Reporte Ejecutivo (Reto)
Predicción de Demanda por Producto para Don Colchón: Implementación de XGBoost y SARIMAX

Equipo 1

Natalia Quiroga Colorado - A01722353

Rogelio Coria López - A01733314

Mateo Zepeda - A01722398

Valeria I Sada Chapa - A00837046

Leonardo De Regil - A00837118

Análisis de Ciencia de Datos

(Gpo 101)

Prof. Rafael Martínez García Peña
Dr. Rasikh Tariq

14 de Febrero de 2023

Indice

1. Introduccion

Desde hace varios años, se comenzó a decir que “todas las empresas son empresas de datos” (Orad, 2020). Lejos quedaron los días donde únicamente big-tech almacenaba y utilizaba datos de forma masiva; en la actualidad, toda empresa competitiva almacena y utiliza datos para tomar decisiones informadas. Y no solamente es buena práctica desde una perspectiva económica y de eficiencia, si no empata con el objetivo 11 de los Objetivos de Desarrollo Sustentable (ODS) de la ONU llamada Producción y Consumo Responsable. En esta, la ONU menciona que “A las empresas les conviene encontrar nuevas soluciones que permitan modelos de consumo y producción sostenibles” (Moran, 2015). Mediante el uso de datos y modelos basados en ellos se pueden tomar decisiones inteligentes sobre la demanda de productos, pudiendo así predecir con anticipación las necesidades de los consumidores. De este modo logrando que las empresas puedan gastar menos recursos y contaminar menos de lo que harían al tomar decisiones sin estos datos y modelos. Por lo mismo, el presente documento tiene como objetivo utilizar datos para crear modelos de aprendizaje automático que ayudan a la predicción de demanda por productos, para fomentar la producción y consumo responsable.

2. Caso de Estudio

Para esta clase, Análisis de Ciencia de Datos (TC2004B), se nos asignó el socio formador Don Colchón para desarrollar nuestro proyecto de Reto. El socio formador se acercó con una problemática en su proceso de predicción de inventario en donde su modelo para predecir la venta semanal de cada uno de sus SKUs (Stock Keeping Units) a nivel nacional había decaído rápidamente de una efectividad de más del 60% hasta cerca del 40%. El socio formador buscaba que nosotros creamos un modelo que tuviera un mayor rendimiento que el que tienen actualmente, idealmente buscando resultados de 85% para arriba. Esto se busca lograr aplicando los aprendizajes de la clase en librerías de python para ciencia de datos, aprendizaje automático, métricas estadísticas, e inteligencia artificial.

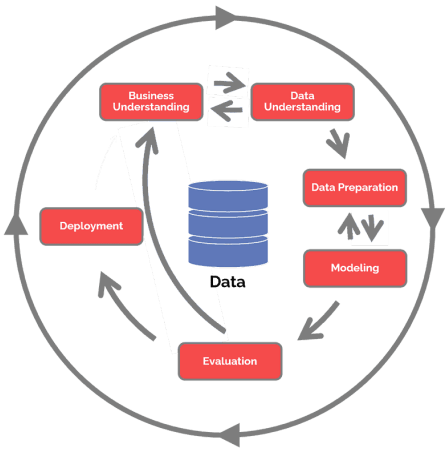
3. Metodologia CRISP-DM

Cross Industry Standard Practice for Data Mining, comúnmente abreviado como CRISP-DM, es un estándar de la industria de la ciencia de datos para realizar proyectos de forma correcta y estandarizada. CRISP-DM es un proceso cíclico de trabajo que se puede realizar las veces que sean necesarias hasta llegar a los resultados deseados, trabajando sobre los aprendizajes adquiridos en iteraciones y fases pasadas. De hecho, “las secuencia de fases no es estricta, la

mayoría de los proyectos se mueven para adelante y para atrás en las fases como sea necesario" (IBM, 2021).

Las fases de CRISP-DM son las siguientes:

1. Comprensión del Negocio
2. Entendimiento de los Datos
3. Preparación de los Datos
4. Modelado
5. Evaluacion
6. Despliegue



En la comprensión del negocio se busca entender las necesidades del negocio y que buscan obtener del proyecto. En el entendimiento de los datos se busca explorar los datos que se proveen, y en preparación de los datos se dividen, limpian, y ordenan estos. En la fase de modelado se modelan los datos con distintos modelos y variando los parámetros de estos, buscando la configuración óptima. En la evaluación se muestra el modelo final al que se llega y los hallazgos obtenidos. Finalmente, en despliegue se planean y aplican los resultados de evaluación y se realiza una recapitulación del proyecto entero.

En el presente documento se trabajará en acorde con esta metodología, y las secciones a continuación se separan al igual que las fases de CRISP-DM.

4. Comprensión del Negocio

Dol Colchón es una marca mexicana con su base de operación ubicada en Guadalupe, Nuevo León, enfocada mayormente en la venta de colchones. Ellos manejan la cadena de distribución completa siendo fabricantes y vendedores de sus productos. La empresa se fundó en 1982 y en su sitio web mencionan que "Grupo Don Colchón, cuenta con más de 60 sucursales distribuidas en el noreste de México" (Don Colchón, 2024.). Interesantemente en la sección de "Encuentra tu Sucursal" de su sitio web muestra solamente 54 sucursales, el desglose de estos se muestra en la Tabla 1.

Ciudad	**Num. de Sucursales**

Monterrey	26
Queretaro	10
Saltillo	5
San Luis Potosi	4
Torreon	4
Nuevo Laredo	2
Reynosa	2
San Miguel de Allende	1

Tabla 1: Sucursales de Don Colchón por Ciudad.

En la fase de entendimiento de los datos se retomarán las sucursales, su ubicación geográfica, y las regiones en las que la empresa las separa en la base de datos.

Actualmente la industria de los colchones atraviesa una problemática en donde el incremento de ventas debido a la pandemia, un crecimiento momentáneo y no sostenible, cosa que nos comentó el socio formador, se ha detenido de forma notable. Esto es evidente cuando los ingresos de Purple, una de las empresas de colchones más grandes de los Estados Unidos, disminuyeron más del 20% del 2021 al 2022 (Barkho, 2023). Específico al mercado Mexicano, la llegada de la empresa Alemana Emma trae una fuerte competencia extranjera al mercado de mil millones de dólares de colchones que existe en México (Echeverría, 2023).

En general la industria de los colchones es una donde las tiendas tienen relativamente pocos empleados, bajos costos operativos, un alto margen de ganancia y mantienen una baja cantidad de inventario, esto debido al tamaño del producto que hacen caro los costos de almacenamiento y transportación Para mantener un equilibrio entre la oferta y la demanda, reducir los costos operativos y mejorar la eficiencia operativa, es esencial hacer una predicción precisa de la cantidad de inventario que se moverá en la industria de los colchones. Una gestión eficiente del inventario libera capital que puede reinvertir en áreas estratégicas del negocio, lo que fomenta la innovación, la expansión y la mejora continua de la experiencia del cliente.

El Maestro Jose Sergio Serna Garza, actual Gerente de Operaciones de Dol Colchón, fue el encargado de presentarnos la problemática específica que estaríamos atendiendo, así como brindándonos acompañamiento a lo largo del proceso. Nos comentó que estaban buscando “una mejor manera de pronosticar el volumen de ventas”. Actualmente cuentan con un modelo (del cual no se nos comento el tipo de modelo utilizado) que se utiliza para la predicción de inventario basándose en la venta de los dos años anteriores. Para predecir el volumen de ventas

total, ha funcionado de manera satisfactoria para ellos pero, no han tenido el rendimiento esperado para predecir la demanda de SKUs específicos. Empezaron con una efectividad del 63% y bajaron hasta un 45% en meses recientes.

A nosotros se nos pide crear un modelo que tiene como objetivo lograr un 85% de efectividad, para la predicción de SKUs semanal a nivel nacional. Se nos informó que en la empresa se manejan 111 números de parte (SKU) de los cuales se dividen en 92 colchones y 19 bases. Cabe mencionar que se nos recalcó que a pesar de este objetivo, cualquier mejora del 45% actual tenía el visto bueno de su parte, y que además si fuera posible tener predicciones con base a regiones era bienvenido.

La métrica de efectividad que el socio formador utiliza para medir el rendimiento de sus modelos de predicciones es un promedio simple de errores de todas sus predicciones por SKU. Para propósitos del reporte este se plantea utilizar la métrica de Error Porcentual Absoluto Medio (MAPE por sus siglas en Inglés). La fórmula de la cual se saca esta métrica se muestra en la Imagen 1.

$$MAPE = \frac{1}{n} \times \sum \left| \frac{actual\ value\ - forecast\ value}{actual\ value} \right|$$

Imagen 1: Fórmula para calcular el Error Porcentual Absoluto Medio (MAPE)

MAPE es una métrica simple de obtener y útil para comparar el rendimiento de distintos modelos. Pero cabe mencionar que “no se debe de usar cuando los valores reales están cerca de cero o en cero” (Stephen Allwright, 2022), y como tenemos semanas en donde productos no se venden, tenemos que buscar una alternativa. En casos donde

5. Entendimiento de los Datos

Para el desarrollo de nuestro modelo se nos pasó una base de datos de las ventas de Don Colchón, en formato xlsx nombrado encoded_db (base de datos codificado). Esta base de datos inicial, sin limpiar y sin ningún tipo de procesamiento cuenta con las siguientes características. Cuenta con 30,565 entradas (renglones) y 28 variables (columnas). Cada renglón representa las ventas de un producto específico en un día. Los tipo de dato son los siguientes:

Tipo de Dato	**Instancias**	**Descripción**
Int64F	2	Se utilizan para índices

Float64	24	Totales de ingresos y cantidad de pedidos
Object	2	Nombre de productos y fecha

Tabla 2: Tipos de Datos en 'encoded_db'

Las columnas y nuestra interpretación de estas:

- Unnamed: 0 - Índice de entrada general del 0 a 30,565
- index - Índice de entrada por cuarto financiero empezando desde 4, el primer dia de los meses de Enero, Abril, Junio, Octubre (mes 1, 4, 7, y 10)
- Date - Fecha en formato dia mes año (ej. 04 marzo 2022) del 02 de Enero del 2022 hasta el 31 de Diciembre del 2023.
- Encoded Products - Nombres de productos en formato Producto # del producto 0 hasta el 273.
- Hay columnas de tipo "total $" y "ctdad" para las regiones Cadereyta, Coahuila, Durango, Expos F, Laredo, Monterrey, Online, Querétaro, Reynosa, San Luis, Total, e Indefinido. "Total $" correspondiendo a los ingresos y "ctdad" a la cantidad del producto que se vendió.

Por motivos de confidencialidad todos los datos de Don Colchon han sido anonimizados salvo los de región. Los ingresos y cantidades de producto vendido han sido escalados con un factor desconocido. Por lo mismo existen casos de "medios productos vendidos", por decir 3.5 colchones, esto se debe despreciar a lo largo del reporte debido al escalamiento aplicado. Además, se nos informó de parte del Maestro Serna que únicamente se incluyen los datos de colchones, y se omiten de esta base de datos las bases. La disparidad entre los 273 productos y los 92 colchones probablemente se debe a colchones de la misma línea y modelo pero en distintos tamaños siendo codificados como productos distintos, así como colchones que se descontinúan.
Retomando la distribución de sucursales vistas en la fase de Entendimiento del Negocio se formula que la distribución de ciudades a región más probable es (dado ciudades y regiones disponibles).

Ciudad	**Region Probable**
Monterrey	Monterrey (27) Cadereyta (1)
Queretaro	Queretaro (14)
Saltillo	Coahuila (5)

San Luis Potosi	San Luis (4)
Torreon	Coahuila (3) Durango (1)
Nuevo Laredo	Laredo (2)
Reynosa	Reynosa (2)
San Miguel de Allende	Queretaro (1)

Tabla 3: Regiones Probables de Cada Ciudad (Cantidad de Sucursales)

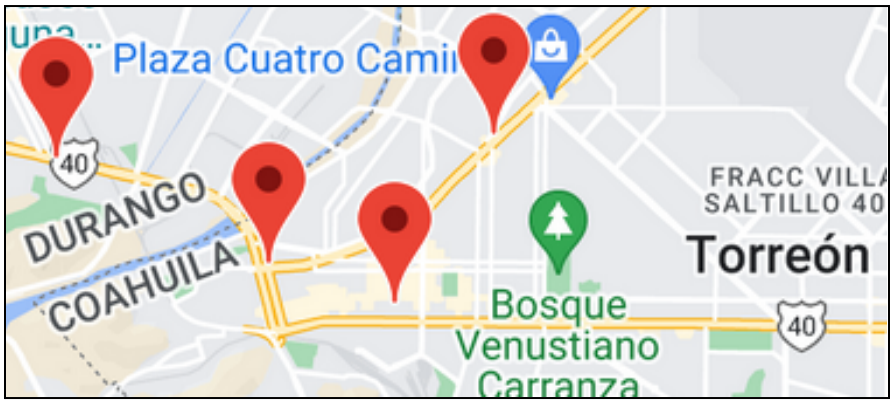


Imagen 4: Sucursales de Don Colchón en Torreón

Imagen 5: Sucursales de Don Colchón en Monterrey

En orden de mayor cantidad de sucursales las regiones serían Monterrey (27), Querétaro (14), Coahuila (8), San Luis (4), Laredo (2), Reynosa (2), Cadereyta (1), y Durango (1). De la Imagen

3 se puede ver la separación de la ciudad de Torreón en un sucursal de Durango y tres de Coahuila, similarmente sólo se atribuye una sucursal de Monterrey a la región Cadereyta por lo visto en la Imagen 4 De las columnas indefinidas, esto se atribuye a compras que no se pudieron atribuir a ninguna otra región, solo existe una fila de valores no nulos para esta categoría. De las columnas de la región Expos F, no se tiene una respuesta concreta pero pudiera ser ventas de todas las expos en las que participa la empresa.

Graficando ventas totales por región obtenemos:

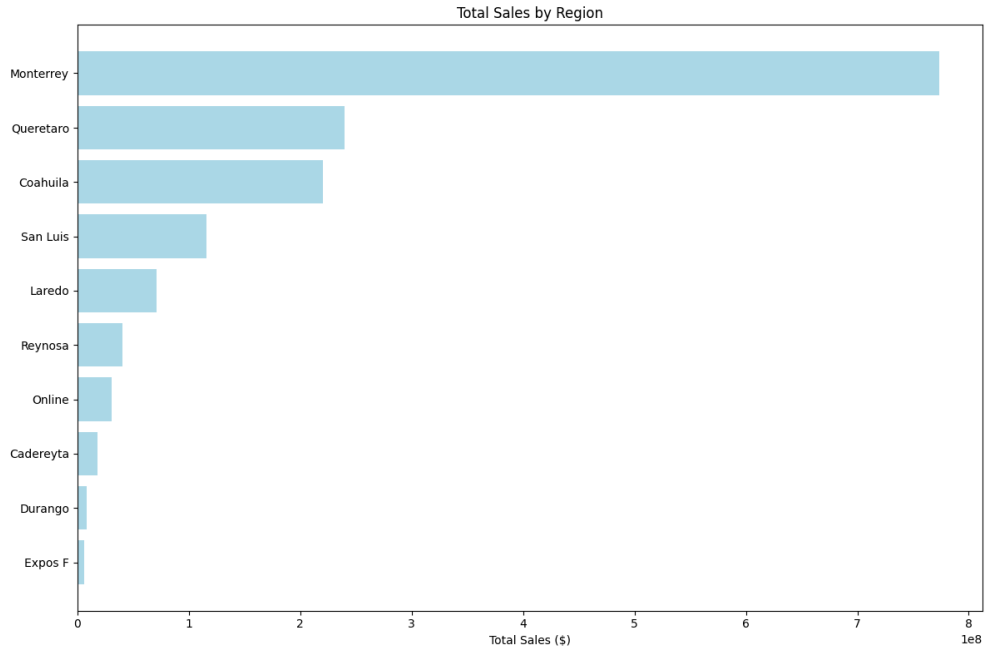


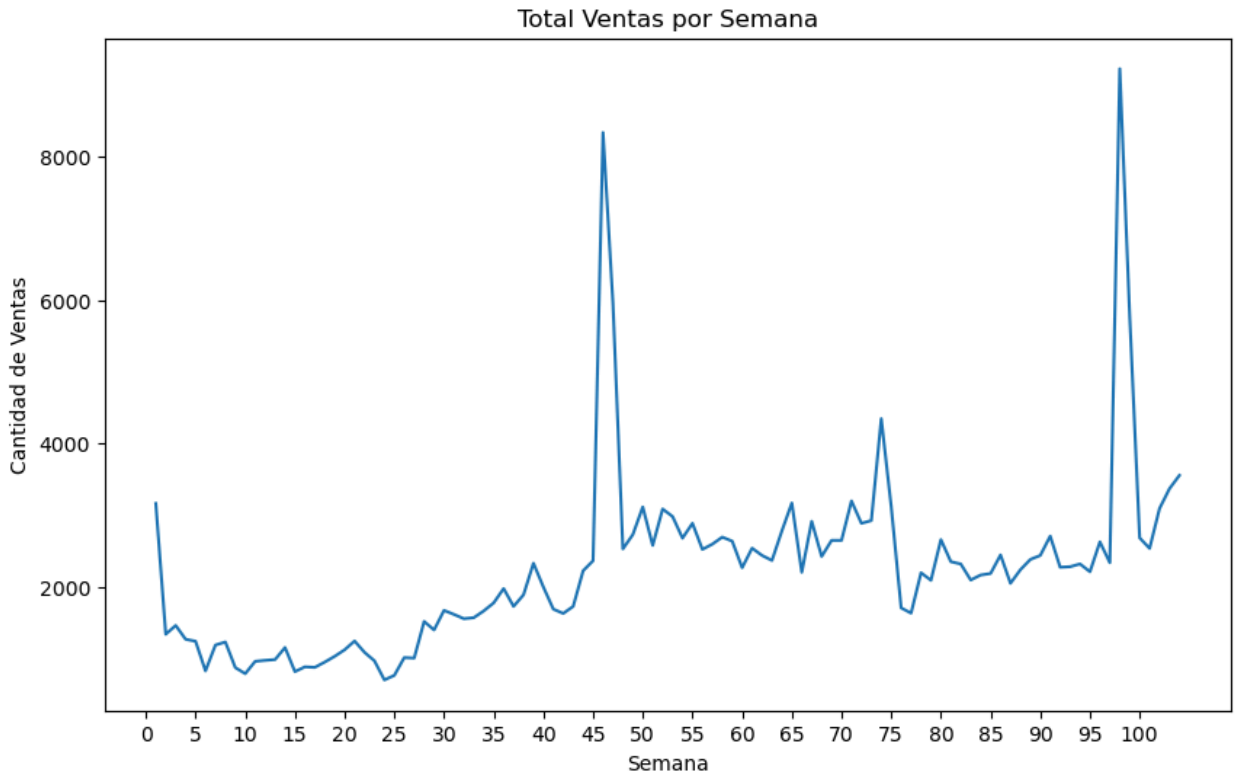
Fig 1. Ventas Totales por Región.

A simple vista de la gráfica, coinciden las ventas totales con la cantidad de sucursales que se le atribuye a cada región. Y además se hace un desglose de la ventas totales de cada región dividida por region. Esto nos permitio a entender de donde venian las ventas de Don Colchon mayormente.

Venta por sucursal por región (durante el periodo de dos años de la base de datos)

1. Laredo, 6339
2. Coahuila, 5181
3. Monterrey, 4992
4. San Luis, 4771
5. Cadereyta, 4382
6. Reynosa, 3656
7. Queretaro, 2900

8. Durango, 1757



6. Preparación de los Datos

Datos Nulos

Primeramente, se abarca el tema de los datos nulos en nuestra base de datos, utilizando la función de la librería pandas llamada isnull, obtenemos lo siguiente:

Unnamed: 0	0
index	0
date	0
Encoded Products	0
Cadereyta total $	0
Cadereyta ctdad	0
Coahuila total $	0
Coahuila ctdad	0
Durango total $	0
Durango ctdad	0
Expos F total $	0
Expos F ctdad	0
Laredo total $	0
Laredo ctdad	0
Monterrey total $	0
Monterrey ctdad	0
Online total $	0
Online ctdad	0
Queretaro total $	0
Queretaro ctdad	0
Reynosa total $	0
Reynosa ctdad	0
San Luis total $	0
San Luis ctdad	0
Total libre de impuestos	0
Ctdad Ordenada	0
Indefinido total $	26815
Indefinido ctdad	26815

Fig 2: Datos nulos en ‘encoded_db’

Únicamente se encuentran datos nulos en las columnas de ‘Indefinido total $’ e ‘Indefinido ctdad’, 26,815 cada uno. Al revisar el excel de donde provienen los datos se observa que esto se debe simplemente a que los espacios de las columnas se registraron en blanco con mayor frecuencia de lo que se registraron con un 0. Para lidiar con esto se imputará un 0 en lugar de cada valor nulo de la base datos.

Valores Atipicos

Identificar y lidiar con valores atípicos se vuelve significativamente más complicado cuando se trata de datos anonimizados. Esto debido a que no se sabe si los datos que se presentan son realmente fuera de lo que se podría considerar posible. Además, al tener contacto infrecuente y no-confidencial con el socio formador, no podemos descontar la posibilidad de que algún valor atípico se deba a una situación extemporánea como lo puede ser la liquidación de algún producto o un pedido masivo de parte de un hotel (Don Colchón tiene un apartado de hoteles en su sitio web).

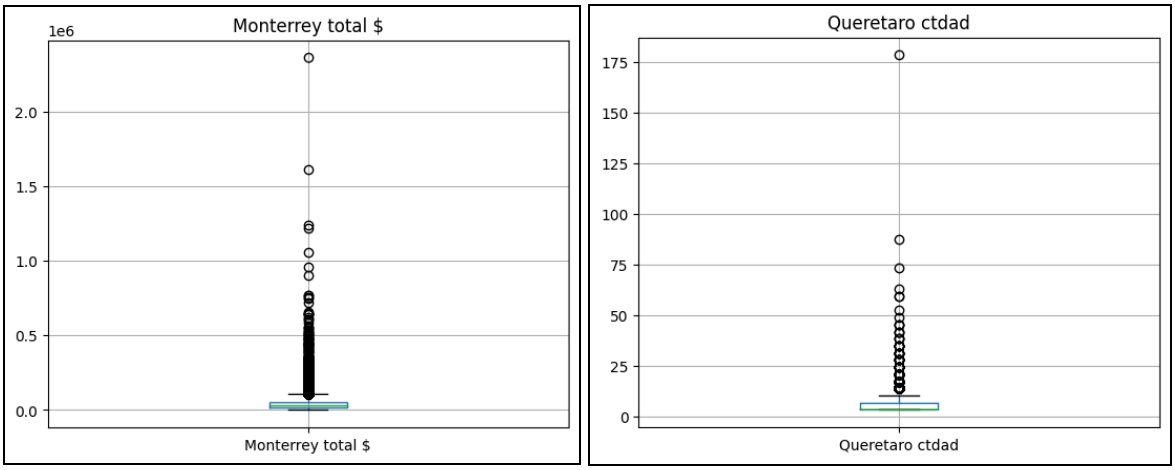


Fig 5.1 & Fig 5.2
Boxplots for 'Monterrey total $' and 'Queretaro ctdad'

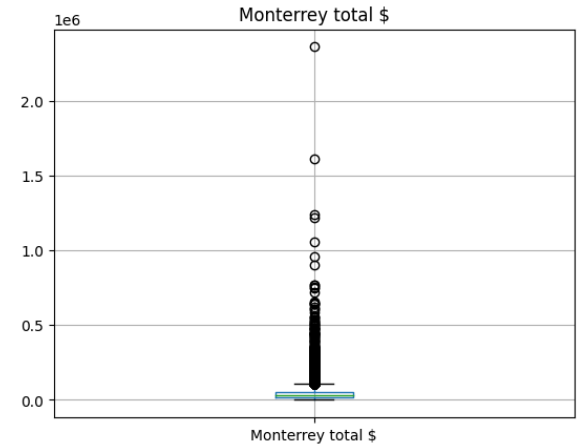
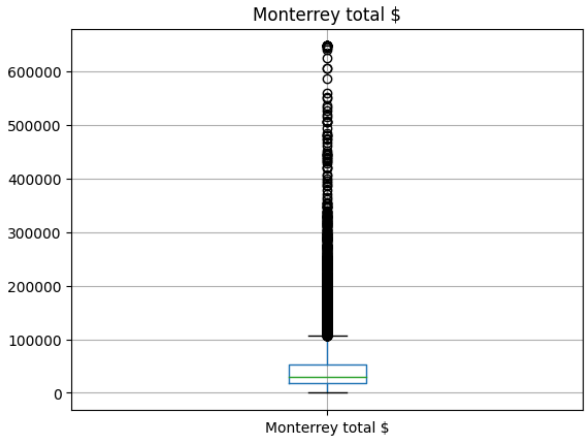
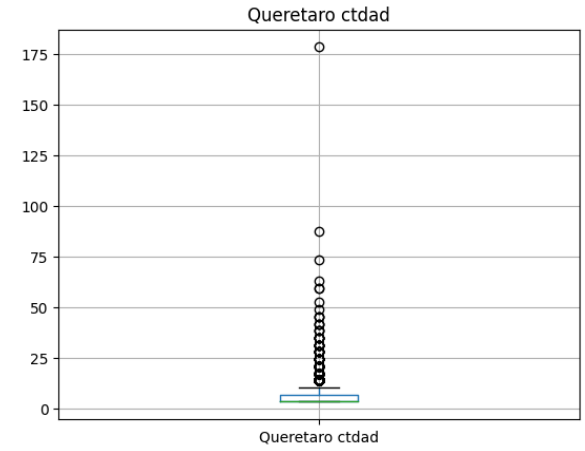
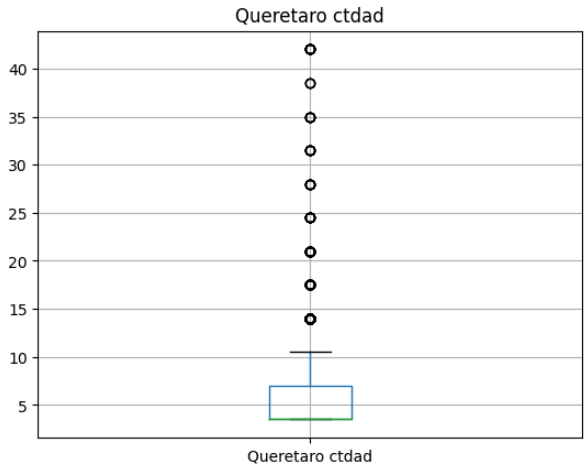
Al crear diagramas de cajas para cada columna (descontando valores igual a 0 para facilitar la visualización e interpretación de estas) se pueden ver varios casos de preocupación como lo son la figura 5.1 y 5.2.

Fig 5.1.1
Fila donde se encuentra el valor máximo de la columna 'Monterrey Total $'

Al observar el caso particular de la figura 5.1 se puede ver como el Producto 58 el 23 de Febrero del 2022 se vende unicamente en Monterrey, un total de 280 veces por $2,360,013.81, como se ha comentado anteriormente al ser anonimizados, estos datos significan pocos por si solos. Sin embargo, dado que se estandarizan por igual los datos, la diferencia entre un dato como el de la fila mostrada en la figura 5.1.1 y las demás instancias mostradas en la figura 5.1 si nos causa ruido. Tomando como supuesto, alguna venta masiva de un producto como el caso anteriormente comentado de un pedido de un hotel, consideramos que aunque este tipo de entradas atípicas son ciertas, no son útiles para los propósitos de nuestro modelo. Nuestro modelo no será capaz de predecir cuándo Don Colchón obtendrá un pedido de esa escala, y por lo mismo es mejor lidiar con estos datos.

Para tratar con los datos erróneos se opta por el método de winsorizacion. En este método se elige un porcentaje de winsorizacion que corresponderá a los percentiles límite para tus datos. Por ejemplo en una winsorizacion del 80% significa que quieres quedarte con el 80% de tus datos originales y se tomará como límite inferior el percentil 10 y límite superior el percentil 90, todo dato menor que el percentil 10 se aumentaría al valor de este y todo valor mayor que el percentil 90 se disminuye al valor de este.

En nuestro caso se opta por una winsorizacion del 99.9%, es decir nuestro percentil inferior es el 0.05 y nuestro percentil superior es el 99.95. Se ajusta el 0.1% de los datos, o dicho simplemente se ajusta 1 de 1000 entradas. Esto hecho para arreglar la problemática específica en nuestras regiones grandes como lo es Monterrey y Querétaro.

Gráficas de Caja Pre-Winsorizacion	**Gráficas de Caja Post-Winsorizacion**
	
	

El impacto es notable…

Columnas Innecesarias

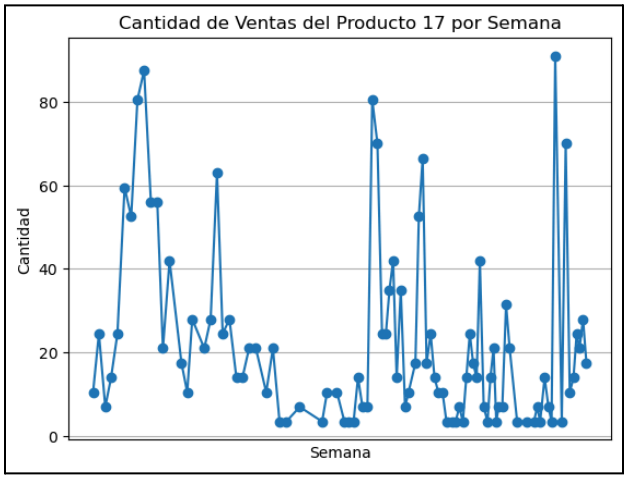
Recordando lo visto en la fase de Entendimiento de Datos, contamos con dos columnas de índices que debido a que excel y pandas ya tienen índices por default, no son de particular ayuda y las quitaremos para simplificar nuestro trabajo.

Lidiar con Columna ‘date’

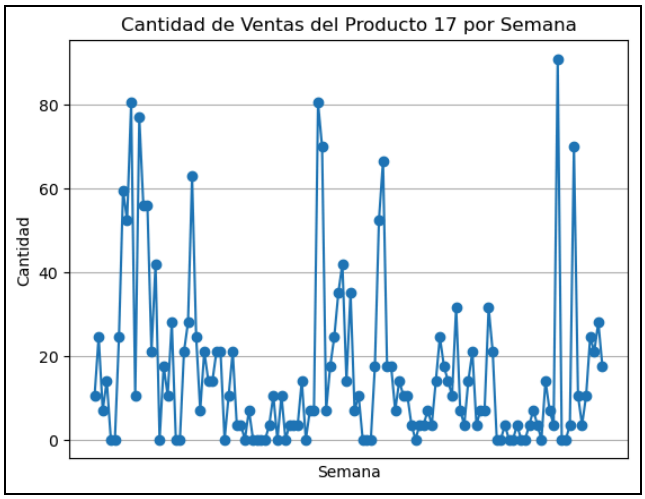
La columna ‘date’ que contiene la fecha de compra de cada producto, es una de tremenda importancia para realizar las predicciones pero requiere de preprocesamiento para sacarle la mayor cantidad de provecho. Primeramente se convierte la variable a una de datetime, haciendo un listado de traducciones de meses como números para evitar fechas que no se traduzcan. Posteriormente se aprovechan los parámetros de los objetos datetime y se crean columnas de Cuarto, Mes, y Semana. En su artículo *Machine Learning with Datetime Feature Engineering* Andrew Longo menciona ‘Otra cosa que quizás quieras hacer es convertir el día de la semana en una variable categórica mediante codificación one-hot. Sin embargo, no necesitamos hacer estas cosas para un método basado en árboles.’ Con base en esto se hará una base de datos con one-hot encoding y otra sin, dependiendo del uso de métodos basados en árboles de decisión o no.

Agrupación por Semana

Puesto que nuestro objetivo es hacer predicciones semanales por SKU se agrupa cada semana de ventas de cada producto en una nueva dataframe, con esta nos permite empezar a hacer gráficas como la siguiente:



Solo que existe un problema con esta gráfica, únicamente incluye semana en donde el producto se vendió a nivel nacional, esto hará que nuestro modelo solo realice predicciones con base en las semanas con ventas cuando en verdad hay muchas semanas en donde los productos se venden 0 veces. Es por eso que los agrupamos de manera distinta ahora teniendo una fila por semana y una columna por cada producto que se vende. Graficando esto nos da una grafica mas adecuada.



Eliminación de productos:

Después de la segunda junta con el socio formador, nos comentó de la discontinuación de productos, lo cual era responsable por la gran cantidad de productos en la base de datos. Al analizar los distintos productos y sus ventas a través de los años, se puede notar que algunos de los productos dejaron de ser vendidos. Creamos una función que elimina todos los productos sin ventas desde noviembre de 2023 Adicionalmente, debido a la gran cantidad de entradas necesarias para realizar un modelado adecuado, se eliminaron los productos que tuvieran menos de 80 semanas de datos, reduciendo nuestra base de datos a solo 98 productos y por casi 10,000 entradas.

Tidy Data

Con base en el artículo de Hadley Wickham nombrado *Tidy Data*, buscamos que nuestra base de datos cumpla con los siguientes tres principios.

1. Cada variable es una columna
2. Cada observación es un renglón
3. Cada celda es una sola medida

Puesto que se cumple con los tres requisitos, nos movemos hacia la fase de final de la fase de preparación de datos.

Test/Training/Validation Splits

Elegir particiones adecuadas para nuestros grupos de prueba, entrenamiento, y validación es esencial para el rendimiento del modelo, y hay un sin fin de particiones posibles. El Dr. Gomede explica varios métodos posibles para distintos casos:

- Simple, ej. (70% entrenamiento, 15% prueba, 15% validación)
- Particiones basadas en tiempo, necesario para trabajar con series de tiempo
- Leave-One-Out-Cross-Validation (LOOCV), cada dato se usa de validacion una vez

Es posible en nuestro caso trabajar con los tres, cada uno en momentos y por motivos diferentes. La partición simple nos servirá inicialmente para probar modelos de forma rápida y poder ajustar hiperparametros entre modelados, particiones basadas en tiempo se utilizaran cuando trabajamos con ese programa de series de tiempo, y LOOCV para probar modelos finales con mayor minuciosidad. En la literatura el estándar es dejar entre 60%-80% de los datos para entrenamiento por lo que optamos por el punto medio de 70%, dejando 15% para testing y 15% para validación.

7. Modelado

“Classification predicts whether something happens, while regression predicts how much something will happen”

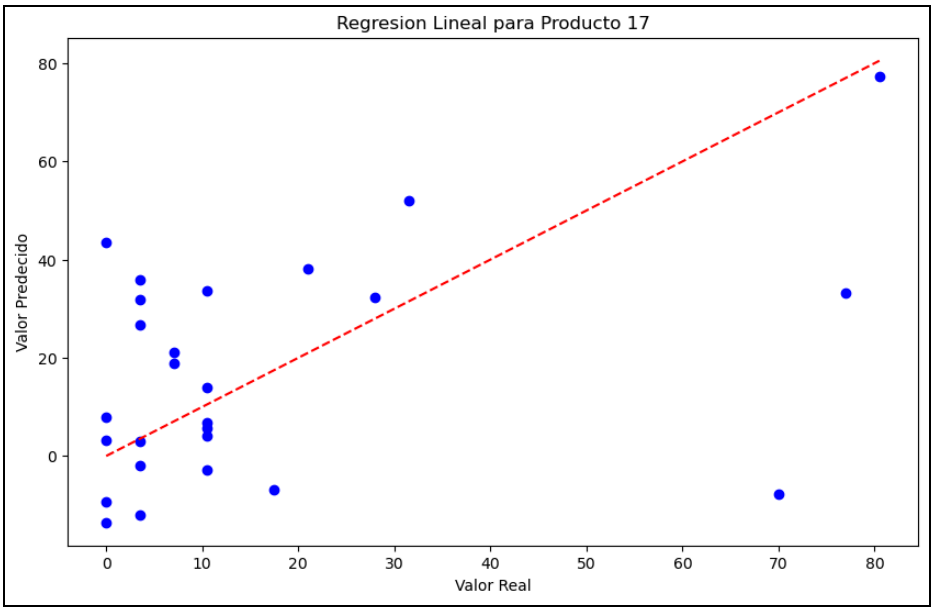
- Provost & Faucet

Como es evidente de leer esta frase de parte de los Autores de *Data Science for Business* nuestro problema es uno de regresión en donde buscamos predecir cuántas ventas se realizarán de cada producto. Mejor información y mejor modelo, menor la incertidumbre semana a semana.

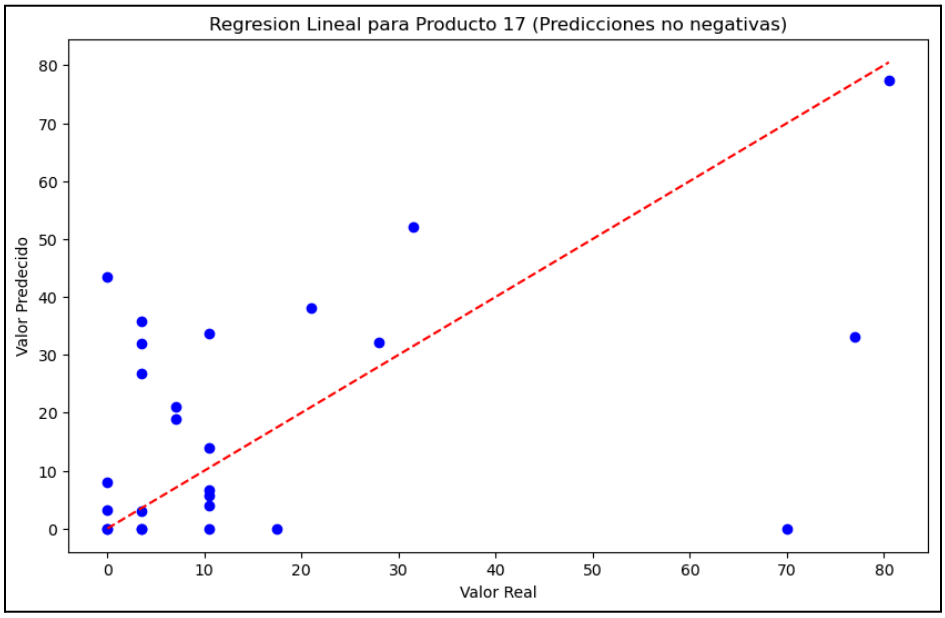
Existen un sinfín de modelos de aprendizaje automático, cada uno mejor adaptado para diferentes objetivos y diferentes datos. Más aún, cada uno de estos modelos cuenta con una cantidad extensa de hiper parámetros que aumentan la complejidad de elegir el mejor modelo para nuestro problema. Se realizan pruebas

Regresion Lineal

Un modelo de regresión que busca una relación lineal entre la variable objetivo y las variables independientes. Se utiliza la librería de Scikit learn para la siguiente gráfica del modelo.



Notamos que nuestro modelo está prediciendo valores negativos, cosa que no es posible para el número de ventas en un día. Se agrega una línea para convertir los negativos a 0.



SARIMAX (seasonal Auto-Regressive Integrated Moving Average with eXogenous Regressors)

Este modelo nace de ARIMA (Autoregressive Integrated Moving Average) pero agregando el aspecto importante de temporalidad. La parte de eXogenous regressors tiene que ver con cómo se lidea con variables que se influyen por cosas fuera del modelo. Cosa que aludió el socio formador a ser un factor importante debido a compras elevadas durante ciertos meses y ventas como el Buen Fin o Hot Sale.

Este modelo se basa en dos ‘grupos’ de hiperparametros, los temporales y los no temporales

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\text{Non-seasonal part of the model}} \quad \underbrace{(P,D,Q)_m}_{\text{Seasonal part of of the model}}$$

m = number of observations per year;

P = Number of seasonal Autoregressive terms;

D = Number of seasonal differences;

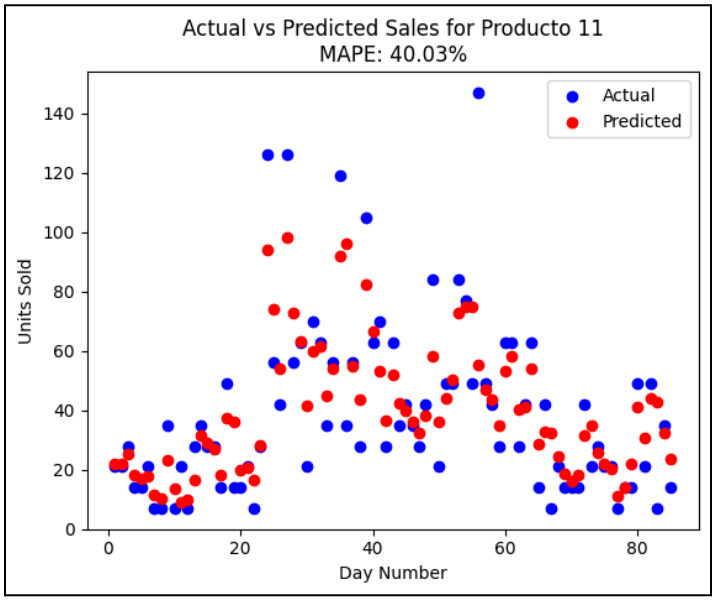
Q = Number of seasonal Moving Average terms

(Santa R, 2023)

Tomando un split simple de 80% para entrenamiento y 20% para prueba (esto tomado en orden ya que SARIMA funciona con series de tiempo). Empezamos con parámetros (1,1,1) (1,1,1,52)

<u>Random Forest Regression</u>

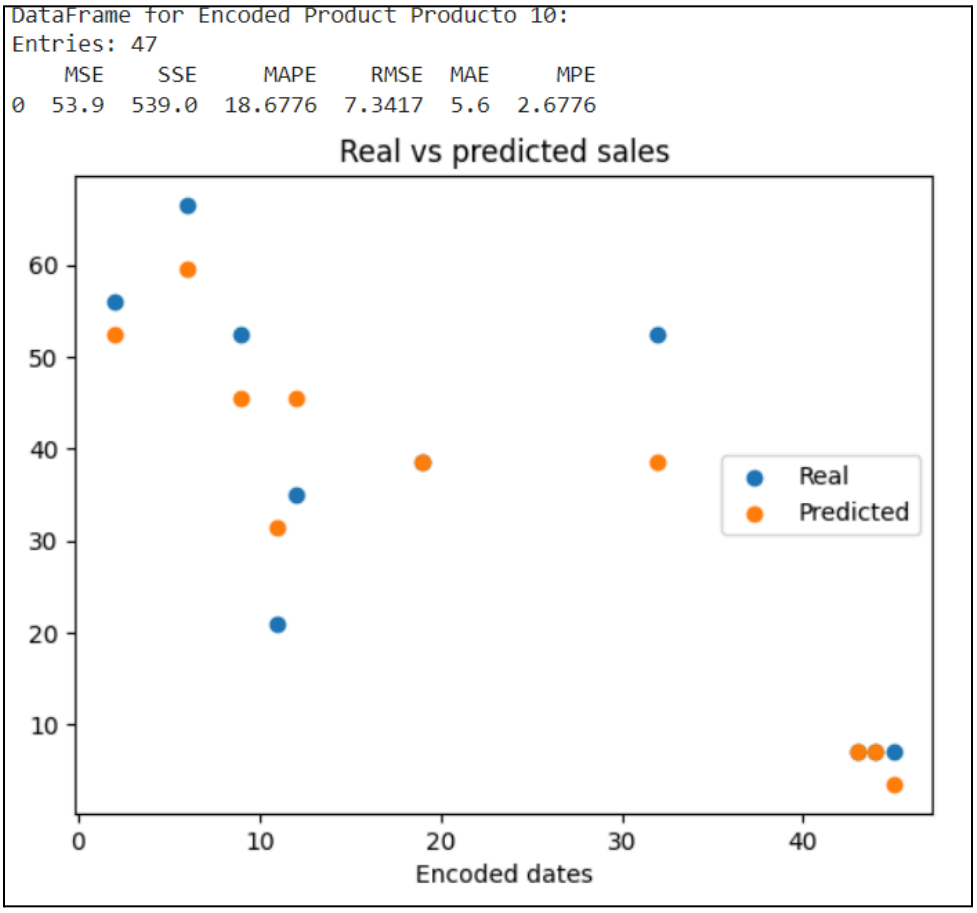
El método de Bosques Aleatorios opera a través de árboles de decisión en los cuales se generan numerosos árboles de decisión distintos, configurando lo que se conoce como el bosque de Regresión de Bosques Aleatorios. Colectivamente, estos árboles individuales contribuyen a realizar una predicción de la variable objetivo.



Decision tree:

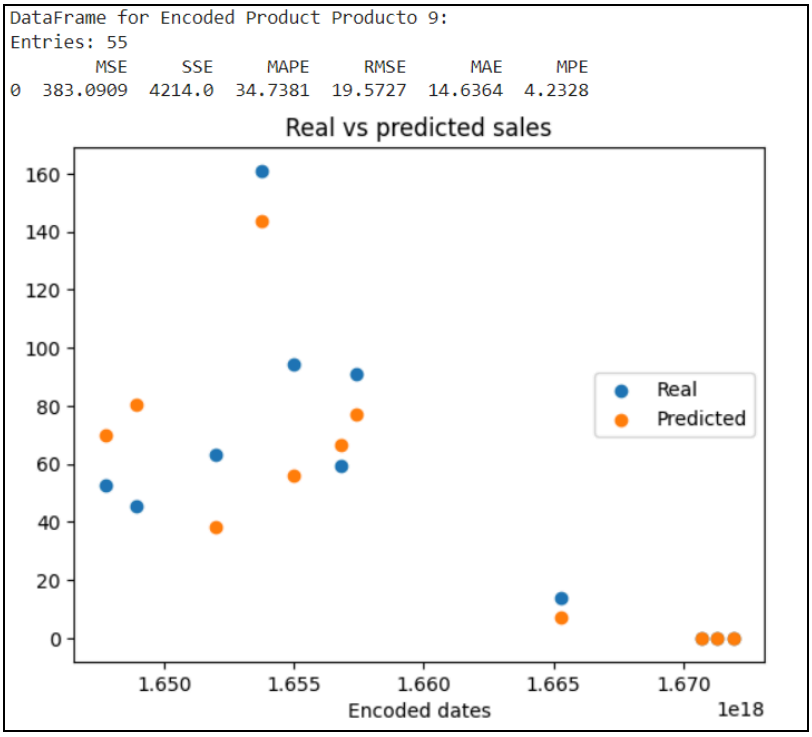
Modelo utilizado para la regresión. Se basa en la idea de dividir los datos en subconjuntos más pequeños, utilizando reglas de decisión basadas en las características de los datos. Es un modelo de aprendizaje automático supervisado que predice valores continuos. Es recursivo. Tiene los siguientes pasos

1. Dividir los datos en categorías dependiendo de los valores de cada variable independiente.
2. Criterios de splitting: Se calculan diferentes criterios de decisión que determinarán la secuencia tomada durante la predicción para minimizar algún estimado como MAE, MSE y otros más. En este caso, usamos MAE(Mean Absolute Error)
3. Expansión del árbol. El árbol se expande, creando criterios de splitting hasta que se llega a una máxima profundidad.
4. Nodos Hoja: Se crean los últimos nodos del árbol que contienen los valores que se van a predecir si se llega a ellos.
5. Predicción: El árbol recibe un input y basándose en los criterios de decisión, se recorre el árbol en cierta secuencia para llegar a una predicción.



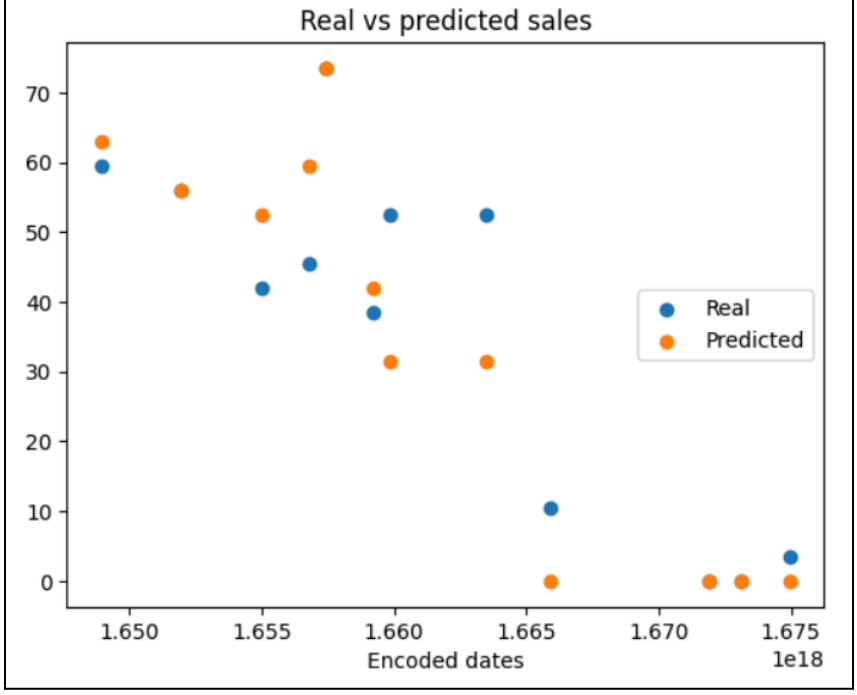
DataFrame for Encoded Product Producto 9:
Entries: 55

	MSE	SSE	MAPE	RMSE	MAE	MPE
0	383.0909	4214.0	34.7381	19.5727	14.6364	4.2328



DataFrame for Encoded Product Producto 4:
Entries: 59

	MSE	SSE	MAPE	RMSE	MAE	MPE
0	111.2708	1335.25	35.0742	10.5485	7.2917	20.9258



XGBoost Regressor

El Regresor XGBoost (eXtreme Gradient Boosting) pertenece a los métodos de aprendizaje de conjunto, específicamente boosting, que combina múltiples aprendices débiles para formar un modelo predictivo fuerte. En XGBoost, los árboles de decisión sirven como los aprendices base, y el algoritmo construye árboles secuencialmente para corregir los errores cometidos por los anteriores. Emplea el aumento de gradiente, donde cada nuevo árbol se entrena utilizando los gradientes de la función de pérdida del árbol anterior, mejorando gradualmente el rendimiento del modelo. Para prevenir el sobreajuste, XGBoost incorpora técnicas de regularización y poda de árboles. Optimiza una función objetivo específica, como la pérdida de error cuadrado, durante el entrenamiento.

El Regresor XGBoost tiene 8 pasos principales para generar una predicción:

1. **Inicialización del Modelo.**

2. **Cálculo del Error**: Se calcula el error residual entre las predicciones del modelo actual y los valores reales del conjunto de datos de entrenamiento.

3. **Ajuste del Árbol de Decisión**: Se ajusta un árbol de decisión a los errores residuales. Este árbol de decisión se diseña de manera que minimice la función de pérdida, que es una medida del error entre las predicciones del modelo y los valores reales.

4. **Cálculo de la Importancia de las Características**: XGBoost calcula la importancia de cada característica en función de cuánto reduce la función de pérdida. Las características que más reducen la pérdida se consideran más importantes para la predicción.

5. **Actualización del modelo:** Se agrega el árbol de decisión ajustado al modelo existente, y se actualizan las predicciones sumando la predicción del nuevo árbol ponderada por una tasa de aprendizaje.

6. **Regularización:** Se aplican técnicas de regularización para evitar el sobreajuste y mejorar la generalización del modelo. Esto puede incluir la penalización de la complejidad del modelo o la limitación del número de nodos hoja en los árboles.

7. **Iteración**: Los pasos 2-6 se repiten varias veces, agregando árboles de decisión adicionales y mejorando gradualmente las predicciones del modelo.

8. **Predicción**: Una vez que se ha construido el modelo final, se utilizan todas las predicciones de los árboles para predecir valores para nuevos datos.

Especificaciones del Modelo

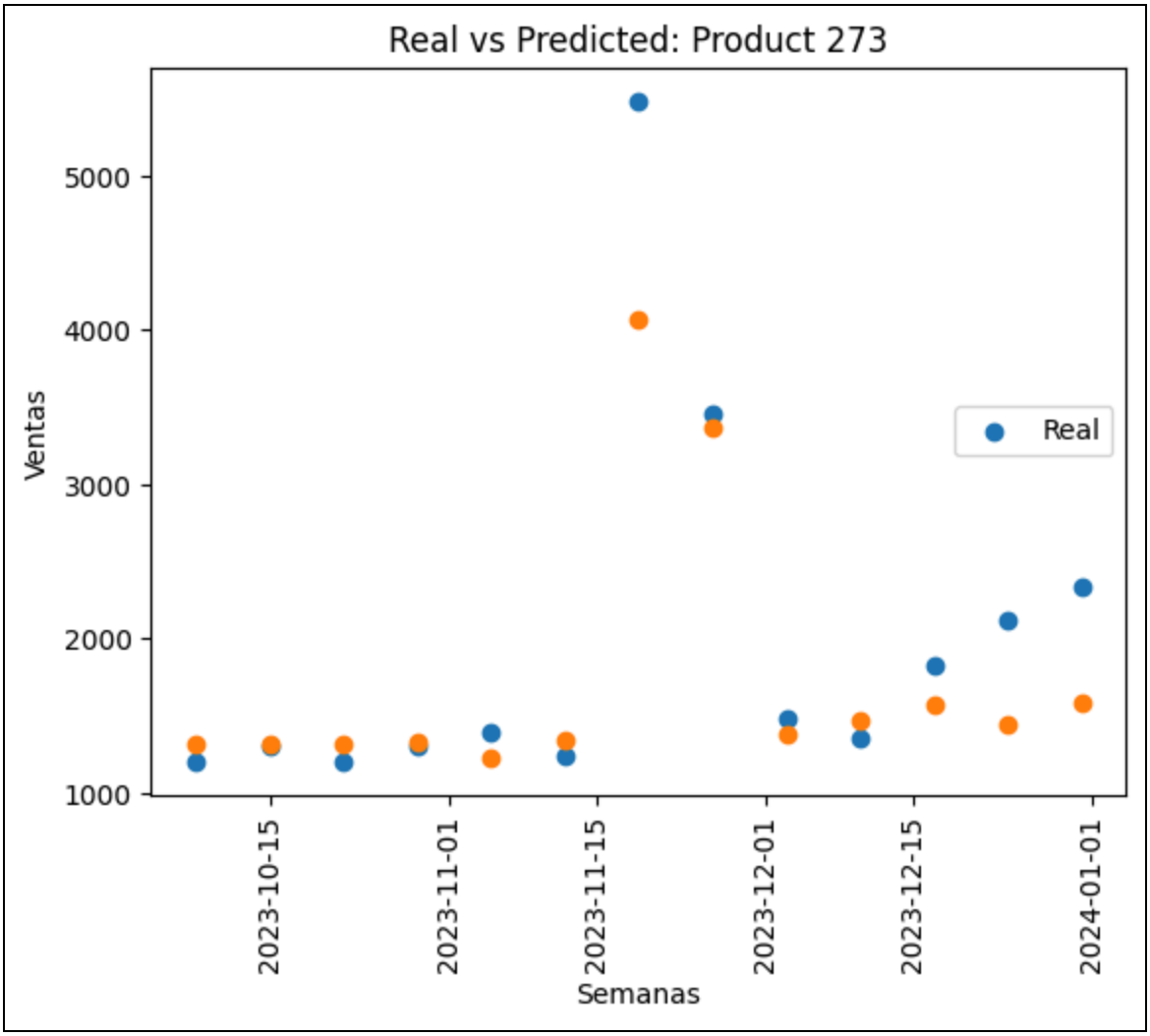
```
xgb_r = xg.XGBRegressor(objective ='reg:linear',
        alpha = 10,
        n_estimators = 10,
        seed = 123)
```

- **objective='reg:linear'**: Este es un parámetro del constructor que especifica la función de pérdida que se utilizará para entrenar el modelo. En este caso, 'reg:linear' indica que estamos construyendo un modelo de regresión lineal.

- **alpha=10**: Este es un parámetro opcional del constructor que controla la regularización L1 (también conocida como regularización de Lasso) del modelo. Un valor más alto de alpha aumenta la fuerza de la regularización, lo que puede ayudar a prevenir el sobreajuste. En este caso, se ha establecido en 10.

- **n_estimators=10**: Este es otro parámetro del constructor que especifica el número de árboles que se utilizarán en el modelo de XGBoost. Cuantos más árboles se utilicen, más complejo será el modelo y más probable será que se ajuste demasiado a los datos de entrenamiento. Aquí se ha establecido en 10, lo que significa que utilizaremos 10 árboles.

Aquí se muestra una gráfica donde se aplica el modelo de XGBoost con las especificaciones antes mencionadas, donde el eje ‘y’ es el número de ventas y el eje ‘x’ la semana.



Aquí podemos ver los resultados de nuestro modelo, a simple vista los valores predichos contra los valores reales se ven muy cercanos, este modelo en específico consiguió un MAPE de 15.05% y un MAE de 506.

Teniendo esto, se eligieron XGBoost, DT y RF como los modelos a automatizar y utilizar y se crearon los siguientes métodos.

DT vs. XGBoost vs. RF:

Método #1: Normal Split

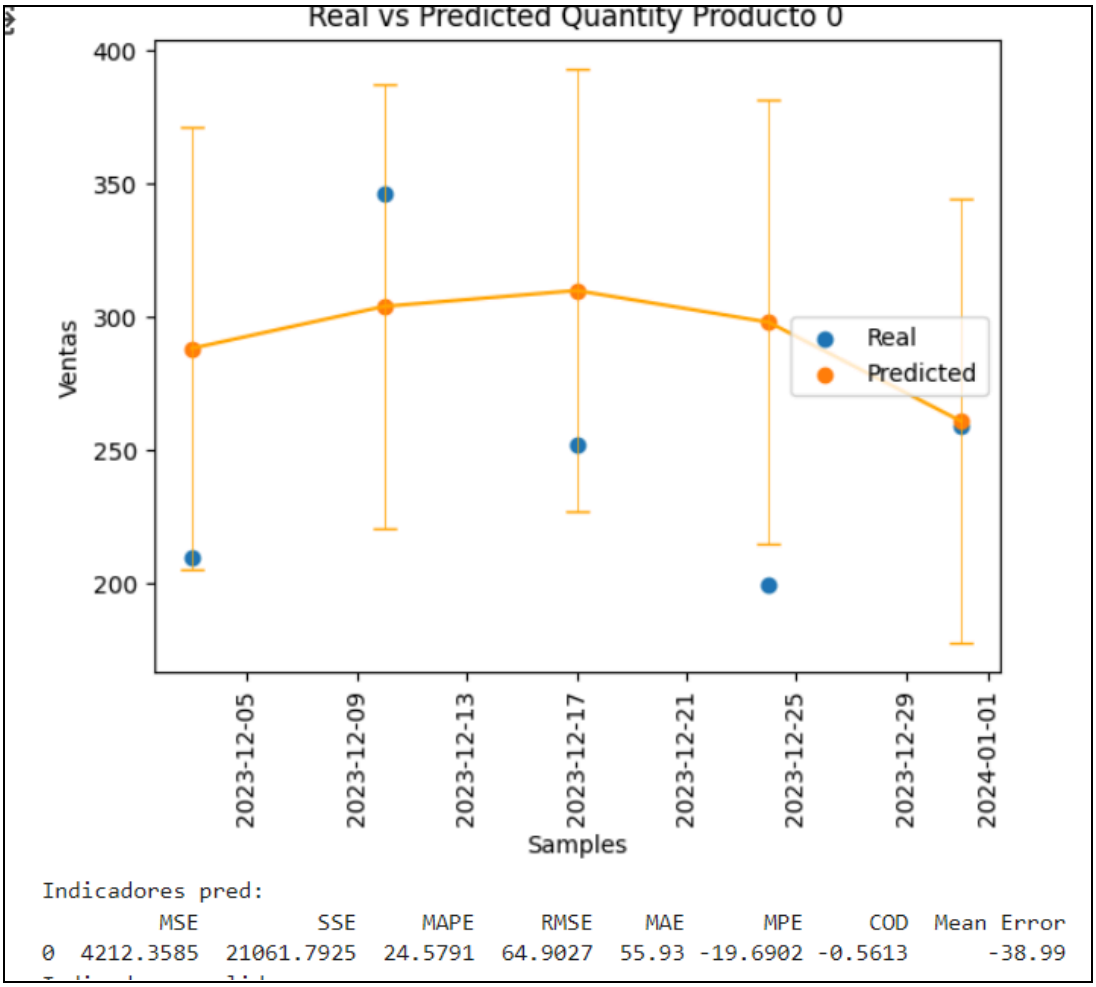
Una vez teniendo una variedad de modelos con sus respectivos hiperparametros, se crea un modelo de decision tree, uno de XGBoost y uno de Random Forest para cada uno de los productos que hayan tenido ventas en los últimos dos meses. Para entrenarlo, se dividen los datos en tres partes. Training y validation son todos los datos anteriores al primero de diciembre 2023. De estos datos, se realiza un split de 0.7/0.3 para training y validation respectivamente. El tercer grupo es Testing que son todos los datos posteriores al primero de diciembre de 2023, ya que queremos probar que nuestro modelo es capaz de predecir hacia el futuro, no solo dentro del rango de fechas que ya tiene. Teniendo esto en cuenta se entrenan todos los modelos usando Training y posteriormente, se validan con Validation data. Después de esto se elige el modelo con menor MAEy se prueban los datos con el set Test para cada producto. De esta manera podemos generar cientos de modelos en cuestión de segundos y obtener el más óptimo con su respectivo error.

Ejemplo: Producto 0

Una vez concluido el entrenamiento, se revisan estadísticos de la validación:

```
XGB:
         MSE           SSE    MAPE     RMSE      MAE     MPE     COD
0  3969.8575  119095.7259  24.612  63.0068  53.2406 -3.4226  0.3147
DT:
         MSE         SSE     MAPE      RMSE      MAE      MPE     COD
0  11609.325  348279.75  37.7596  107.7466  83.1833 -11.7322 -1.0041
RF:
         MSE           SSE     MAPE    RMSE      MAE     MPE     COD
0  4255.2131  127656.3925  24.6079  65.232  52.9317 -7.8972  0.2654
```

Se puede observar que en tanto MAPE como MAE, Random Forest, supera las expectativas, por lo que se elige Random Forest y se predice el mes de diciembre.



Indicadores pred:

	MSE	SSE	MAPE	RMSE	MAE	MPE	COD	Mean Error
0	4212.3585	21061.7925	24.5791	64.9027	55.93	-19.6902	-0.5613	-38.99

Podemos observar que aunque el MAPE sea prometedor, se tiene un MAE de 55, lo que aunque no es alto para una predicción con media de 300, si puede llevar a una gran subestimación o sobreestimación que resulte en la pérdida monetaria del socio formador.

Look-ahead-bias:

Cuando se habla de una serie de tiempo y la predicción de una variable dependiente, la variable independiente de fecha o tiempo siempre va a ser mayor al rango usado para entrenamiento y validación, por lo que existe Look-ahead-bias. Esto se debe a que la selección del modelo con la validación está basada en valores que no se encontraran durante la aplicación del modelo, en otras palabras, se está validando con datos que son generalizados dentro del rango de tiempo de entrenamiento y no a futuro. Por lo que queda rechazado y se buscará un método diferente.

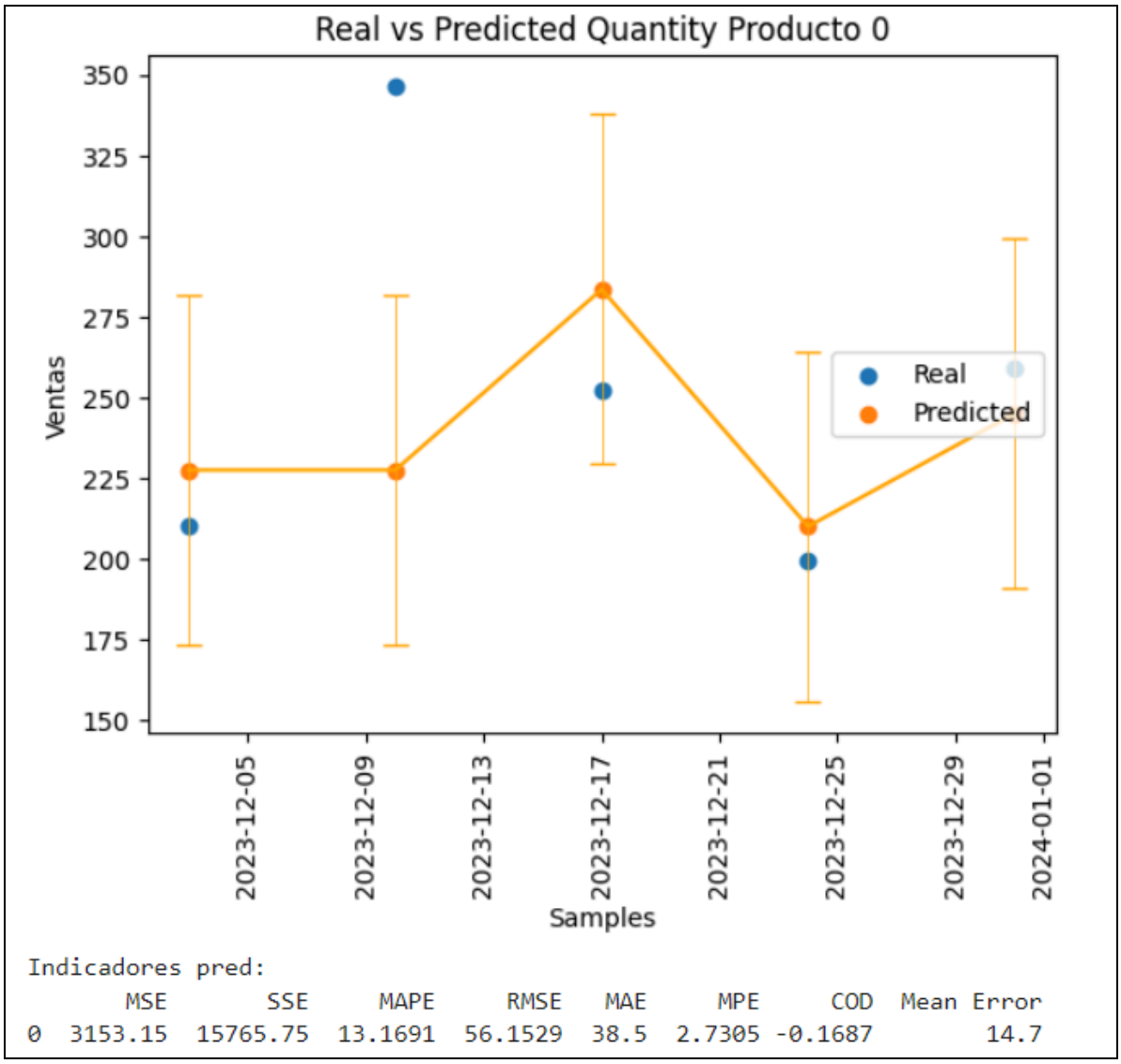
Método #2: Time series split

Una vez teniendo una variedad de modelos con sus respectivos hiperparametros, se crea un modelo de decision tree, uno de XGBoost y uno de Random Forest para cada uno de los productos que hayan tenido ventas en los últimos dos meses. Para entrenarlo, se dividen los datos en tres partes. Training que incluye todos los datos anteriores al primero de octubre 2023. Validation que incluye todos los datos anteriores al primero de diciembre de 2023 pero posteriores al al primero de octubre 2023. El tercer grupo es Testing que son todos los datos posteriores al primero de diciembre de 2023, ya que queremos probar que nuestro modelo es capaz de predecir hacia el futuro, no solo dentro del rango de fechas que ya tiene. Teniendo esto en cuenta se entrenan todos los modelos usando Training y posteriormente, se validan con Validation data. Después de esto se elige el modelo con menor MAEy se prueban los datos con el set Test para cada producto. De esta manera podemos generar cientos de modelos en cuestión de segundos y obtener el más óptimo con su respectivo error.

Ejemplo: Producto 0

```
XGB:
         MSE         SSE     MAPE     RMSE      MAE      MPE     COD
0  9597.1541  76777.2325  28.6342  97.9651  79.9827  11.2424 -0.6557
DT:
         MSE      SSE     MAPE     RMSE    MAE     MPE     COD
0  4014.9375  32119.5  24.0603  63.3635  54.25 -9.1663  0.3074
RF:
         MSE         SSE     MAPE     RMSE      MAE     MPE     COD
0  8447.4316  67579.4525  32.2577  91.9099  79.8438 -3.0461 -0.4573
```

Por haber usado un diferente set de entrenamiento, se tienen diferentes errores. Se puede observar que los errores son mayores al Método 1. Esto es porque los datos de validación ahora son a futuro. Analizando, obtenemos el mejor modelo que es Decision Tree Regression Bias con Mae de 54.25 y comprobamos con el set Test:



Indicadores pred:

	MSE	SSE	MAPE	RMSE	MAE	MPE	COD	Mean Error
0	3153.15	15765.75	13.1691	56.1529	38.5	2.7305	-0.1687	14.7

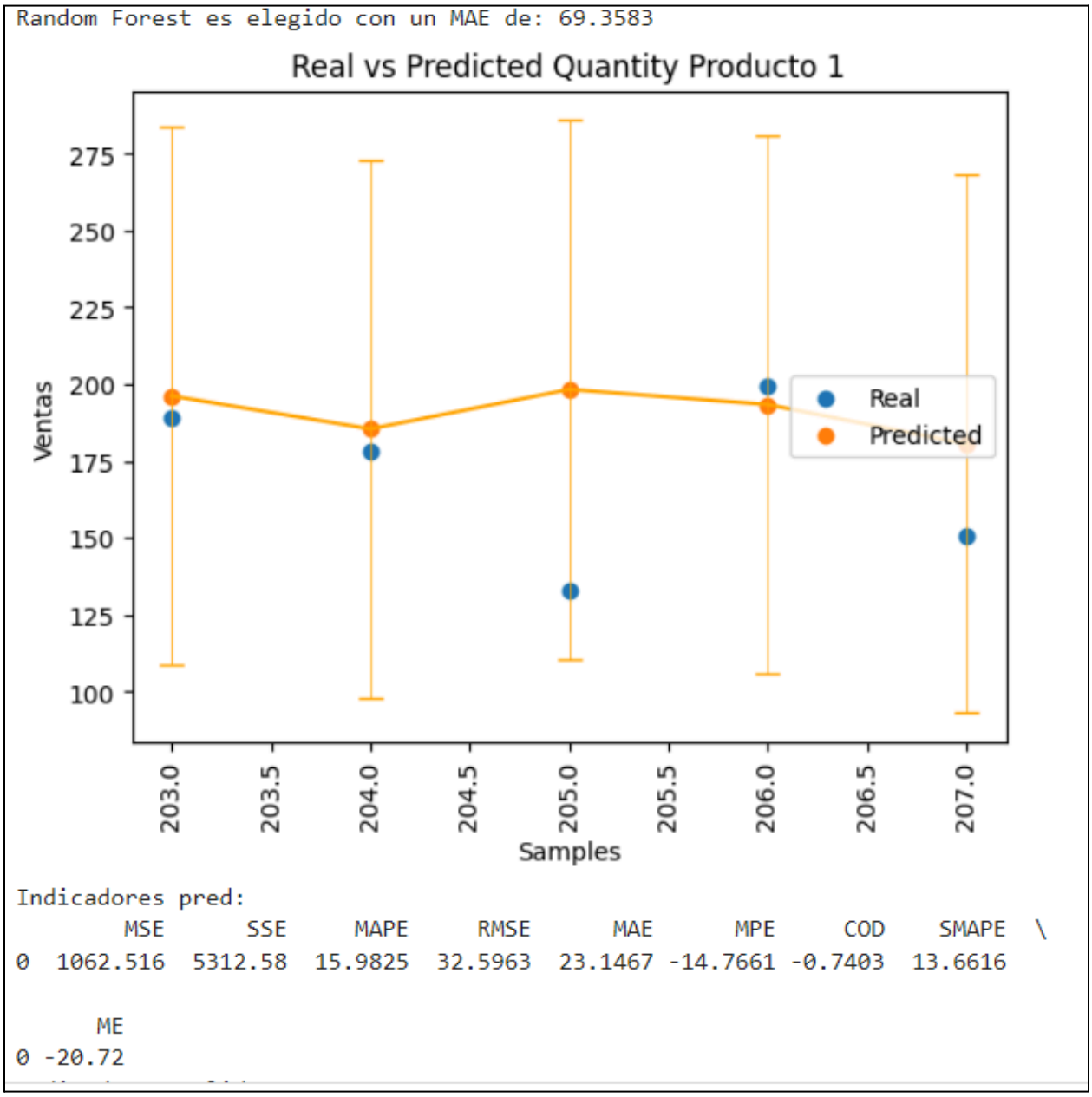
Estos datos muestran un muy buen modelo, que es mejor que el generado por el método 1, sin embargo solo se compararon tres modelos con parámetros fijos, probados con solo uno de los productos. Es por eso que creamos el método #3.

Método #3: Time series split con validación de parámetros:

En este método se dividen inicialmente los datos en dos grupos, los anteriores a octubre de 2023 y los posteriores. Con el primer grupo se aplica la función `GridSearchCV` de scikit learn para realizar una selección de los mejores parámetros posibles para cada uno de los modelos: DT, RF y XGBoost. Después de esto se divide este grupo de datos en Training y Validation con el mismo criterio del método anterior. Training es usado para entrenar cada uno de los modelos más óptimos y se obtienen estadísticas con la validation data. Después, con estos datos se elige el

modelo con el menor MAE, y se modelan las predicciones del último mes con el modelo con menor MAE.

Ejemplo Modelo 1:



Después de ajustar los tres modelos, se obtiene el mejor que es Random Forest y se modela el último mes con un MAPE de 16% y un MAE de 23.15.

8. Evaluacion

Look Ahead Bias

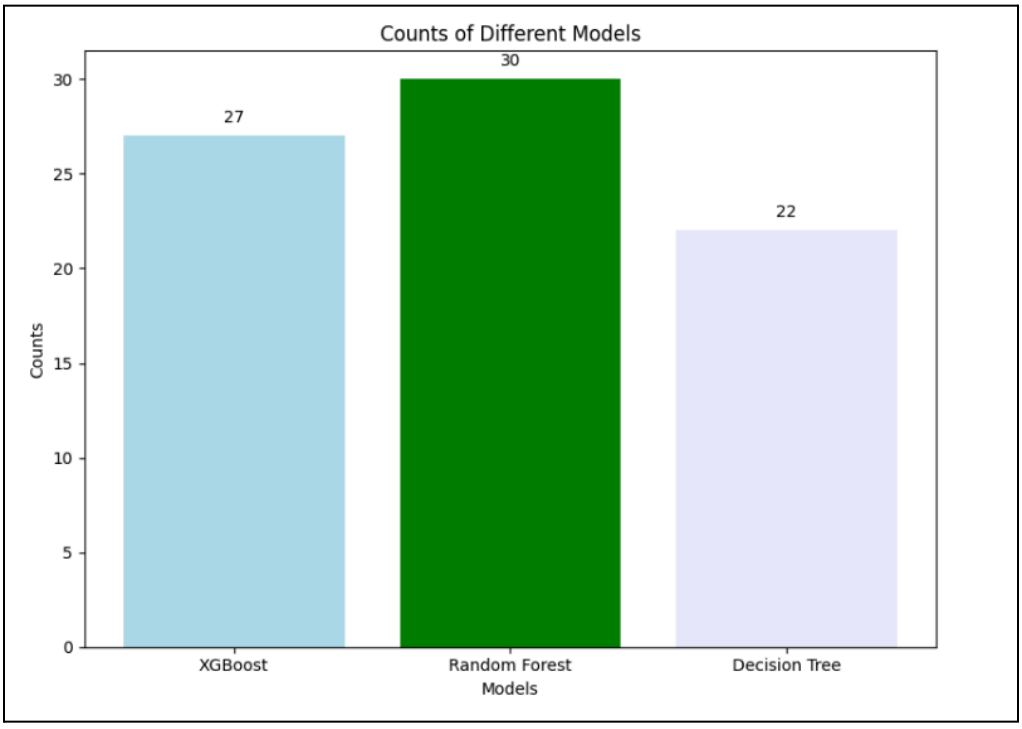
"One very general and important concern during data preparation is to beware of "leaks" (Kaufman et al. 2012). A leak is a situation where a variable collected in historical data gives information on the target variable—information that appears in historical data but is not actually available when the decision has to be made."

- Provost & Faucet

Dentro de la creación de modelos de predicción existe una problemática muy común pero muy alarmante. Los modelos a veces pueden llegar a predecir a la perfección los valores reales pero porque el manejo de las variables que se le entregan para entrenar tienen información que lleva de manera directa a la respuesta correcta, un ejemplo aplicado a nuestro reto es el total de venta. A nuestro modelo no le dimos acceso al total de venta ya que con esta información el modelo podría haber deducido de manera exacta cuántas piezas se vendieron y por lo tanto predecir a la perfección este valor. A esto se le conoce como data leakage.

Una vez se tomó en cuenta esta situación y se verificó que ninguno de los modelos tuviera esta problemática, hicimos nuestra selección de modelos.

La selección de Modelos fue:



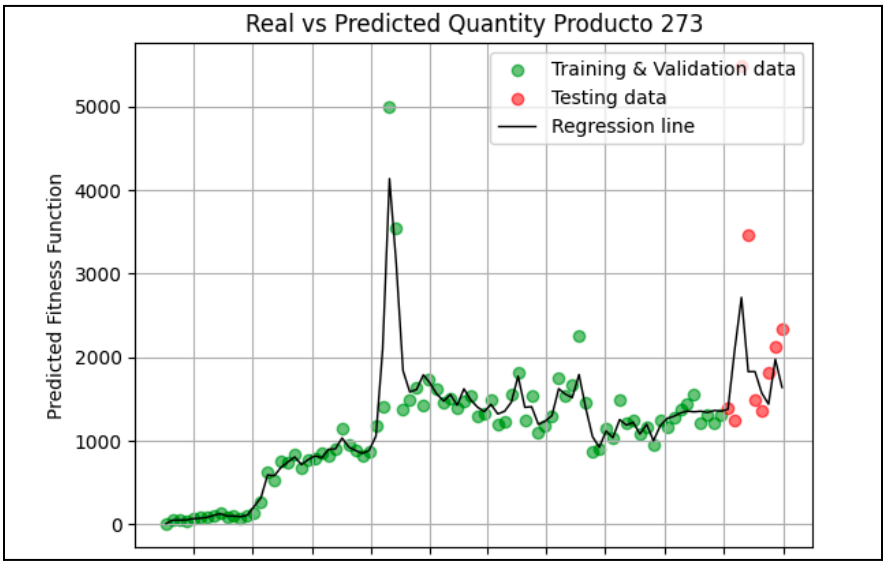
Esta gráfica representa la cantidad de producto los cuales se ajustan mejor a cada tipo de modelo, 27 productos para XGBoost, 30 productos para Random Forest y 22 productos para Decision Tree.

Obtenemos estadísticos sobre las pruebas:

Modelo	PRODUCTO	MEDIA	MAE	MAPE	% (1-MAPE)
RF	273	1,213	359 [29%]	19.48	80.52 %
RF	1	177.2	23.14 [13%]	15.98	84.02%
DT	5	106.26	14.78 [14%]	10.8	89.2%
DT	8	63.23	31.63 [50%]	35	65%

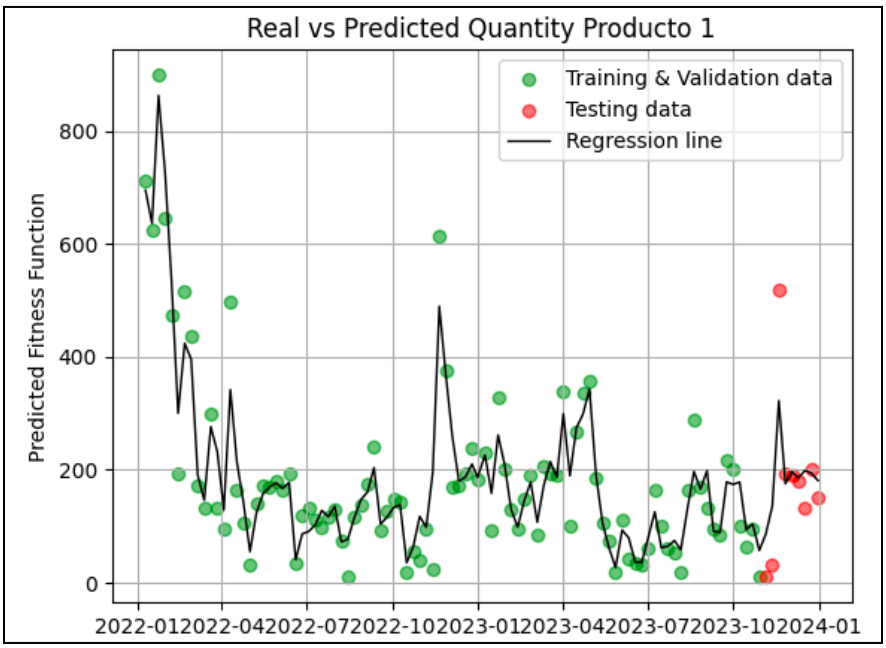
Se presentan los datos sobre los 4 productos más vendidos:273,1,5 y 8:

Producto 273:



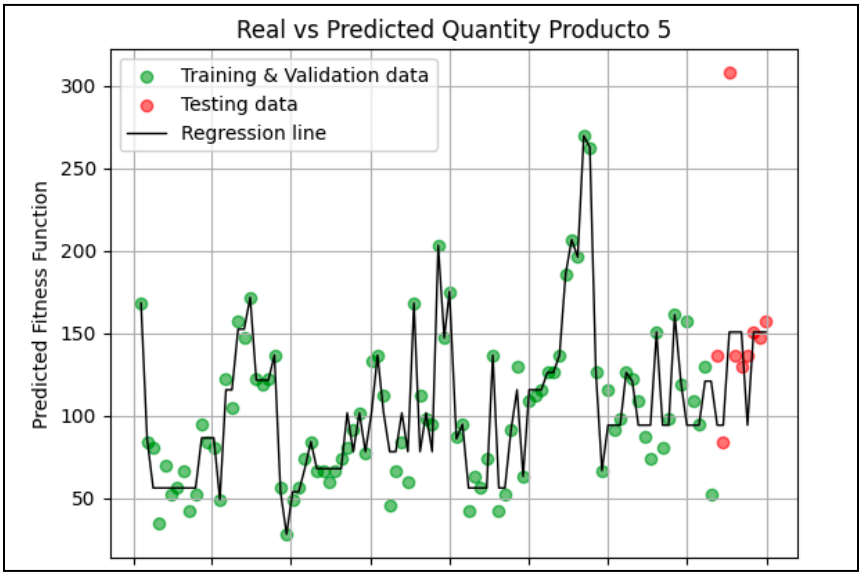
Se tiene un MAPE muy bueno de 19.48 con un error medio absoluto de 359. Este MAE, a pesar de parecer alto, solo representa el 29% de la media. La gráfica muestra un muy buen ajuste para los datos de entrenamiento y validación y un aumento de la diferencia para el testing, sin embargo los estadísticos nos dicen que es un buen modelo.

Producto 1:



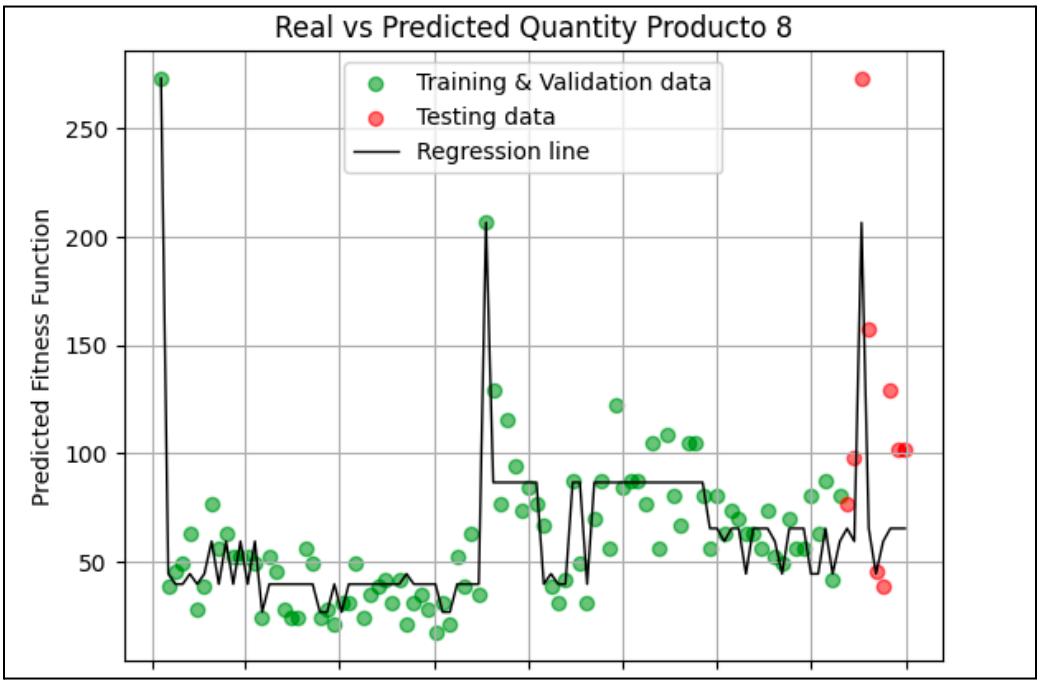
Se tiene un MAPE muy bueno de 15.98 con un error medio absoluto de 23.14. Este MAE es excelente, considerando que se venden en promedio más de 150 unidades y en ocasiones más de quinientas. La gráfica muestra un muy buen ajuste para los datos de entrenamiento y validación y un aumento de la diferencia para el testing, no logrando capturar el valor del buen fin, sin embargo los estadísticos nos dicen que es un buen modelo.

Producto 5:



Se tiene un MAPE mejor que para cualquier producto de 10.8 con un error medio absoluto de 14.78. Este MAE es excelente, considerando que se venden en promedio más de 100 unidades y en ocasiones más de doscientas. La gráfica muestra un muy buen ajuste para los datos de entrenamiento y validación y un aumento de la diferencia para el testing, no logrando capturar el valor del buen fin, sin embargo los estadísticos nos dicen que es un buen modelo.

Producto 8:



Para el producto 8 se presenta el MAPE más alto con 35. Tiene un MAE de 31.63 que representa el 50% de la media, sin embargo se vio un aumento de la media en el último año a casi 100. A pesar de presentar altos valores en los estadìsticos, este modelo sì logró capturar la subida de ventas en el buen fin, lo que puede sugerir que existe una mejor captura de tendencias pero sin precisión en la escala.

9. Conclusión

Para esta problemática se hicieron múltiples modelos distintos, y en base a los resultados de cada uno, pudimos concluir en base a los modelos, que los 3 mejores que se acoplan muy bien a la mayoría de los productos son, Decision Tree, Random Forest y XGBoost. También cabe resaltar que cada modelo tiene mejores o peores resultados dependiendo de el producto al que se le aplique. Nuestros resultados plasmados aquí son simplemente un acercamiento para el problema y no representan una solución definitiva, para esto debe entrar la interacción humana y la toma de decisiones empresariales, lo que nosotros ofrecemos es simplemente una herramienta de predicción. Para poder desplegar una solución definitiva, se requeriría trabajar más cerca con el socio formador para poder predecir sus necesidades y resultados meta.

La predicción de demanda de SKUs representa un problema que va más allá de los datos numéricos. Las decisiones empresariales y la intuición del negocio representan una parte importante de las tendencias de demanda para cada producto, con experiencia que se necesitan más de dos años de datos para modelar. Nuestra elección de método de modelaje dio como resultado una predicción adecuada. A pesar de no llegar al 15 de MAPE establecido por el socio formador., logramos obtener modelos que generan una buena idea sobre las posibles tendencias que ayuden a reducir el impacto de una sobreproducción o subproducción.

Podemos observar los porcentajes (MAPES) más altos con el modelo de Decision Tree, pero es importante notar que esto es cuando producimos los productos con más entradas. Para poder llegar a una conclusión *real* de cuál es el mejor modelo, sería necesario reducir las limitaciones actuales del modelo (ex: la falta de datos). Como dice Provost “While the probabilities can be estimated from data, the costs and benefits often cannot.They generally depend on external information provided via analysis of the consequences of decisions in the context of the specific business problem. Indeed, specifying the costs and benefits may take a great deal of time and thought. (199).

Este proyecto nos presentó una oportunidad para aplicar lo visto en clase de una manera útil en el mundo real. La cita “Academic programs in statistics, machine learning, and data mining often present students with problems ready for the application of the tools that the programs teach (Provost 277) demuestra esto apartamente.

10. Referencias

Orad, A. (2020, February 14). *Council Post: Why Every Company Is A Data Company*. Forbes. https://www.forbes.com/sites/forbestechcouncil/2020/02/14/why-every-company-is-a-data-company/?sh=74e6f93817a4

Moran, M. (2015). *Consumo y producción sostenibles - Desarrollo Sostenible*. Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/

IBM. (2021, August 17). *CRISP-DM Help Overview*. Www.ibm.com. https://www.ibm.com/docs/en/spss-modeler/SaaS?topic=dm-crisp-help-overview

¿Quiénes Somos? | Tienda de colchones Don Colchón. (2024). Don Colchon. Retrieved March 8, 2024, from https://doncolchon.com.mx/pages/quienes-somos

Encuentra tu sucursal. (2024.). Don Colchon. Retrieved March 8, 2024, from https://doncolchon.com.mx/pages/encuentra-tu-sucursal

Barkho, G. (2023, July 31). *"Startups were no more than marketing firms": The mattress industry is starting to stabilize following years of heavy M&A activity*. Modern Retail. https://www.modernretail.co/operations/startups-were-no-more-than-marketing-firms-the-mattress-industry-is-starting-to-stabilize-following-years-of-heavy-ma-activity/

Echeverría, M. (2023, November 28). *El gigante alemán que va por el lucrativo negocio del descanso en México*. Expansión. https://expansion.mx/empresas/2023/11/28/marca-colchones-emma-mexico

Provost, F., & Fawcett, T. (2013c). *Data science for business : what you need to know about data mining and data-analytic thinking*. O'Reilly Media.

Allwright, S. (2021, October 27). *What is a good MAPE score and how do I calculate it?* Stephen Allwright. https://stephenallwright.com/good-mape-score/

Long, A. (2020, February 2). *Machine Learning with Datetime Feature Engineering: Predicting Healthcare Appointment No-Shows*. Medium. https://towardsdatascience.com/machine-learning-with-datetime-feature-engineering-predicting-healthcare-appointment-no-shows-5e4ca3a85f96

Wickham, H. (2014). Tidy Data. *Journal of Statistical Software*, *59*(10). https://doi.org/10.18637/jss.v059.i10

Gomede, E. (2023, August 28). *The Significance of Train-Validation-Test Split in Machine Learning*. Medium. https://medium.com/@evertongomede/the-significance-of-train-validation-test-split-in-machine-learning-91ee9f5b98f3

Santra, R. (2023, July 18). *Introduction to SARIMA Model*. Medium. https://medium.com/@ritusantra/introduction-to-sarima-model-cbb885ceabe8

ARIMA, SARIMA, and SARIMAX Explained. (n.d.). Zero to Mastery. Retrieved March 11, 2024, from https://zerotomastery.io/blog/arima-sarima-sarimax-explained/

Provost, Foster, Fawcett, Tom. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking* . Sebastopol, California: O\'Reilly.