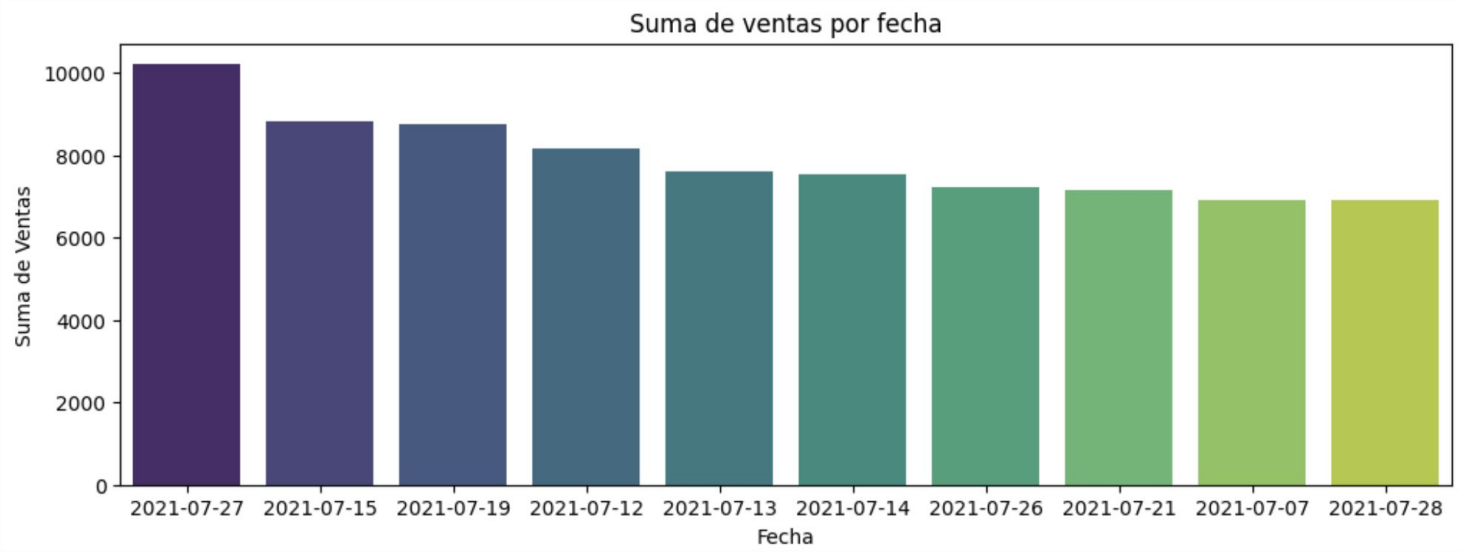
Insights

- Respecto a los horarios, los horarios más frecuentes para lanzar ofertas fueron las 13 hs, 19 hs y 7 hs, mientras que los picos de ventas fueron a las 20 y a las 10hs, con 65 y 10 unidades vendidas en promedio, mostrando el éxito de los horarios seleccionados para lanzar ofertas relámpago.



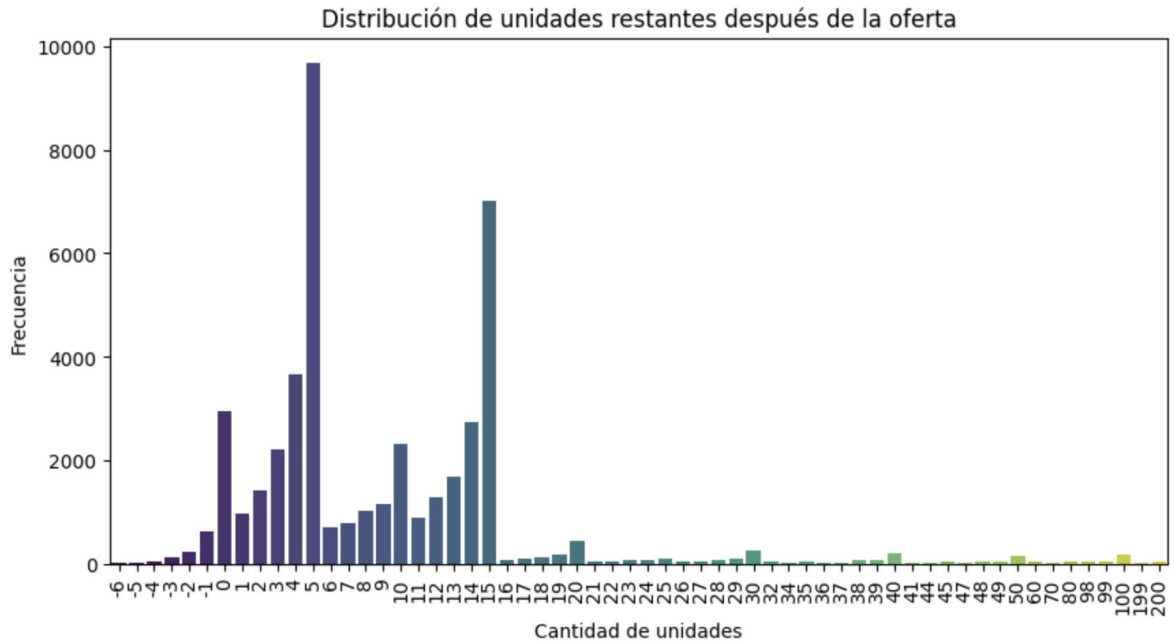
Insights

- La mayoría de las ventas se concentraron en julio, especialmente durante las vacaciones de invierno en Argentina (21 de julio a 1 de agosto). Esa semana fue la que mayor cantidad de ofertas inició, 8147, mientras que la fecha donde se vendió la mayor cantidad de unidades fue el 27 de julio, con 10195 unidades.



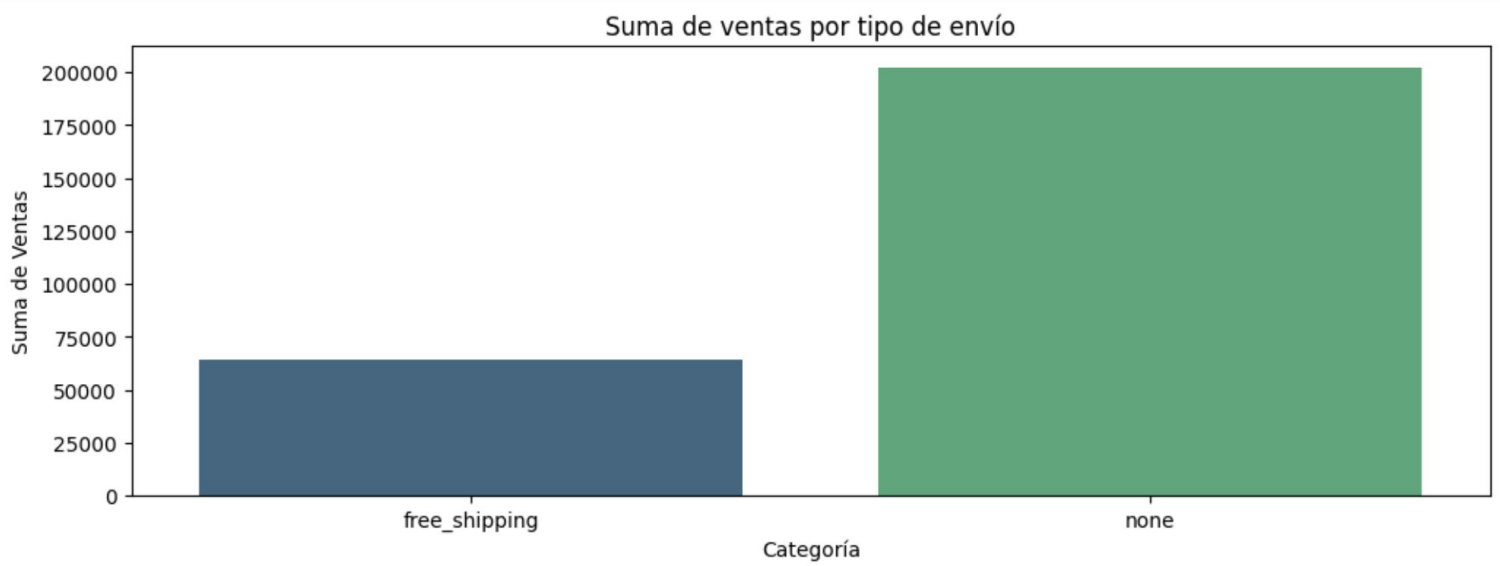
Insights

- En ocasiones, el stock remanente luego de la oferta relámpago tiene valores negativos, posiblemente indicando una sobreventa, donde se vendieron más unidades de las disponibles originalmente, o un error de lógica. Se debería revisar la lógica de "REMAINING_STOCK_AFTER_END" y "SOLD_QUANTITY", los productos vendidos.



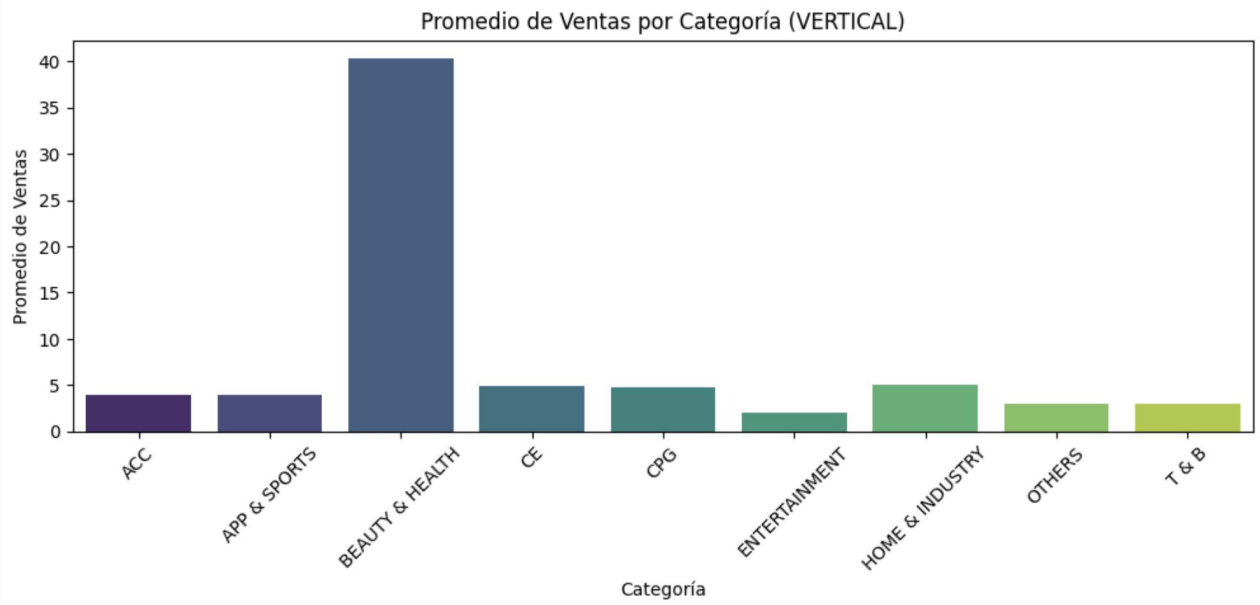
Insights

- Curiosamente, las ofertas con envío gratis no fueron más exitosas que las que no lo tenían (64278 unidades vendidas en total contra 202349).



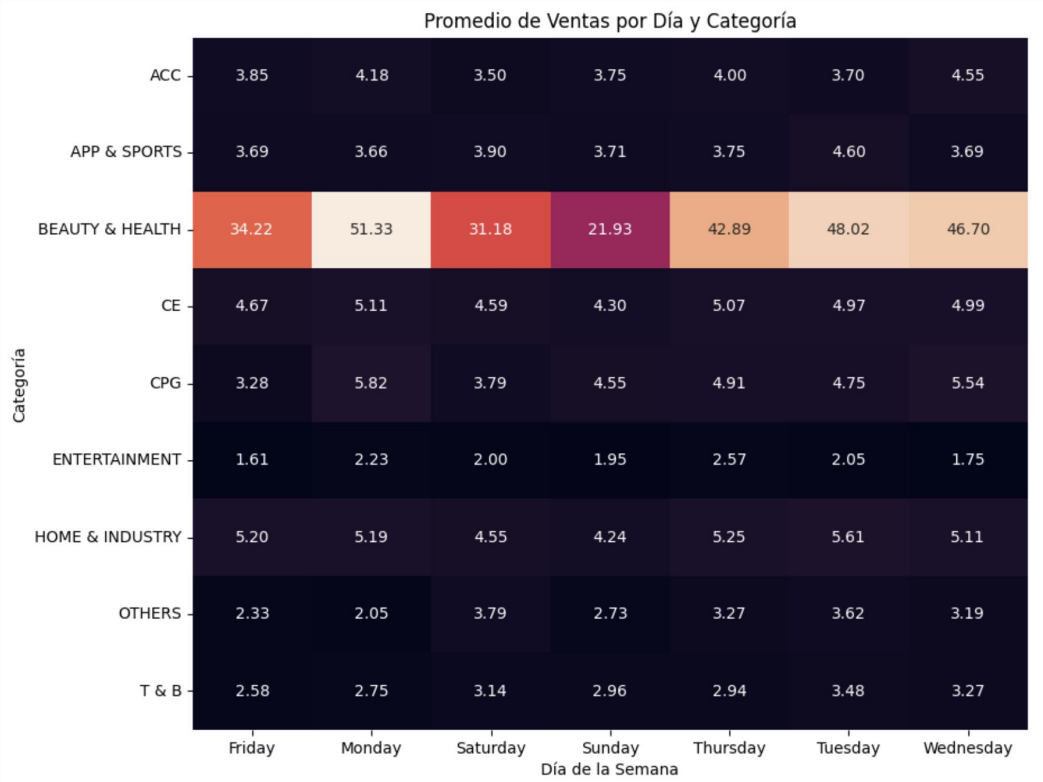
Insights

- Si bien las categorías APP & SPORTS y HOME & INDUSTRY fueron las que mayor cantidad de ofertas relámpago tenían (13065 y 10822 respectivamente), la categoría BEAUTY & HEALTH fue la que más unidades vendió en promedio (aproximadamente 40 unidades, contra 3.8 de APP & SPORTS y 5 de HOME & INDUSTRY.



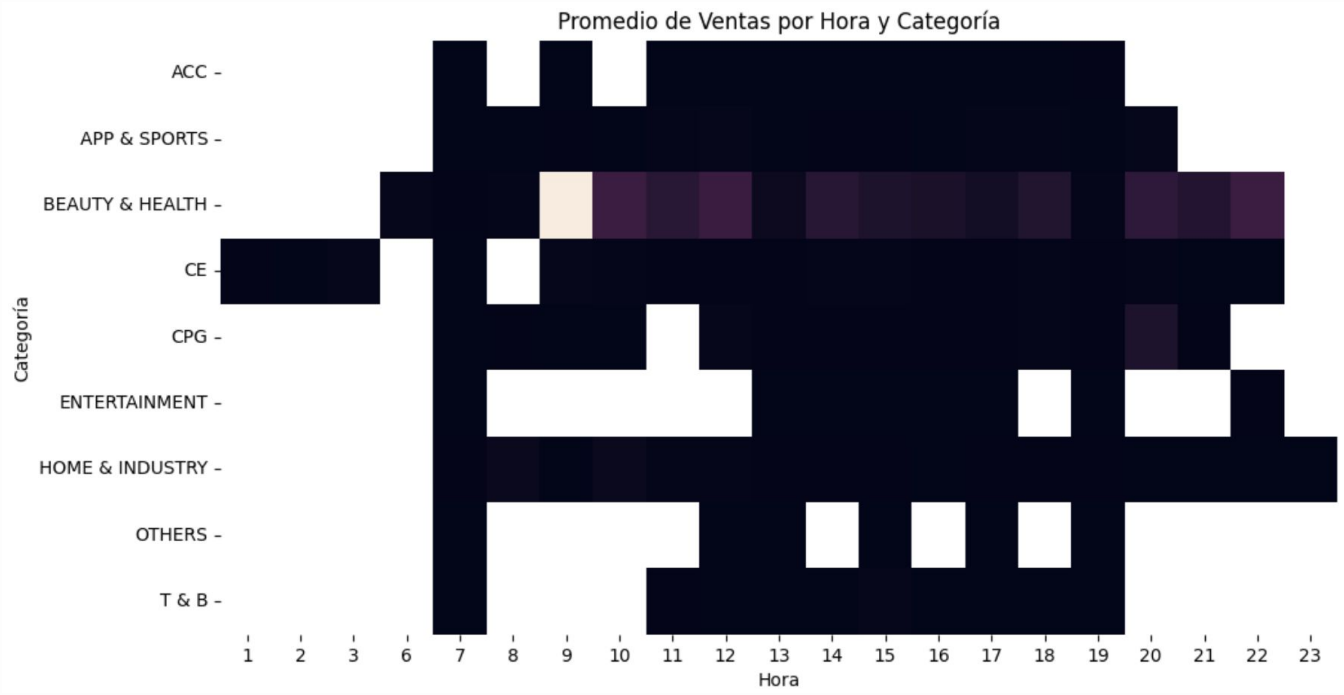
Insights

- Dentro de las ventas de la categoría BEAUTY & HEALTH, el promedio más alto se observa los lunes, seguido por los martes y miércoles (51.3, 48 y 46).



Insights

- Las 9, 10, 12 y 20hs son los horarios donde los productos de la categoría BEAUTY & HEALTH concentraron sus ventas (755, 129, 122, 102)



Contexto

La metodología aplicada en la realización del EDA fue

- Cargar librerías y dataset
- Explorar columnas, tipos de datos, duplicados
- Análisis univariado profundo
  -Se exploró cada columna por separado, particularmente sus valores y comportamiento
  -Se generaron counts por valores y se generaba un gráfico para visualizar el comportamiento y patrones propios de la columna
  -Se anotó lo observado
- Análisis multivariado
  -Exploración de correlaciones
  -Exploración de relación entre columnas
  -El objetivo era encontrar patrones y cómo una columna afectaba o segmentaba otra. Por ejemplo, ventas por categorías, ventas por días de semana/fechas/horarios, ventas por categorías por días de semana, etc

Conclusiones y futuros pasos

- El análisis confirmó que los factores temporales y la categoría de productos son críticos para el éxito de las ofertas relámpago. Se recomienda por lo tanto concentrar las ofertas en la segunda quincena de julio
- Lanzar ofertas relámpago a las 7, 13 y 19hs
- Concentrar ofertas en categorías de BEAUTY & HEALTH
- Explorar oportunidades no explotadas en junio y en HOME & INDUSTRY, la siguiente categoría con más ventas en promedio

Futuros pasos:

- Evaluar si estos patrones se repiten en otros períodos del año, ya que solo se evalúan 2 meses.
- En conjunto con el punto anterior, entrenar para predecir la demanda y optimizar el stock involucrado en las ofertas. Es importante extender la ventana de tiempo, ya que un modelo de ML entrenado solamente en estos 2 meses podría generalizar mal
- Probar distintas combinaciones de horarios y descuentos adicionales, para realizar A/B testing

02

Series de Tiempo

Objetivo - Problema a resolver

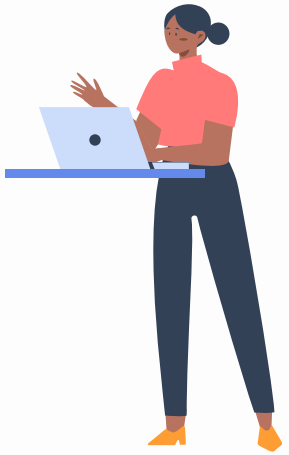
- El objetivo era predecir la cantidad de unidades vendidas diariamente para tres categorías durante los próximos 21 días utilizando series de tiempo

- Esto relevante para la planificación de inventarios y estrategias de marketing de e-commerce

Hipótesis

- Cada categoría tiene patrones temporales únicos.

- La presencia de outliers sugiere picos inusuales de demanda.



Información utilizada

- Se utilizó el dataset de series, que contaba con 3 columnas, CATEGORY, DATE y UNITS_SOLD, la variable target.

- Al tratarse de series de tiempo, la idea fue aplicar modelos ARIMA y otros de regresión, como RandomForestRegressor, XGBRegressor y LGBMRegressor. Para aplicar los modelos correctamente, se consultó la siguiente documentación

  https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html
  https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.AutoARIMA.html#pmdarima.arima.AutoARIMA (ARIMA con tuneo de hiperparámetros como GridSearch)
  https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
  https://machinelearningmastery.com/xgboost-for-regression/
  https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html

Insights

- La serie de tiempo tiene saltos en las fechas, lo que podría afectar los modelos, en caso de no resolverlo. Además, fue necesario dividir la serie en 3, una por categoría, debido a las diferencias en patrones y comportamientos entre sí, para no mezclar las observaciones.

- ARIMA no parece capturar bien los patrones complejos de los datos (RMSE de 54.09 para CATEG_1, 5.15 para CATEG_2 y 1608.32 para CATEG_3), independientemente de que para CATEG_2 haya sido un modelo intermedio.

- Los modelos basados en árboles son muy efectivos.

- Para CATEG_1, RandomForest GridSearch, con un RMSE de 27.62, fue el mejor modelo.

- Para CATEG_2, RandomForest GridSearch, con un RMSE de 4.49, fue el mejor modelo.

- Para CATEG_3, XGBoost GridSearch, con un RMSE de 1400.29, fue el mejor modelo.

Contexto

- Se utilizaron un modelo basado en series de tiempo, ARIMA, y además se utilizaron modelos basados en árboles, Random Forest, XGBoost y LightGBM.

- Luego de notar que el dataset tenía 3 categorías, fue necesario separarlo en 3 series, una por categoría.

- Se encontró que no había secuencialidad en las fechas, ya que faltaban algunas. Por lo tanto, se resolvió este tema, al imputar por 0, ya que eran fechas donde no se habían concretado ventas.

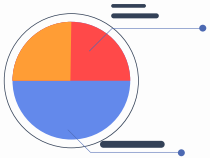
- Se dividieron los datos en conjuntos de train y test, con shuffle=False, para no mezclar el conjunto de test, y con test_size=21, así se evaluaban los últimos 21 días, ya que el objetivo era predecir 21 días.

- Para ARIMA, se evaluó si cada serie era estacionaria. Al no serlo, se aplicó diferenciación, luego se realizaron gráficos de ACF y PACF para encontrar los parámetros p y q para entrenar ARIMA (luego de tener d=1 por la diferenciación). Luego se graficaron los resultados.

- Para cada modelo de árbol, se entrenó uno en su versión base, sin tocar sus hiperparámetros. Luego se aplicó GridSearchCV para tunear hiperparámetros y finalmente se compararon los RMSE.

Solución

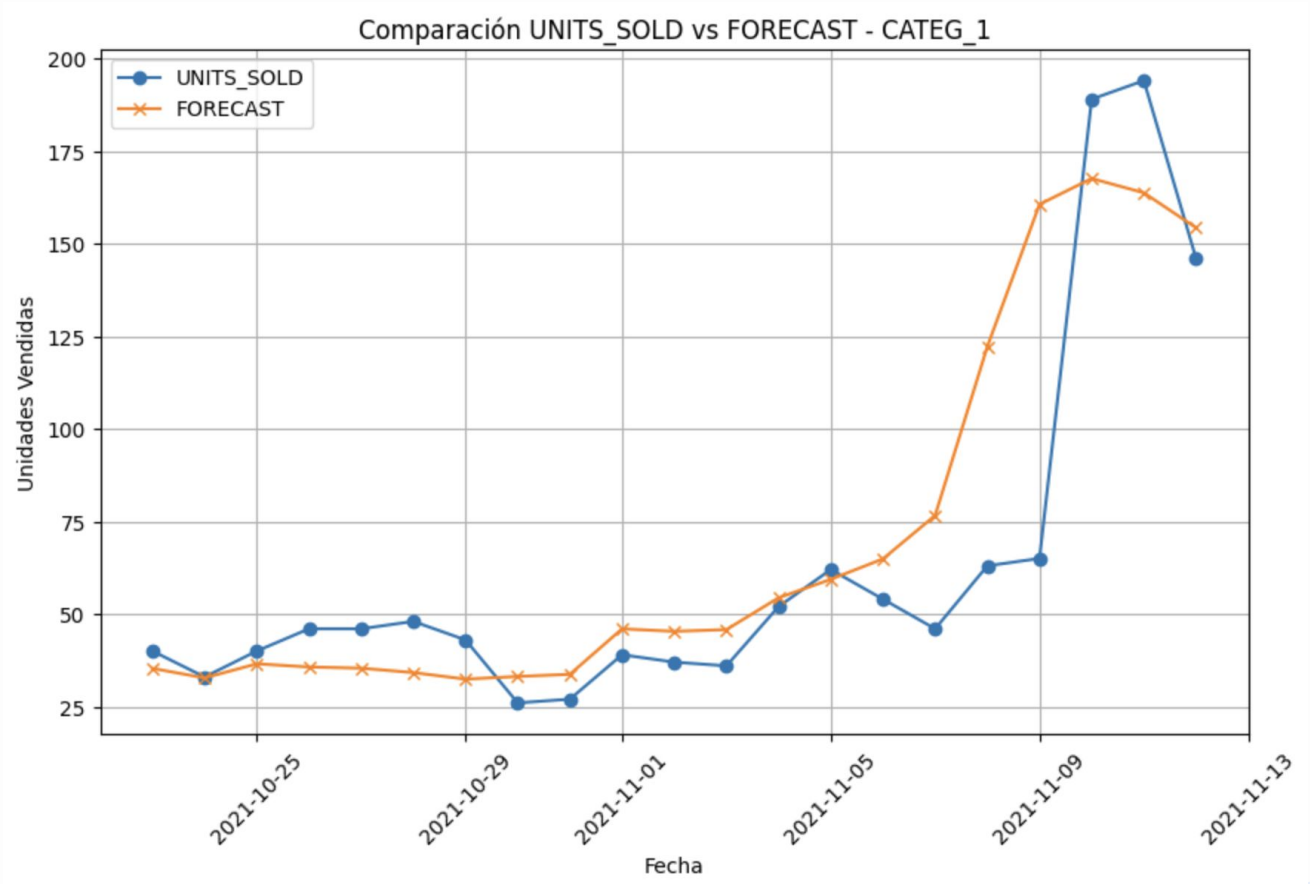
- La solución es utilizar modelos de Random Forest con Gridsearch para las primeras 2 categorías, y XGBoost con GridSearch para la tercera.
- Los modelos parecen ajustarse bien a las series temporales, permitiendo predecir las ventas diarias de forma precisa.
- La división por categorías permite pronósticos más ajustados a cada segmento.
- Permite planificar inventarios y estrategias de marketing específicas para cada categoría.



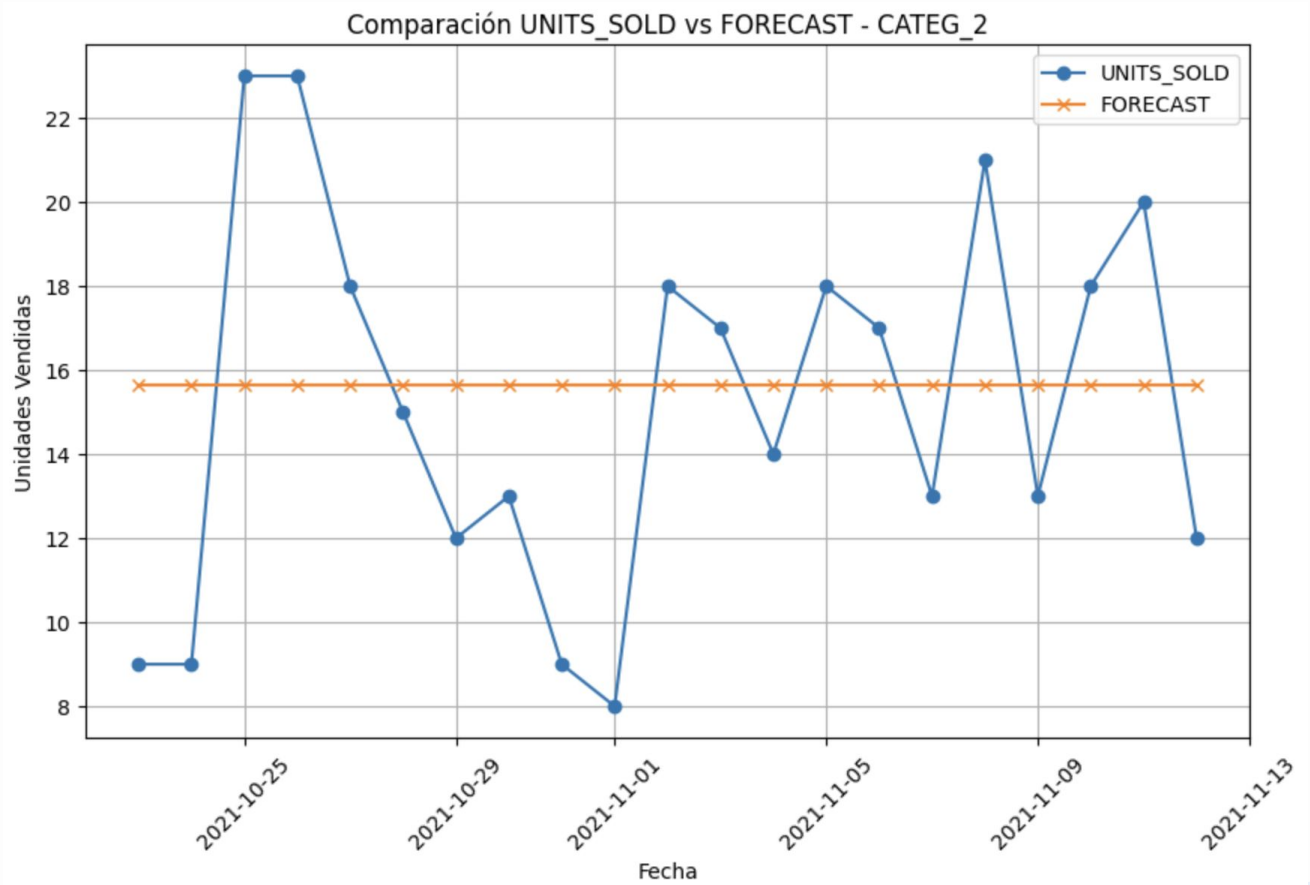
Comparativa de Modelos

	Modelo	CATEG_1 RMSE	CATEG_2 RMSE	CATEG_3 RMSE
0	ARIMA	54.090000	5.150000	1608.320000
1	RandomForest Base	28.943325	4.656527	1433.489050
2	RandomForest GridSearch	27.618986	4.489356	1417.578219
3	XGBoost Base	31.811167	4.708411	3097.147069
4	XGBoost GridSearch	32.495501	14.834316	1400.288223
5	LightGBM Base	29.827313	6.143250	1446.964082
6	LightGBM GridSearch	35.459674	14.608565	1447.880200

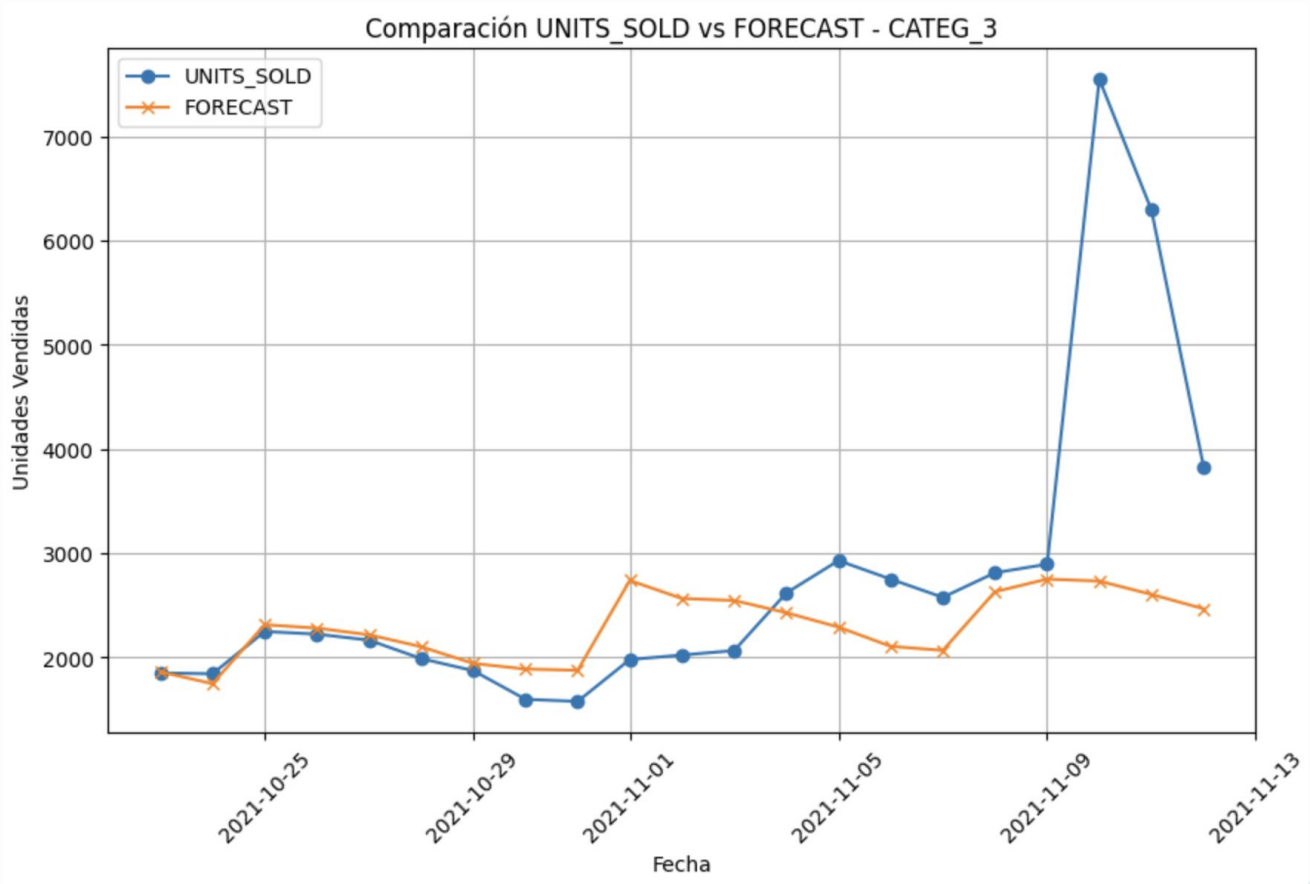
CATEG_1 vs Forecast - RF GS



CATEG_2 vs Forecast - RF GS



CATEG_3 vs Forecast - XGB GS



Conclusión y Próximos Pasos

- Los modelos aplicados son adecuados para capturar patrones temporales y manejar outliers.
- La secuencialidad de fechas fue un aspecto crítico para el éxito del forecasting.

Próximos pasos:

- Incluir variables externas (para probar con SARIMAX)
- Probar modelos con datasets históricos, para evaluar si mejora la performance de algún modelo
- Evaluar pronóstico a futuro más extenso, para evaluar períodos de reentrenamiento de los modelos seleccionados

03

SQL

Objetivo - Problema a resolver

- El objetivo del caso era analizar el rendimiento de las ventas de e-commerce mediante consultas SQL
- Identificar usuarios con alto volumen de ventas en fechas determinadas
- Determinar a los mejores vendedores por mes en la categoría de Celulares durante el 2020
- Analizar la distribución de ventas por categoría y por día para entender el rendimiento y las tendencias de compra
- Crear un sistema reprocesable para registrar el estado y precio de los ítems de forma diaria, asegurando la trazabilidad y la capacidad de análisis histórico
- En resumen, se trataba de responder preguntas clave para la toma de decisiones comerciales estratégicas y para mejorar la eficiencia del marketplace

Información utilizada

- Se utilizó la información proporcionada respecto a las entidades: Customer, Order, Item, Category

- Atributos clave utilizados:
  De Customer: id_customer, nombre, apellido, fecha_nacimiento.
  De Order: id_order, fecha_venta, costo_total, cantidad.
  De Item: id_item, precio, estado, fecha_de_baja.
  De Category: id_category, nombre_cat para identificar categorías.

- Hipótesis:

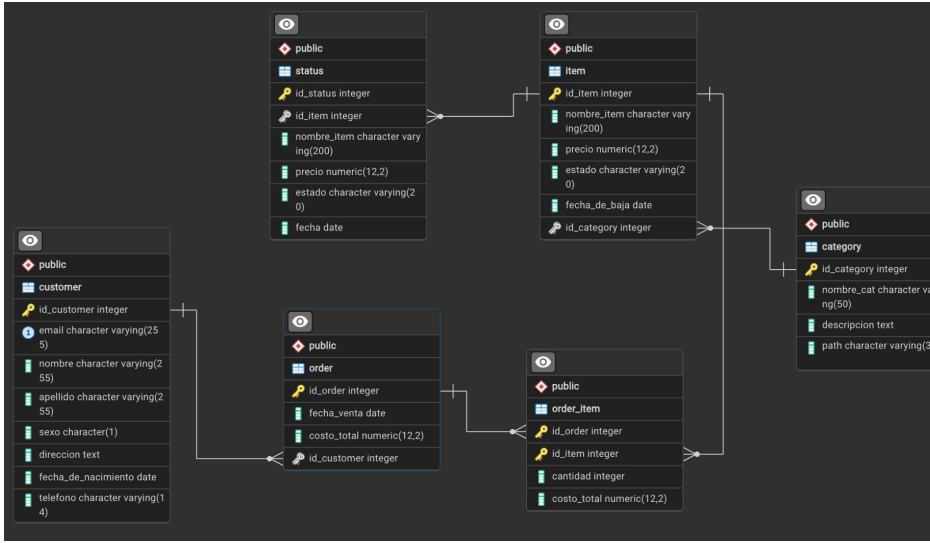
-La distribución de ventas diarias y por categoría ayudaría a entender patrones estacionales o tendencias de compra.

Insights*

- La consulta de usuarios con más de 1500 ventas en enero 2020 mostró una concentración de ventas en pocos usuarios, sugiriendo la existencia de vendedores estratégicos.
- La consulta de los top 5 vendedores de Celulares en 2020 reveló que ciertos meses tenían picos claros en las ventas, posiblemente ligados a promociones
- La consulta de ventas por categoría reveló que "Celulares" dominó el % de ventas diarias.
- La tabla Status permitió mantener un histórico claro del estado y precio de los ítems, útil para auditorías y análisis histórico.

*Al no trabajar sobre un dataset existente, se infieren los resultados y causas de las consultas. El dataset utilizado para EDA, contiene una ventana de 2 meses del 2021, no compatible con este ejercicio, concentrado en el 2020.

Contexto y metodologías



- En primer lugar, se diseñó un DER, junto con el DDL para la creación de las tablas.

- Un cliente (Customer) puede tener muchos pedidos (Order), pero cada pedido pertenece a un solo cliente (1 a N)

- Una categoría (Category) puede tener muchos productos (Item), pero cada producto pertenece a una sola categoría (1 a N)

- Se creó una tabla intermedia, Order_Item, para resolver la relación N a M entre Order (pedidos) y Item (productos), ya que una compra puede tener muchos productos, y un tipo de producto puede estar en muchos pedidos, independientemente de que el caso haya sido simplificado al no contar con un flujo de carrito de compras

- Para la creación del script DDL, se tuvo de referencia las descripciones proporcionadas en la consigna.

- Se trató de tener la menor cantidad de columnas posibles para tener una estructura normalizada, con el fin de evitar redundancias y garantizar integridad referencial

Contexto y metodologías

Para cada consulta, se buscó responder la pregunta al dividirla en partes

- Para obtener los usuarios que cumplan años el día de hoy cuya cantidad de ventas realizadas en enero 2020 sea superior a 1500:
  -Se realizó un join entre Customer y Order
  -Se filtró por fecha de venta entre 1 y 31 de enero del 2020, junto con usuarios que cumplan años el día de hoy (CURRENT_DATE)
  -Se agrupó por id, nombre y apellido de usuarios
  -Se aplicó un filtro sobre una agregación, en este COUNT de pedidos, mayores a 1500

- Para obtener el top 5 de usuarios que más vendieron($) en la categoría Celulares, por cada mes del 2020:
  -Se creó una CTE, para luego utilizarla desde el FROM de la consulta
  -Se joinearon todas las tablas relevantes (Order, Order_Item, Category y Customer)
  -Se filtró por categoría de 'Celulares' y el año
  -Se agrupó por mes, año, nombre y apellido, para aplicarle agregaciones a los datos
  -Se calculó la cantidad de ventas, cantidad de productos vendidos y monto total
  -Se creó otra CTE que llamaba a la anterior, para crear un RANK
  -Se llamó a la última CTE para filtrar por el RANK
  -Se filtró por la columna de rank, los registros menores o iguales a 5

Contexto y metodologías

- Para obtener el % de venta ($) que representa cada categoría respecto del total vendido ($) por día, junto con venta mínima y máxima:
  -Se creó una CTE para encontrar el monto total vendido, para utilizar posteriormente en la columna de %, al joinear Order y Order_Item, para agrupar por fecha y sumar el monto
  -Se creó otra CTE que calculaba el monto total, luego de joinear Order, Order_Item y Category, para luego agrupar por categoría por día
  -Se creó otra CTE para extraer ventas mínimas y máximas por fechas
  -Se joinearon las CTEs y se calculó la columna de %

- Para poblar una nueva tabla con el precio y estado de los Ítems a fin del día, que se ejecute diariamente y sea reprocesable:
  -Primero se creó una tabla Status
  -Luego se armó el INSERT desde Item
  -Se agregó la lógica para reprocesar, al agregar un filtro en la fecha de Item, que permita seleccionar la fecha y cuánto restarle, a través de las variables ${BUFFER_SIZE} y ${EXECUTION_DATE}, implementadas fuera de la query, en algún script, para posteriormente llamar en el orquestador en Matillion o Airflow
  -Se agregó una opción adicional, comentada, de DATEADD, en caso que INTERVAL no sea compatible con la DBMS, y se agregó un cron tentativo

Conclusión y futuros pasos

- La solución proporciona un marco sólido para gestionar las transacciones y analizar el comportamiento de ventas por categorías, fechas y productos

- La estructura relacional garantiza la integridad y facilita el mantenimiento a largo plazo

Futuros pasos:

- Analizar los EXPLAIN plans para agregar índices en columnas clave (fecha_venta, id_category, etc.)

- Implementar Matillion o Airflow para automatizar las inserciones en la tabla Status. Usar DAGs para programar reprocesamientos diarios

- Correr consultas con datos reales, para poder evaluar si los resultados son relevantes para optimizar campañas de marketing, reportes, etc.

- Ampliar modelo al agregar tablas y columnas adicionales que aporten datos adicionales para mejorar el nivel de análisis

- Crear dashboards en herramientas como Tableau o Power BI para visualizaciones en tiempo real de ventas y stock.

- Crear modelos de Machine Learning para predecir ventas

Muchas gracias!