Entrega Final Proyecto: Predicción inteligente ventas de productos del sector de decoración

Camila Patricia Malagón Suarez
Luis David Gutiérrez Loaiza
David Alejandro Rojas Castro
David Zapata Vásquez
Grupo 5

Resumen del problema

Contexto

La empresa XYZ, líder en el sector de soluciones de decoración en Colombia, maneja un amplio portafolio de productos que incluye soluciones para clientes particulares hasta productos para otros negocios como contratistas o grandes constructoras. A pesar de su liderazgo, la empresa enfrenta desafíos en la optimización de su portafolio de productos. Algunos productos presentan bajos retornos financieros, lenta rotación o altos costos de producción, almacenamiento o transporte, lo cual impacta negativamente en su rentabilidad. En este mercado tan competitivo, la empresa necesita información detallada y predictiva para tomar decisiones estratégicas fundamentadas en datos.

Pregunta de negocio y alcance del proyecto

El proyecto tiene como objetivo responder a la pregunta de negocio: **¿Cómo puede la empresa XYZ identificar patrones y tendencias en las ventas de sus productos para facilitar la toma de decisiones estratégicas en su portafolio?** Para responder la pregunta, se desarrollará un tablero predictivo que permita a los usuarios visualizar proyecciones de ventas. Este tablero será una herramienta de apoyo para la empresa, ayudándola a optimizar su producción, acciones de marketing y publicidad en función al desempeño esperado de sus productos.

Descripción de los datos

El conjunto de datos utilizado en el proyecto abarca registros de ventas de la empresa para todo el año 2022, 2023, y primer cuatrimestre del 2024. Estos datos incluyen variables clave como la Unidad de Negocio (UEN), región geográfica (Regional), canal de venta, descripción de producto, código de producto, número de clientes que piden el producto, número de facturas emitidas para compras del producto, volumen de ventas en galones, valor de las ventas, utilidad bruta, costos y margen de cada producto. Esta información proporciona un nivel de detalle granular, esencial para realizar un análisis preciso y elaborar proyecciones.

Con respecto a la segunda entrega, se realizó un ajuste al esquema del tablero:

- Para la sección de análisis predictivo se tenía contemplado hasta la segunda entrega un único input por Canal Comercial. Para la implementación final del tablero, se expandió y se crearon los nuevos inputs: Unidad de negocio (UEN), regional, marquilla, código de producto, producto, y meses a proyectar. Estos inputs corresponden a los datos que espera recibir el API para la predicción de la venta.

Modelos desarrollados y evaluación

Para abordar el problema de negocio de predecir las ventas futuras de los principales productos de la empresa XYZ, se entrenaron varios modelos de regresión que fueron evaluados para determinar su precisión y utilidad en la toma de decisiones. Se utilizó un enfoque de arquitectura de datos en niveles (Bronze, Silver y Gold) para preparar los datos antes de ser usados en los modelos.

- **Bronze:** en esta carpeta se cargaron los datos crudos.
- **Silver:** A partir del archivo en Bronze, se aplicaron transformaciones iniciales para limpiar los datos y eliminar valores inconsistentes, también se eliminaron los productos con bajas ventas y de registros que contenían valores negativos.
- **Gold:** Finalmente, se seleccionaron los productos que representan el 80% de las ventas acumuladas y se aplicó una codificación para variables categóricas, asegurando que los datos estuvieran listos para el entrenamiento del modelo predictivo.

Se desarrollaron y evaluaron varios modelos utilizando las variables categóricas (*Unidad de negocio*, *Canal Comercial*, *Marquilla* y *Región*) y otras variables predictoras como volumen de ventas y margen, excluyendo *Costos* y *Utilidad Bruta* debido a su alta correlación con la variable objetivo.

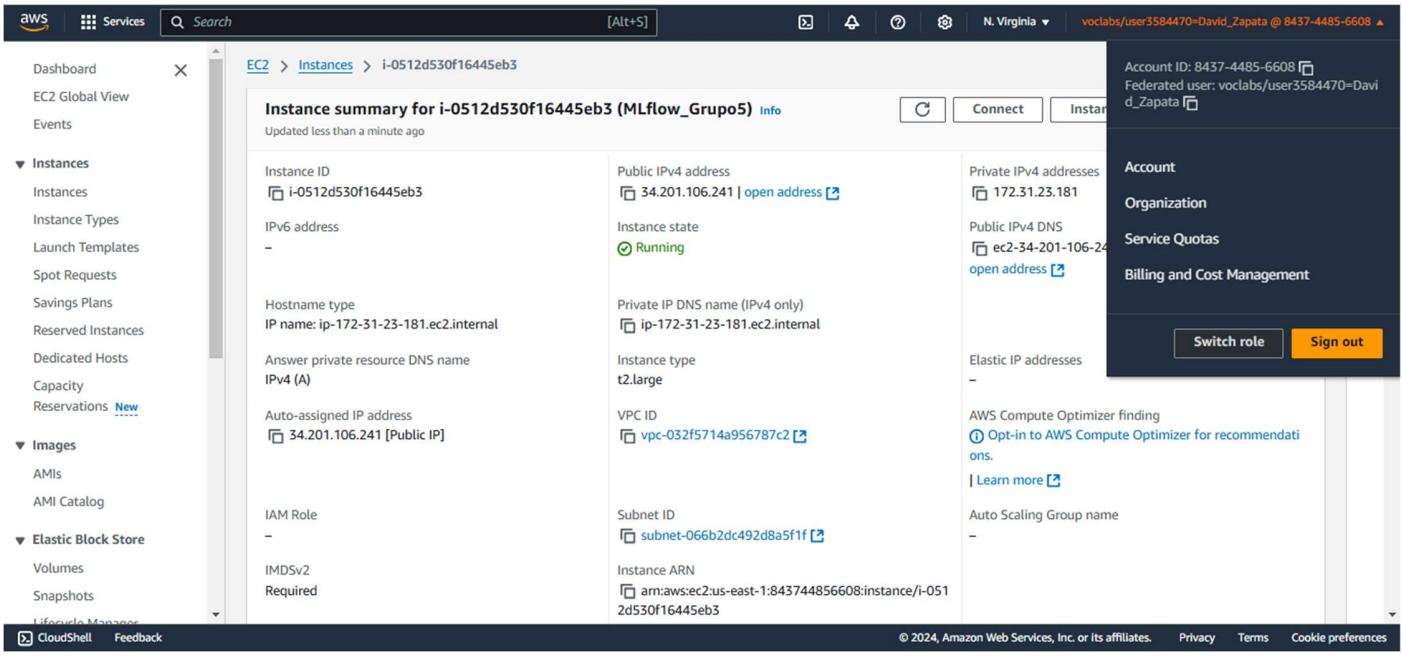
La selección de modelos en este análisis se fundamenta en la naturaleza del problema de predicción de ventas y en la diversidad de enfoques que los modelos permiten explorar:

1. **Random Forest Regressor:** Este modelo de ensamble es robusto ante el sobreajuste, especialmente en problemas de predicción con múltiples variables. Al promediar múltiples árboles de decisión, Random Forest captura relaciones no lineares en los datos sin requerir demasiada configuración de hiperparámetros. Su optimización mediante GridSearchCV permite ajustar sus hiperparámetros para mejorar el rendimiento.
2. **XGBoost:** Es uno de los modelos más poderosos para problemas de regresión debido a su capacidad para optimizar el rendimiento a través del boosting. XGBoost mejora el modelo al corregir iterativamente los errores de los modelos anteriores, lo cual es oportuno para este caso a partir de datos complejos. El uso de GridSearchCV asegura que se encuentren parámetros óptimos.
3. **Gradient Boosting Regressor** y **Linear Regression**: Estos modelos sirven como puntos de comparación. Gradient Boosting ofrece una alternativa al XGBoost en el marco de técnicas de boosting, mientras la regresión lineal proporciona un modelo base simple. Comparar los resultados obtenidos con dos enfoques permite evaluar si el modelo de complejidad en otros modelos genera una mejora significativa en su rendimiento.

Las métricas de evaluación utilizadas fueron el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el coeficiente de determinación (R^2).

Para gestionar de manera eficiente los experimentos de modelado y comparar diferentes enfoques, se creó una instancia en **AWS EC2** que actúa como servidor para correr los experimentos en MLflow. Esta instancia permite ejecutar los experimentos de forma centralizada y almacenar los resultados, lo que facilita la comparación y el seguimiento de cada uno de los modelos desarrollados. MLflow se utilizó para registrar las métricas claves de cada modelo, así como para guardar los hiperparámetros y resultados obtenidos, esto es clave para seleccionar el mejor modelo en términos de rendimiento.

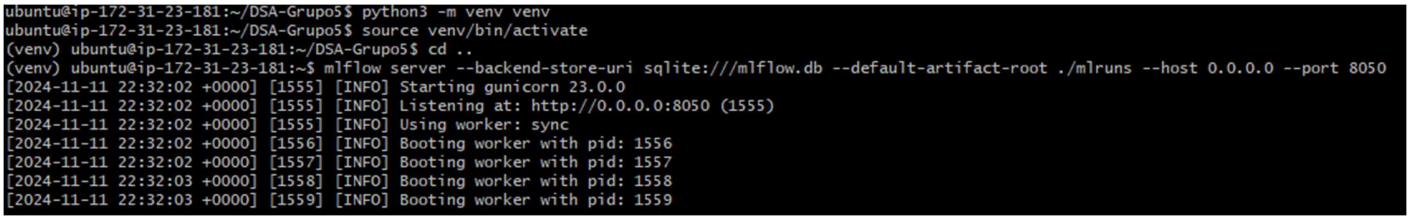
A continuación, se presenta pantallazos de los experimentos realizados en MLflow en la máquina virtual en AWS EC2.



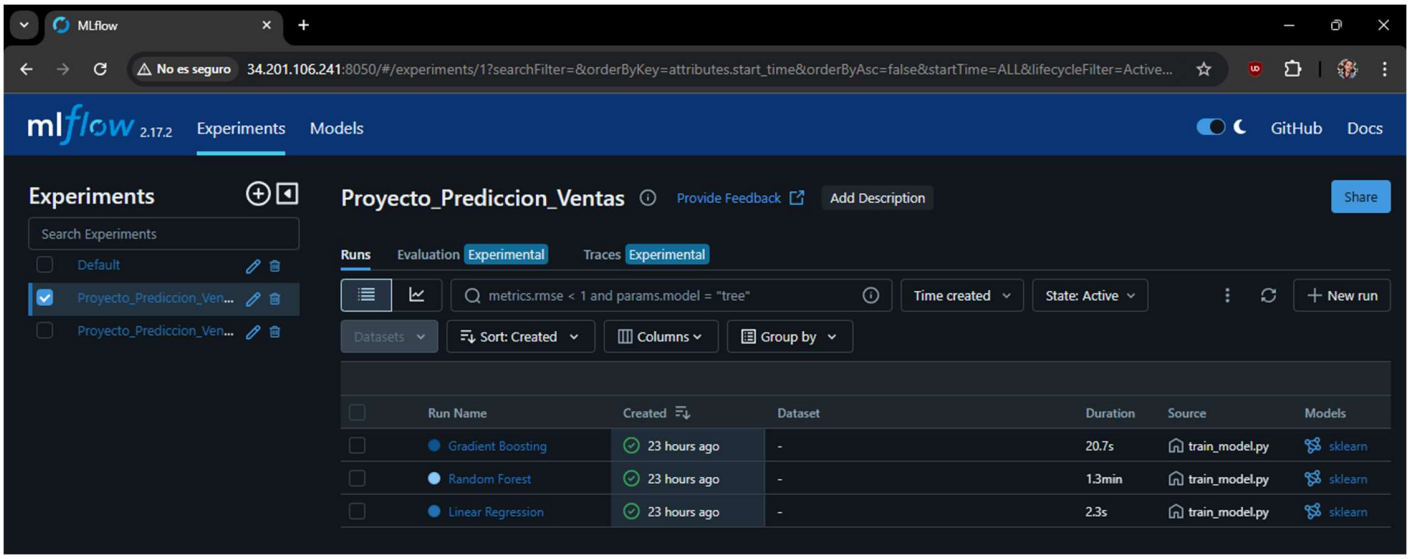
En la imagen anterior se muestra la instancia de AWS EC2 configurada para ser servidor de los experimentos. Con esta instancia se puede centralizar y gestionar de manera eficiente la ejecución de los modelos, permitiendo un entorno seguro y accesible desde cualquier lugar.

```
david@DavidZ MINGW64 /
$ ssh -i '/c/Users/david/OneDrive/Maestría/MasterTrack 3/Despliegue de soluciones analiticas/Proyecto/MLflow/Proy_Grupo5.pem' ubuntu@34.201.106.241
The authenticity of host '34.201.106.241 (34.201.106.241)' can't be established.
ED25519 key fingerprint is SHA256:8qVHe773aUMLHb+/16Z98epHfXgtliWkXLjoJp+C8kk.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:18: 98.80.74.166
    ~/.ssh/known_hosts:21: 34.235.166.100
    ~/.ssh/known_hosts:22: 34.234.78.58
    ~/.ssh/known_hosts:23: 54.226.25.248
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.201.106.241' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Nov 11 22:28:29 UTC 2024

  System load:  0.0               Processes:             112
  Usage of /:   28.1% of 28.02GB  Users logged in:       0
  Memory usage: 2%                IPv4 address for enX0: 172.31.23.181
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

20 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Mon Nov 11 15:36:36 2024 from 190.70.80.138
```

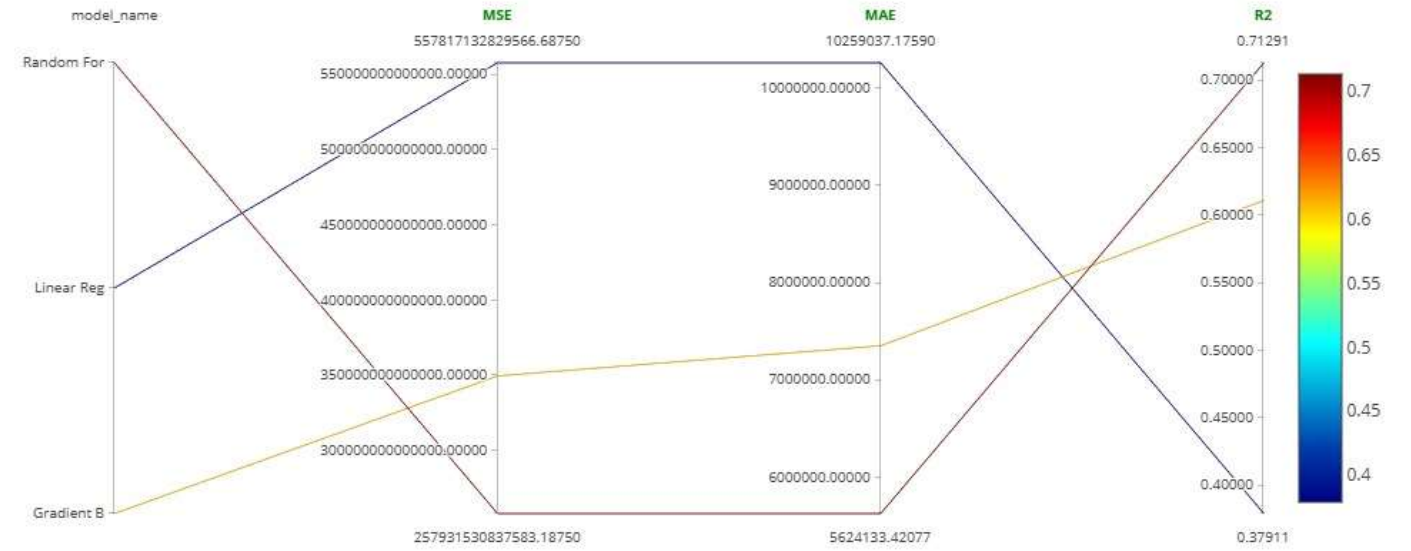
La captura anterior evidencia la conexión con la máquina virtual en AWS.



En esta imagen se muestra la configuración del servidor en AWS EC2 para correr los experimentos.



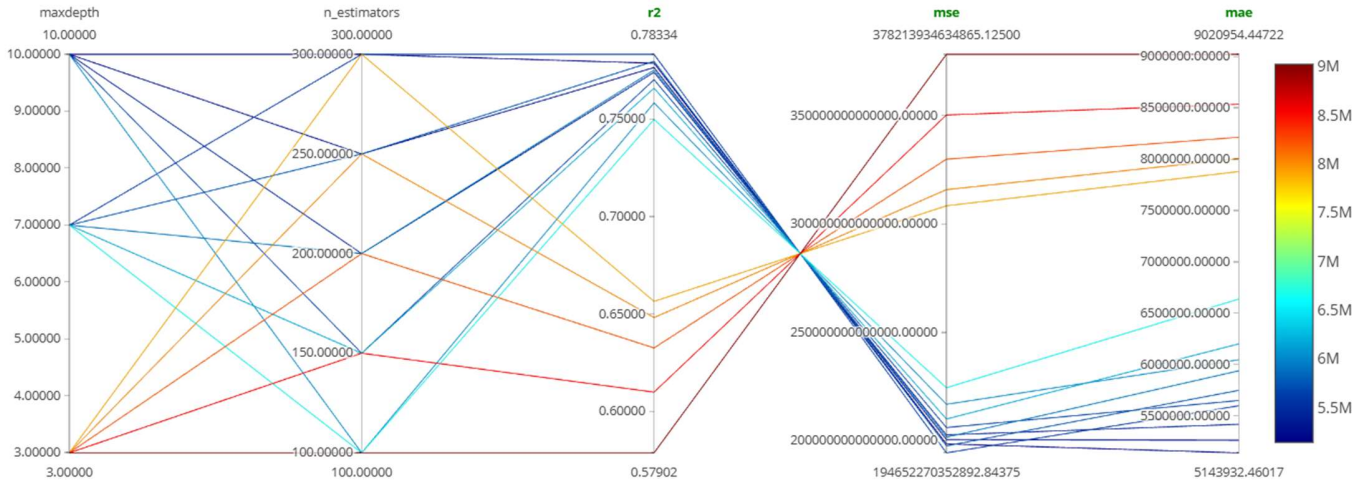
Por último, con el servidor de MLflow corriendo se puede acceder a su interfaz, en esta imagen se puede apreciar algunos de los experimentos ejecutados. Esta herramienta permitió registrar las métricas relevantes como MSE, MAE, R^2 y también los hiperparámetros, para cada uno de los modelos. Esto facilita la comparación objetiva de los resultados y la selección del mejor modelo, por ejemplo, con visualizaciones de los modelos ejecutados y sus resultados:



Comparación en MLFlow de los primeros modelos testeados usando las variables predictoras originales

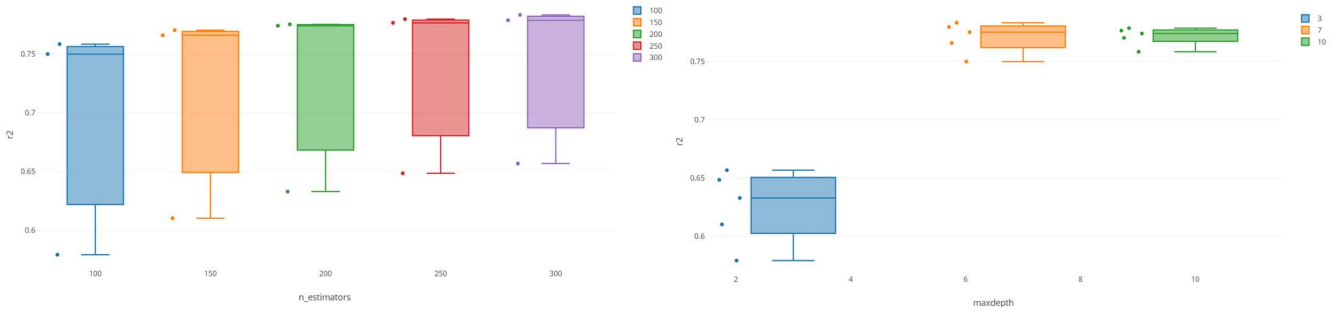
Después, se ajustaron las variables predictoras utilizadas para probar nuevamente los modelos y verificar si existía alguna mejora en el desempeño de estos. Para esto, se evaluaron varios modelos utilizando las

variables *Año, mes, Unidad de negocio, Canal Comercial, Marquilla, Región, Código y Nombre de producto*. Realizando el cambo de variables predictoras y ajustando diferentes modelos como Random Forest Regressor y XGBoost, se encontró que el mejor desempeño se obtenía con XGBoost. Debido a lo anterior, se probó en MLFlow diferentes escenarios variando los números de estimadores y la profundidad, con lo cual se obtuvo los siguientes resultados:



Comparación en MLFlow de los modelos de XGBoost variando número de estimadores y profundidad

Adicional, se exploró mediante boxplot el comportamiento del R^2 conforme varían el número de estimadores y la profundidad, encontrando que a partir de 250 estimadores se reduce la velocidad de mejoría de la métrica, y que la profundidad usada en el modelo deberá ser mínimo de 7 para alcanzar un R^2 superior a 70%.



Boxplots de comportamiento R^2 *conforme varia número de estimadores y profundidad en el modelo XGBoost*

En la siguiente tabla se encuentra un resumen de los principales modelos testeados y los resultados alcanzados en las métricas de error cuadrático medio (MSE), error absoluto medio (MAE) y R^2:

#	Modelo	Hiperparámetros	MSE	MAE	R2
1	Random Forest Regressor	Default	291686836405496	6.036.291	0.6753
2	Random Forest Regressor – Ajuste variables predictoras	GridSearchCV(CV=3)	205955135679480	4.767.559	0.7707
3	XGBoost	GridSearchCV(CV=3)	270529027885732	5.990.930	0.6988
4	XGBoost	GridSearchCV(CV=5)	269206019057069	5.949.237	0.7003
5	XGBoost	GridSearchCV(CV=10)	268599776753441	5.920.606	0.7010
6	XGBoost – Ajuste variables predictoras	GridSearchCV(CV=3)	191795317379772	5.393.625	0.7865
7	Gradient Boosting	Default	257931530837583	5.624.133	0.713
8	Linear Regression	Default	557817132829566	10.259.037	0.379

Observaciones y conclusiones de los modelos

- Los resultados muestran que incluir la información de identificación del producto proporciona una mejor predicción de las ventas en valor, comparado con usar solo el margen como predictor. Esto se observa en las diferencias de MSE y MAE entre ambas aproximaciones.
- La evaluación de modelos reveló que **XGBoost** con ajustes de hiperparámetros a través de **GridSearchCV** resultó ser el modelo más efectivo, logrando los mejores valores de R (hasta 0.7865)
- La variable Utilidad Bruta y los Costos se excluyeron por su correlación con la variable objetivo, lo que contribuyó a simplificar el modelo sin comprometer su precisión.
- La creación de una instancia en **AWS EC2** para que sirva como servidor permitió correr experimentos en MLFlow y gestionar de manera eficiente los diferentes modelos desarrollados, facilitando así el seguimiento y comparación de resultados.

Estos resultados indican que la estructura de datos preparada con la arquitectura de niveles y la selección cuidadosa de variables predictoras y de hiperparámetros ayudó a optimizar el rendimiento de los modelos. La implementación en el tablero permitirá a la empresa XYZ a realizar predicciones y tomar decisiones estratégicas en su portafolio.

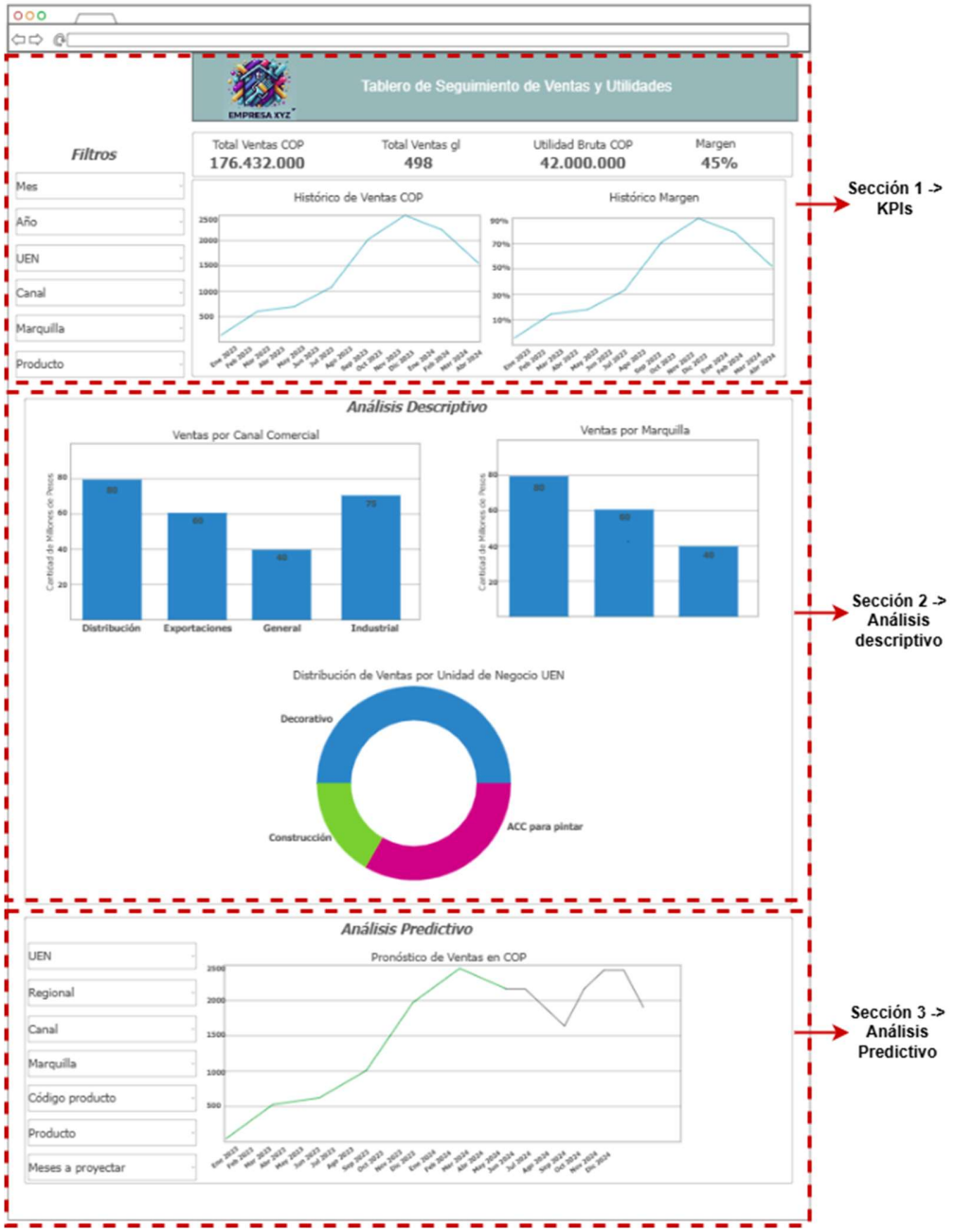
Descripción del tablero desarrollado y funcionalidad

De acuerdo con la pregunta de negocio el objetivo principal del proyecto es facilitar la toma de decisiones estratégicas, para esto el diseño del tablero contempla la presentación visual de dos tipos de técnicas analíticas:

- **Analítica descriptiva:** Presenta un análisis histórico de las ventas de los productos, según los Canales comerciales, Marquilla y unidades de negocio**.** Estas primeras secciones dentro del tablero le permitirán a la compañía realizar un análisis de cómo se han comportado las ventas en periodos de tiempo anteriores.

- **Analítica predictiva:** Esta sección del tablero permitirá identificar y predecir las ventas futuras de los productos para los canales, para que la organización se apoye en estos datos para decidir sobre el portafolio existente.

A continuación, se presenta el mockup del tablero propuesto con cada sección a implementar, más adelante se detallará cada una.



- **Sección 1 – Filtros, KPIs, Históricos:**

En esta sección se pretende dar una visión general rápida y dinámica del estado actual de las ventas en el portafolio. Aquí los usuarios pueden aplicar filtros específicos (como Unidad de Negocio Canal Comercial, Marquilla, Region) para analizar diferentes segmentos de ventas a lo largo del tiempo. Además, se presentan KPI's que resumen el rendimiento de la empresa: Ventas en volumen y valor, utilidad bruta, margen. Con la información disponible esta sección se puede complementar para que despliegue información de número de facturas emitidas y clientes que solicitaron los productos seleccionados, información que puede ser útil para entender la estacionalidad de pedidos a lo largo del año.

Fuente de datos: Los datos utilizados para la elaboración de estos gráficos provienen de la capa Gold de la arquitectura Medallion propuesta. Esta fuente contiene datos ya procesados y preparados para su

consumo en herramientas de análisis y generación de reportes, asegurando su confiabilidad y relevancia para los fines analíticos de la visualización.

Implementación Técnica: Esta sección ha sido implementada dividiendo el espacio en 3 subsecciones, zona de filtros, zona de KPIs y zona de gráficos de evolución histórica. Cada zona tiene una función independiente, partiendo de la generación de los filtros con dropdown lists, en divs independientes, pasando a la generación de los KPIs donde se incluye el cálculo de cada métrica y finalmente, dos funciones para las dos gráficas históricas.

Sección 1 - Filtros, KPIs y gráficos históricos del tablero desplegado de forma local

- **Sección 2 - Análisis descriptivo:**

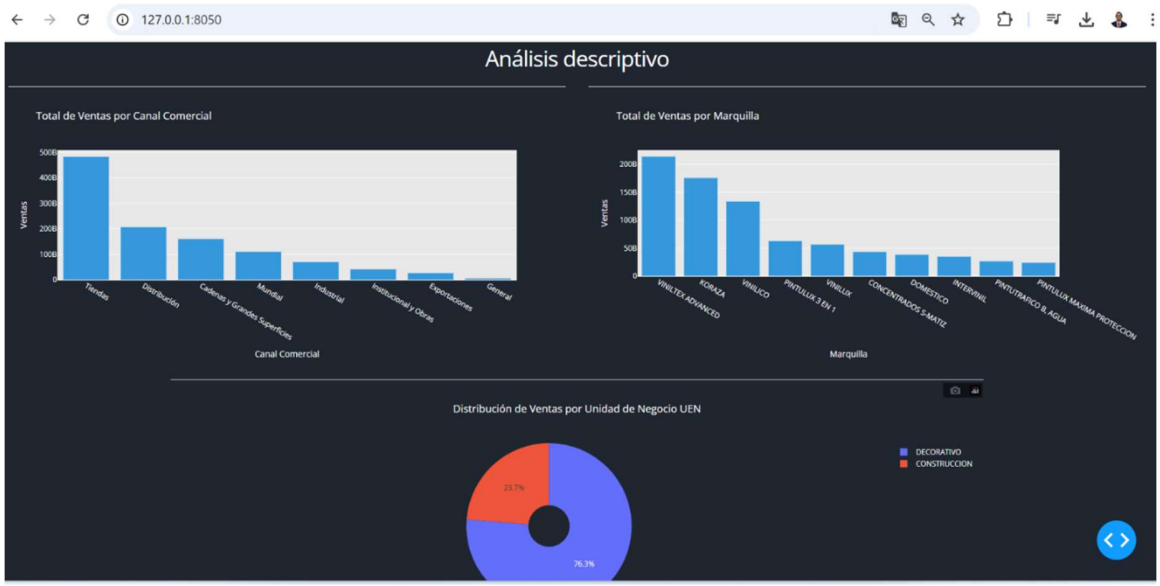
Esta sección está diseñada para visualizar el comportamiento histórico de las ventas a través de tres gráficos, permitiendo un análisis detallado de las ventas por Canal Comercial, Marquilla, y la distribución entre las distintas Unidades de Negocio (UEN).

Fuente de Datos: Los datos utilizados para la elaboración de estos gráficos provienen de la capa Gold de la arquitectura Medallion propuesta. Esta fuente contiene datos ya procesados y preparados para su consumo en herramientas de análisis y generación de reportes, asegurando su confiabilidad y relevancia para los fines analíticos de la visualización.

Implementación Técnica: La sección de análisis descriptivo ha sido implementada en Dash mediante una estructura modular. Cada gráfico corresponde a una función independiente que recibe los datos cargados, realiza los agrupamientos necesarios y configura el gráfico según el análisis requerido (ya sea a nivel de Canal, Marquilla, o UEN). Estas funciones también definen los parámetros visuales de cada gráfico, como los colores, títulos, y etiquetas de los ejes, proporcionando así una presentación coherente y visualmente atractiva de los datos.

Dentro del diseño del tablero, esta sección está organizada en una disposición de dos gráficos en la primera fila y un gráfico adicional en la segunda fila, centrado horizontalmente. Cada uno de estos gráficos es un componente independiente dentro del layout de Dash, y están vinculados al callback principal de la aplicación. En la función ***`update_output_div`***, se invocan las funciones de cada gráfico y se devuelve su

resultado para su visualización en el tablero, permitiendo una actualización dinámica y precisa de los datos mostrados en la interfaz.



Sección 2 - Análisis descriptivo del tablero desplegado de forma local

- **Sección 3 - Análisis predictivo:**

Esta sección está orientada a la visualización de predicciones de ventas en pesos, proporcionando una proyección de las ventas esperadas como insumo clave para la toma de decisiones estratégicas en la compañía.

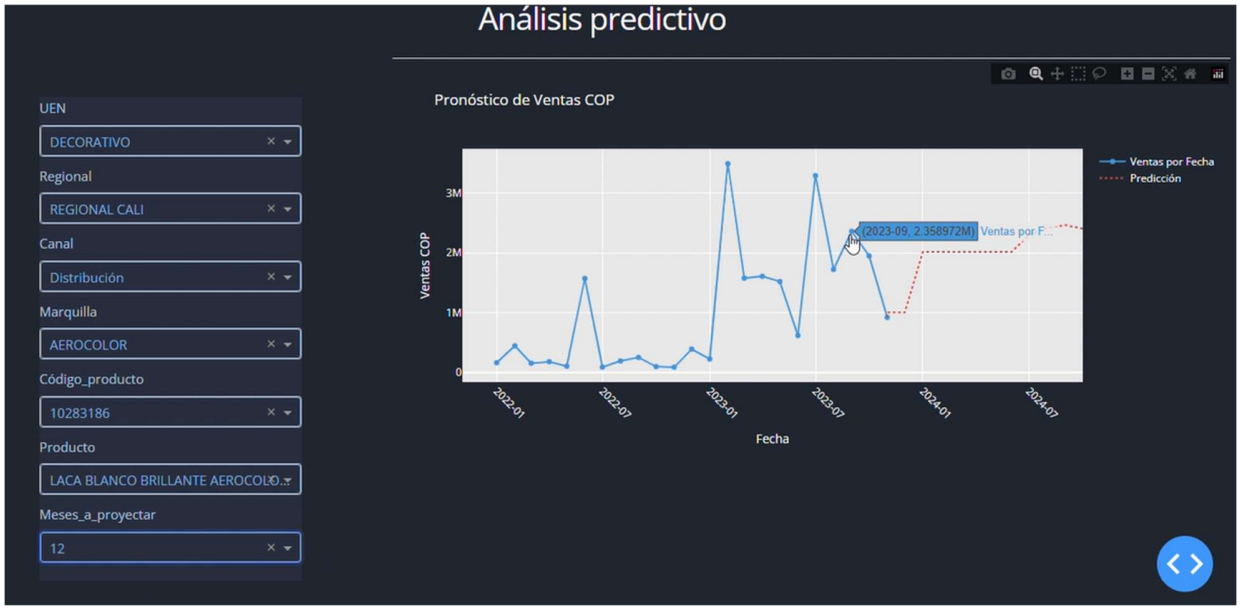
Fuente de Datos: La información utilizada para la gráfica del pronóstico de ventas se obtiene de dos fuentes principales:

1. **Histórico de Ventas:** Los datos correspondientes al histórico de ventas, para la combinación específica de UEN, regional, canal, marquilla y producto, se extraen de la capa Gold del sistema de datos.
2. **Pronóstico de Ventas:** Los datos relacionados con el pronóstico, correspondientes a la cantidad de meses indicada en el campo de entrada (input), se obtienen a través de la respuesta a una solicitud POST realizada a la API de predicción. La respuesta se entrega en formato JSON, que se transforma a la estructura requerida para integrarlo en el gráfico.

Implementación Técnica: La sección de análisis predictivo se estructura de forma modular, siguiendo el enfoque de las secciones anteriores. Para permitir la selección de los inputs, se implementaron filtros mediante componentes Dropdown en Dash. Además, se configuraron estilos específicos para el tamaño y el valor predeterminado de los filtros. Estos componentes se integran en el diseño del tablero en la Sección 3, asignándole un ancho de cuatro columnas y destinando ocho columnas para el gráfico resultante. Para la visualización de predicciones, se define un espacio dedicado en la Sección 3 del tablero, especificando el componente de gráfica que se empleará posteriormente en el callback.

Se incorporan los filtros (componente de entrada) y el gráfico (componente de salida) en el *callback* de la función principal, permitiendo así que los cambios en la selección del canal se reflejen en el tablero. Con

los campos de entrada se arma el JSON de la solicitud que se realizará al API, y posteriormente se captura la respuesta entregada. Esta respuesta se convierte a un dataFrame que es enviado a la función que genera la gráfica del pronóstico de venta.



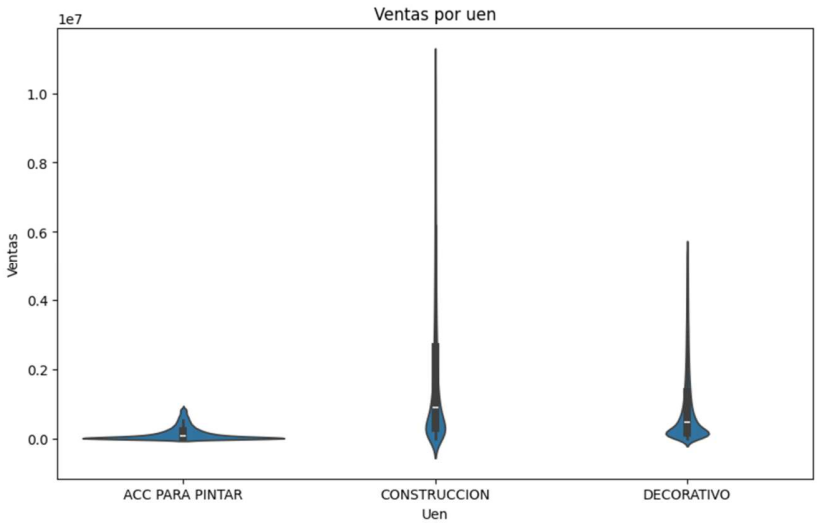
Sección 3 - Análisis predictivo del tablero desplegado

Principales resultados y conclusiones

Integración tecnológica: Para el desarrollo del proyecto se utilizaron herramientas para asegurar la trazabilidad y escalabilidad del flujo de trabajo. Esto incluye:

- **AWS EC2**: Utilizada para el despliegue de máquinas virtuales que permiten centralizar el procesamiento y gestión de modelos y datos.
- **MLFlow**: Para gestionar experimentos, registrar métricas y almacenar los modelos entrenados.
- **DVC**: Para el versionamiento de datos en diferentes niveles, facilitando la trazabilidad de los datos y su limpieza.

Modelo analítico: A través del Análisis Exploratorio de datos y el desarrollo del modelo analítico se encontró que las ventas en volumen de los productos de la compañía XYZ dependen del período del año donde se realicen, entre los meses de julio a noviembre se incrementa el volumen de meses en comparación al resto de meses del año. El canal comercial tiene una relación directa con las ventas realizadas por la compañía, en la medida en que las grandes superficies y exportadores adquieren volúmenes mayores que las adquiridas en las tiendas. También hay diferencias en las demandas de las diferentes unidades de negocios y en los productos que las componen, como se muestra a continuación. Teniendo en cuenta lo anterior, el modelo analítico desarrollado, XGBoost, contempla la relación de las variables anteriormente enunciadas con la predicción de la demanda de cada código de producto en el período deseado.



Tablero analítico: Con el desarrollo del tablero se integran análisis descriptivos y predictivos, lo cual permite a la empresa visualizar el comportamiento de las ventas históricas y realizar predicciones por región, canal y otros filtros.

Finalmente, con los resultados obtenidos se destaca la efectividad de los modelos entrenados, en especial XGBoost, para predecir las ventas de la empresa XYZ, con métricas que reflejan un ajuste adecuado con los datos históricos disponibles. El tablero analítico implementado, permite a la empresa identificar patrones clave de ventas y visualizar predicciones que permitan accionar la venta de ciertos productos.

La solución propuesta responde a la pregunta de negocio planteada y sienta bases para su evolución futura, con posibilidades de ampliar la capacidad predictiva e incorporar nuevas dimensiones de análisis para apoyar la toma de decisiones estratégicas basadas en datos.

Reporte de trabajo en equipo

En el desarrollo del proyecto, cada integrante ha contribuido desde diferentes frentes, logrando integrar los componentes necesarios para el seguimiento y análisis predictivo de las ventas de la empresa XYZ.

- **Camila Malagón:** Fue responsable de explorar y proponer diferentes modelos de regresión para predicción de ventas. Su enfoque en la selección y ajuste de modelos permitió una base sólida para el análisis predictivo del proyecto. Preparó el manual de usuario del tablero.
- **Luis David Gutiérrez:** Implementó DVC (Data Version Control) para asegurar el versionamiento de datos utilizados en el proyecto. Además, apoyó la construcción del tablero analítico y el consumo de la API desde el tablero.
- **David Alejandro Rojas:** Diseñó y desarrolló el tablero analítico y API, que integra aspectos descriptivos y predictivos, para que la empresa XYZ supere sus ventas con el nivel de detalle necesario para tomar decisiones. Construyó el manual de instalación del tablero.
- **David Zapata:** Creó una instancia en AWS para habilitar el seguimiento de modelos entrenados mediante experimentos gestionados con MLflow. También ajustó los códigos de depuración y limpieza de los datos para el modelado.

Cada miembro ha colaborado en el repositorio de GitHub del proyecto, con commits regulares, para asegurar una integración continua de los avances individuales, colaborar y organizar los entregables del

proyecto, por último, entre todos se socializó cómo se iba a presentar el video, David Zapata se encargó de su grabación/edición con los insumos del equipo encargado del tablero (David Rojas y Luis David Gutiérrez).

Soportes

a. Repositorios de los códigos:
   https://github.com/Raegar-dot/DSA-Grupo5

b. Fuentes de los modelos desarrollados:
   El código con el entrenamiento y versionado de los modelos se encuentra en la siguiente ruta:
   https://github.com/Raegar-dot/DSA-Grupo5/tree/main/notebooks

c. Fuentes de tablero y API:
   El código de la API se encuentra en la siguiente ruta: https://github.com/Raegar-dot/DSA-Grupo5/tree/main/api
   El código del tablero se encuentra en la siguiente ruta: https://github.com/Raegar-dot/DSA-Grupo5/tree/main/dash

d. Video presentación: https://youtu.be/x2_F1CjuqZk