

END-TO-END MACHINE LEARNING

Master thesis: DSMarket - your next generation store ([data_dsmarket](#))



Laura López Garrido | Marina Ramiro Pareta | Sofia Schweiger

ÍNDICE

0. Preprocesamiento de los datos originales
1. Análisis
2. Agrupación (Clustering)
3. Pronóstico de ventas
4. Caso de Uso: Reposición de Tiendas
5. PowerBI

0. Preprocesamiento de los datos originales

Notebook: [0_1_aux_db.ipynb](#)

- Archivos origen:
 - **item_sales.csv** (Carpeta: **originals**)
 - **daily_calendar_with_events.csv** (Carpeta: **originals**)
 - **item_prices.csv** (Carpeta: **originals**)
- Archivos resultantes:
 - **item_sales_mod.csv** (Carpeta: **new_files**): ventas de artículos por día (“d”).
 - 58M de registros (58,327,370)
 - 9 columnas: “id”, “item”, “category”, “department”, “store”, “store_code”, “region”, “d”, “units”
 - **daily_calendar_with_events_mod.csv** (Carpeta: **new_files**)
 - 1913 registros (total de días)
 - 8 columnas: “date”, “weekday”, “weekday_int”, “d”, “event”, “week”, “year”, “year_week”
 - **item_prices_mod.csv** (Carpeta: **new_files**): precios de los artículos con su correspondiente “year-week”.
 - ~7M de registros (6,965,706)
 - 5 columnas: “id”, “sell_price”, “year_week”, “year”, “week”
 - **sales_calendar.csv** (Carpeta: **new_files**):
 - merge: **item_sales_mod.csv**, **daily_calendar_with_events_mod.csv**
 - 58M de registros
 - 16 columnas: “id”, “item”, “category”, “department”, “store”, “store_code”, “region”, “d”, “units”, “date”, “weekday”, “weekday_int”, “event”, “week”, “year”, “year_week”
 - **sales_calendar_prices_full.csv** (Carpeta: **new_files**):
 - merge: **item_sales_mod.csv**, **daily_calendar_with_events_mod.csv**, **item_prices_mod.csv**
 - 58M de registros
 - 17 columnas: “id”, “item”, “category”, “department”, “store”, “store_code”, “region”, “d”, “units”, “date”, “weekday”, “weekday_int”, “event”, “week”, “year”, “year_week”, “sell_price”.

Notebook: [0_2_merged_db.ipynb](#)

- Archivos origen:
 - **unit_id.csv** (Carpeta: **new_files**)
 - **item_sales.csv**
 - **date_id.csv** (Carpeta: **new_files**)
 - **item_prices_mod.csv** (Carpeta: **new_files**)

- *Archivos resultantes:*
 - **sales_calendar_prices.csv** (Carpeta: **new_files**)
 - **gb_year.xlsx**: “**sales_calendar_prices.csv**” agregado por año, para simplificar los datos y que pueda ser más útil de cara al powerBI. (Carpeta: **task_1_powerbi**)

1. Análisis

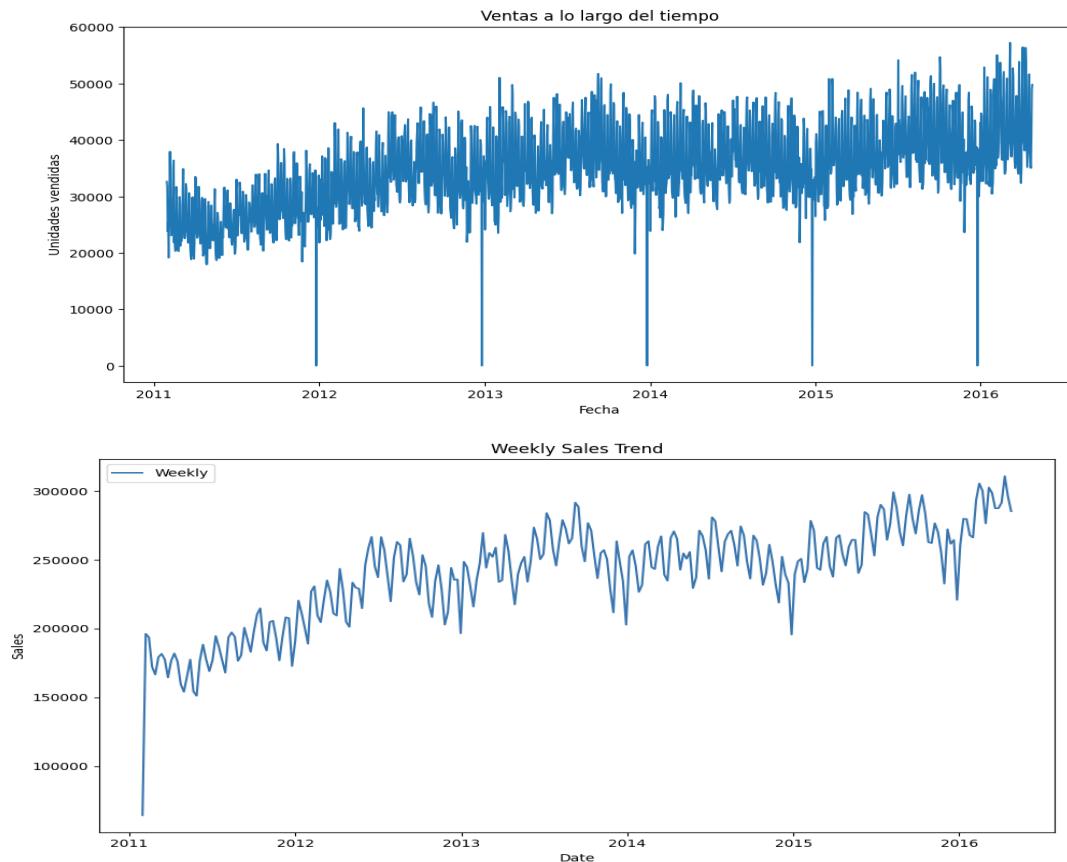
En esta tarea, Michelle, la Directora Digital, quiere que analicemos el estado actual de la empresa.

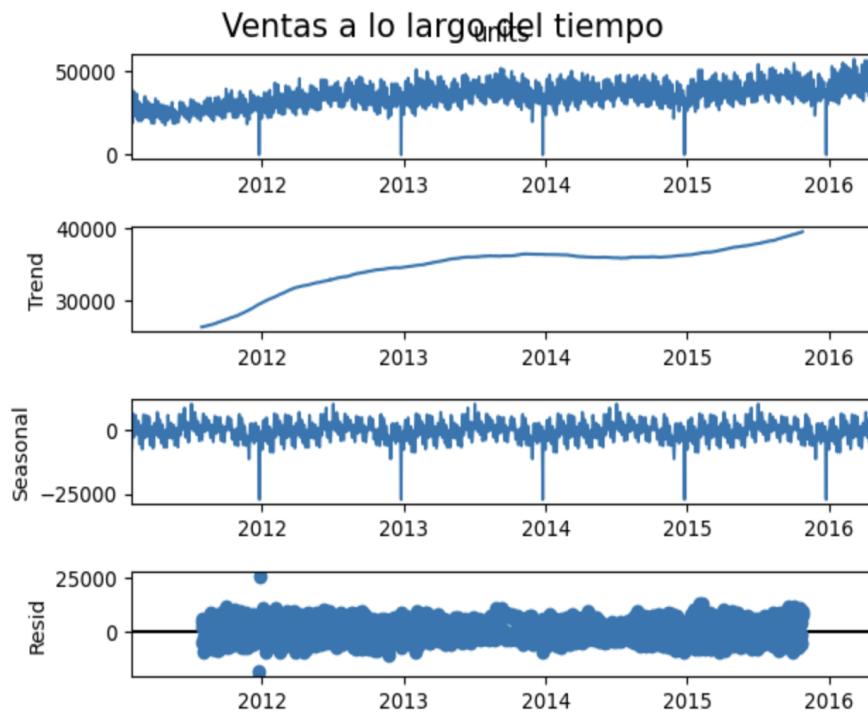
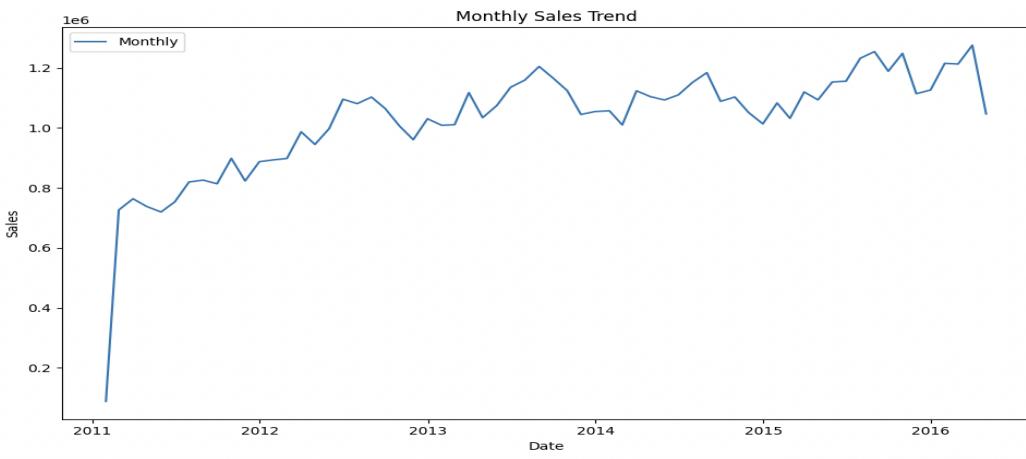
Carpeta: task_1_powerbi

1.1. Analizar las tendencias de ventas a lo largo del tiempo, identificando cualquier patrón estacional o tendencia general.

Notebook: 1_1_edo.ipynb

En primer lugar, realizamos un análisis detallado de las ventas globales de la compañía. Al examinar los datos en términos temporales, pudimos identificar una tendencia creciente tanto en el total de ventas como en el total de ganancias anuales.





Analizamos, además, la distribución de las ventas y ganancias por meses, y encontramos un crecimiento significativo durante ciertos meses específicos. Los meses con mayor número de ventas se encuentran principalmente entre febrero y abril. Esto coincide con la ocurrencia de eventos importantes como (San Valentín, Carnaval), Pascua, Ramadán y la Super Bowl. Estos eventos parecen influir positivamente en las ventas, lo que nos brinda información valiosa para planificar estrategias comerciales en fechas clave.

En los meses de verano también se registran ventas altas, aunque no sean los meses con mayores ventas.

Cabe destacar que, paradójicamente, durante los meses de Navidad no se observa un aumento significativo de las ventas y las ganancias, a pesar de ser un periodo de alto

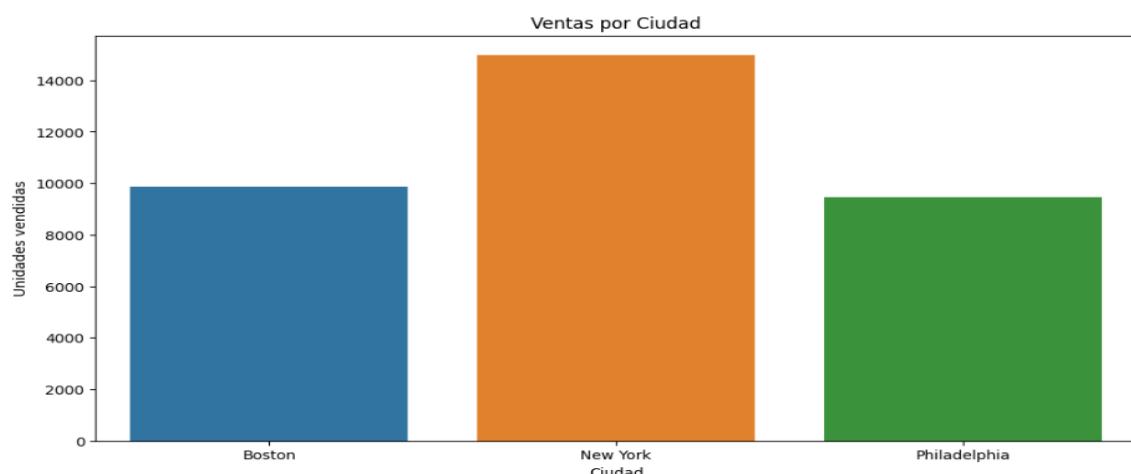
consumo en general. Este hallazgo nos brinda una valiosa oportunidad para investigar más a fondo los factores que podrían estar influyendo en este comportamiento.

Por otro lado, al analizar las ventas en función de los días de la semana, hemos encontrado que los días de mayor venta ocurren los fines de semana, lo cual es lógico debido a que muchas personas disponen de más tiempo libre para realizar compras. Aunque hay algunas pequeñas variaciones en otros días de la semana, esta tendencia general nos proporciona información valiosa para planificar la distribución de recursos y personal en nuestras tiendas. Estos hallazgos nos permitirán optimizar nuestras operaciones, especialmente en períodos de alta demanda, y evaluar cómo mejorar el rendimiento durante las épocas navideñas.

1.2. Comparar las ventas entre las ciudades para determinar si hay diferencias significativas.

Al evaluar las ventas globales por ciudad, notamos que la ciudad de Nueva York, con sus 4 tiendas, lidera las ventas. Le siguen Philadelphia y Boston, con ventas globales muy similares y 3 tiendas en cada ciudad. Estos resultados nos ofrecen una visión más clara sobre el rendimiento de nuestras tiendas en diferentes áreas geográficas, lo que puede ser de gran utilidad para futuras expansiones, apertura de nuevas tiendas y estrategias de marketing focalizadas.

En resumen, el análisis de productos nos ha proporcionado información valiosa sobre los artículos más vendidos a nivel global por región, tienda y departamento. También nos ha permitido comprender las preferencias de compra en diferentes ciudades. Estos hallazgos son fundamentales para enfocar nuestras operaciones y estrategias de manera más efectiva, optimizar nuestro inventario y brindar a nuestros clientes la mejor experiencia de compra posible.



1.3. Examinar la popularidad de los productos e identificar los más vendidos en cada ciudad.

Los productos más vendidos se comparten de manera importante entre las 3 ciudades de nuestros datos.

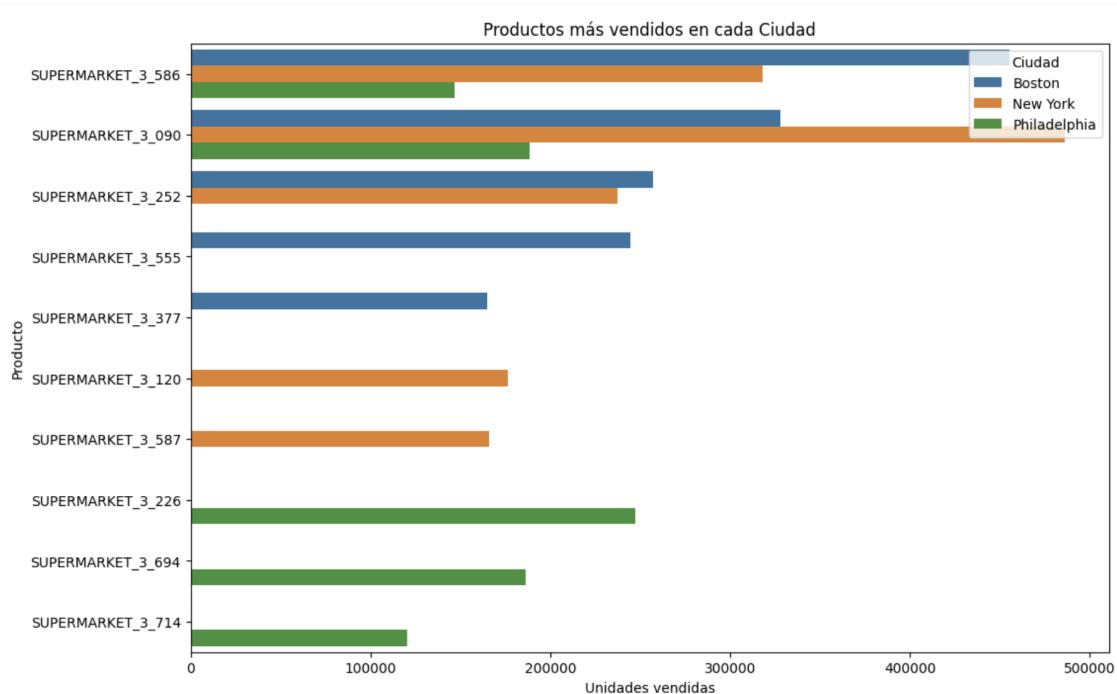
En cuanto al análisis regional de productos, hemos identificado los artículos más vendidos a nivel global (por región) y en cada tienda.

Esto demuestra la alta demanda y preferencia de nuestros clientes por estos productos en todas nuestras tiendas, lo que nos brinda una valiosa oportunidad para maximizar su disponibilidad en nuestro inventario.

Además, hemos realizado un análisis por departamentos y hemos encontrado que el departamento con más ventas es SUPERMARKET_3, con un total de 32 millones de ventas, seguido por HOME_&_GARDEN_1 con 12 millones de ventas.

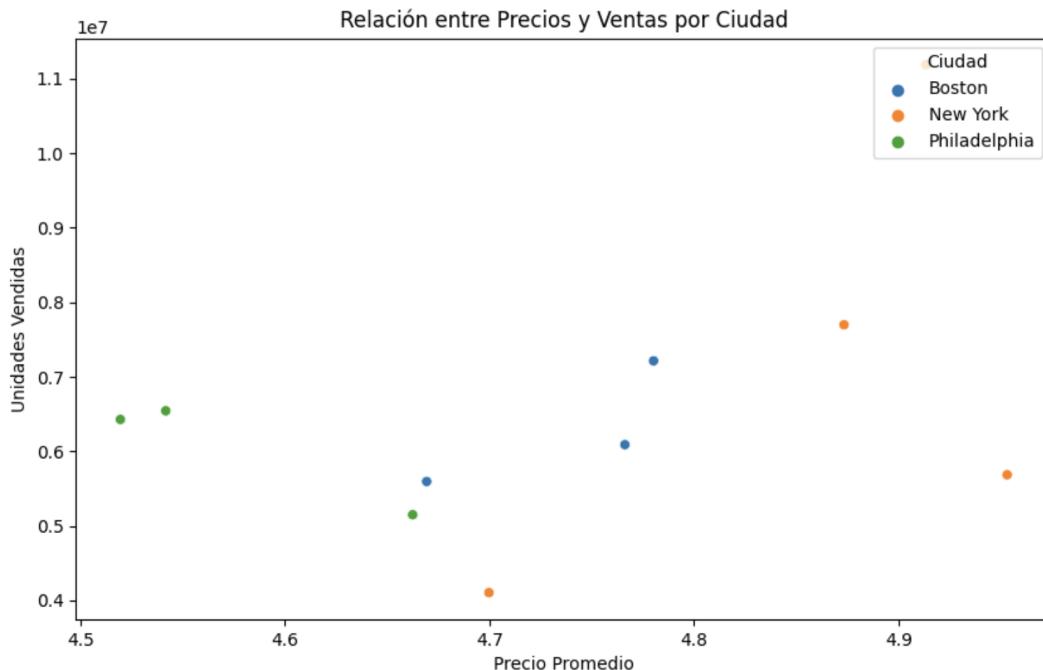
Esta información nos ayuda a entender qué categorías de productos tienen un mayor impacto en nuestras ventas y nos permite enfocar nuestros esfuerzos en aquellas áreas que son más exitosas.

Al analizar las ventas por tienda de manera más detallada, hemos observado que los departamentos con más ventas son consistentes en la mayoría de nuestras tiendas, con la única excepción de la tienda de Brooklyn, donde el segundo departamento con más ventas es ACCESORIES_1. Esta variación en las preferencias de compra puede deberse a diferentes factores y nos brinda una oportunidad para adaptar nuestra estrategia a las preferencias locales de nuestros clientes.



1.4. Analizar las variaciones de precios entre las tiendas y ciudades, y determinar si hay alguna relación entre los precios y las ventas.

En Philadelphia destan 2 tiendas con el precio medio más bajo y una cifra alta de unidades vendidas. En New York, por su parte, lo que resalta es que habitualmente el precio sale más elevado que en las otras ciudades.



1.5. Preparar un informe completo que incluya visualizaciones, tablas y conclusiones sobre los patrones y tendencias encontrados en los datos.

Notebook: [1_2_powerBI.ipynb](#)

- Archivos resultantes:
 - **full_df_gb_date.csv** (Carpeta: **task_1_powerbi**): aquí están todos los datos; se relacionan directamente en el powerBi mediante una relación con la columna “date”.
 - **gb_year.csv** (Carpeta: **task_1_powerbi**): lo mismo que “**full_df_gb_date.csv**” pero agrupado por años. Se pierde la resolución de día/semana/mes, pero pesa menos.
 - **new_calendar.csv** (Carpeta: **task_1_powerbi**)





2. Agrupación (Clustering)

Joelle, la Gerente de Marketing, sugiere identificar grupos de productos que se comporten de manera similar.

Carpetas: task_2_clustering

En la actualidad, nuestra tienda cuenta con un inventario de **3049 productos**, cada uno con sus propias características y patrones de consumo. Este volumen y diversidad presentan desafíos a la hora de tomar decisiones informadas y precisas sobre cuestiones como la disposición de la tienda, las promociones, el inventario, entre otros.

La clusterización de estos productos, es decir, su agrupación en base a ciertas similitudes nos permitirá optimizar nuestras estrategias de marketing y ventas, mejorar la gestión de inventario, e incluso personalizar la experiencia de compra de nuestros clientes.

2.1. Preprocesar los datos y generar variables útiles para la segregación de los productos en base a la información proporcionada.

Notebook: 2_1_general_preprocessing_year_week.ipynb

- Objetivo:

Hacer el **preprocesamiento de los datos agrupados por semana** y generar variables que puedan ser útiles para la clusterización de los productos.

- Archivos origen:

- sales_calendar_prices.csv (Carpeta: new_files)
- daily_calendar_with_events_mod.csv (Carpeta: new_files)

- Preprocessing:
 - Tenemos “date” como tipo object, habrá que pasarlo a fecha.
 - Podemos pasar “year” y “week” a integers, y luego también a un formato de fecha.
 - Podemos generar también una columna de “month” y de “day”, para acabar de tener todos los datos.
 - No hay duplicados.
 - Hay valores nulos en las columnas “sell_price”, “year” y “week”, ya que se nos proporcionaban los precios por semana (“year-week”). “year” y “week” son fáciles de sacar ya que tenemos la columna de date. Para imputar los nulos de “sell_price” hay dos estrategias:
 - Eliminar directamente las filas que contengan NaNs. Son aproximadamente el 36% de los datos, es demasiado.
 - Imputar los valores NaN en la columna 'sell_price' con los valores más recientes de la misma columna**, teniendo en cuenta el mismo ‘id’.
 Vamos a usar el método `bfill()`/`ffill()` junto con `groupby()` para realizar el relleno hacia atrás/delante por grupos. Primero vamos a asegurarnos que los datos estén ordenados antes de realizar el back/forward fill.
- Generación de variables:
 - **revenue**: units * sell_price
 - **event_SuperBowl**: hemos extendido las ocurrencias del evento a 7 días antes.event_SuperBowl_sales: ventas durante la ocurrencia de la Superbowl.
 - **event_Thanksgiving**: hemos extendido las ocurrencias del evento a 7 días antes.event_Thanksgiving_sales: ventas durante la ocurrencia de Acción de Gracias.
 - **event_NewYear**: hemos extendido las ocurrencias del evento a 7 días antes.event_NewYear_sales: ventas durante la ocurrencia de Año Nuevo.
 - **event_Easter**: hemos extendido las ocurrencias del evento a 7 días antes. event_Easter_sales: ventas durante la ocurrencia de Pascua.
 - **event_Ramadan**: hemos extendido 30 días posteriores al inicio del Ramadán starts”. event_Ramadan_sales: ventas durante la ocurrencia de Ramadán.
 - **event_any**: incluye todos los eventos en versión extendida. **event_sales**: incluye todas las ventas en cualquiera de los eventos.
 - **christmas_sales**: ventas durante los meses de Noviembre y Diciembre.
 - **summer_sales**: ventas durante los meses de Junio, Julio y Agosto.
 - **sales_2011** no está completo.
 - **sales_2012**
 - **sales_2013**
 - **sales_2014**
 - **sales_2015**
 - **sales_2016**: no está completo, acaba el 2016-04-18.

- **sales_2015_rel**: ventas de 2015 relativas al periodo de 2016, acaba el 2015-04-18.
- Archivos resultantes:
 - **daily_calendar_with_events_mod_extended.csv** (Carpeta: **new_files**)
 - **df_forecast_1.xlsx** (Carpeta: **new_files**)

Notebook: 2_2_general_preprocessing_days.ipynb

- Objetivo:
Hacer el **preprocesamiento de los datos agrupados por día** y generar variables que puedan ser útiles para la clusterización de los productos.
- Archivos origen:
 - **item_sales_mod.csv** (Carpeta: **new_files**)
 - **daily_calendar_with_events_mod.csv** (Carpeta: **new_files**)
- Generación de variables:
 - **weekend_sales**: ventas totales del fin de semana (Sábado y Domingo).
 - **week_sales**: ventas totales durante los workdays de la semana (Lunes-Viernes).
 - **item_first_sale**: fecha de la primera compra de cada artículo.
 - **item_last_sale**: fecha de la última compra de cada artículo.
 - **item_exhibition_days**: tiempo de exposición para cada artículo.
 - **days_with_purchases**: días de compra de los artículos al año.
 - **purchase_to_exhib_ratio**: ratio de días de compras vs. tiempo de exposición, una medida de la eficiencia de los productos.
- Archivos resultantes:
 - **df_cluster_item_extra.csv** (Carpeta: **new_files**)

Notebook: 2_3_item_cluster_preparation.ipynb

- Objetivo:
Unificar los archivos anteriores, con las variables generadas a nivel de semana y de día, y acabar de generar las variables de interés para la clusterización de productos.
- Archivos origen:
 - **df_forecast_1.xlsx** (Carpeta: **new_files**)
 - **df_cluster_item_extra.csv** (Carpeta: **new_files**)
- Generación de variables:
 - **initial_item_price**: precio inicial para cada artículo. Tomamos el precio máximo de los artículos como referencia.
 - **first_discount**: primer descuento (>5%) de los productos.
 - **sell_price_first_discount**: precio de los artículos en el primer descuento (>5%).
 - **days_until_first_discount**: días hasta el primer descuento.

- **discount_on_date_days**: días en el que los artículos se encuentran rebajados (>5%). **discount_on_date_sales**: ventas de los artículos en días de descuento.
 - ***sales_%**: transformación de las variables de ventas en porcentaje respecto al número total de unidades vendidas (**week_sales_%**, **weekend_sales_%**, **summer_sales_%**, **christmas_sales_%**, **event_sales_%**, **event_SuperBowl_sales_%**, **event_Thanksgiving_sales_%**, **event_Ramadan_sales_%**, **event_NewYear_sales_%**, **event_Easter_sales_%**, **discount_on_date_sales_%**).
 - **mean_pvp**: revenue / units.
 - **units_per_purchase**: units / days_with_purchases.
 - **revenue_per_purchase**: revenue / days_with_purchases.
 - **full_price_prediction_temp**: initial_sell_price * units
 - **mean_discount**: (full_price_prediction_temp - revenue) / full_price_prediction_temp.
 - **2016_vs_2015**: tendencia de ventas totales de 2016 respecto al año anterior. En este caso no se ha usado todo el periodo de 2015, ya que la fecha máxima para el dataset de 2016 es 2016-04-18; así que se ha usado "sales_2015_rel" para generar la tendencia.
 - **2015_vs_2014**: tendencia de ventas totales de 2015 respecto al año anterior.
 - **2014_vs_2013**: tendencia de ventas totales de 2014 respecto al año anterior.
 - **% de ventas en las diferentes tiendas** (**BOS_1_%**, **BOS_2_%**, **BOS_3_%**, **NYC_1_%**, **NYC_2_%**, **NYC_3_%**, **NYC_4_%**, **PHI_1_%**, **PHI_2_%**, **PHI_3_%**).
- Archivos resultantes:
 - **df_cluster_item_final.csv** (Carpeta: **task_2_clustering**)
 - **df_cluster_item_final_ext.csv** (Carpeta: **task_2_clustering**): con los % de ventas en las diferentes tiendas.

2.2. Utilizar técnicas de agrupación, como el algoritmo de k-means, para agrupar los productos según su comportamiento.

Llegamos a este punto tenemos todo listo para construir nuestro pipeline.

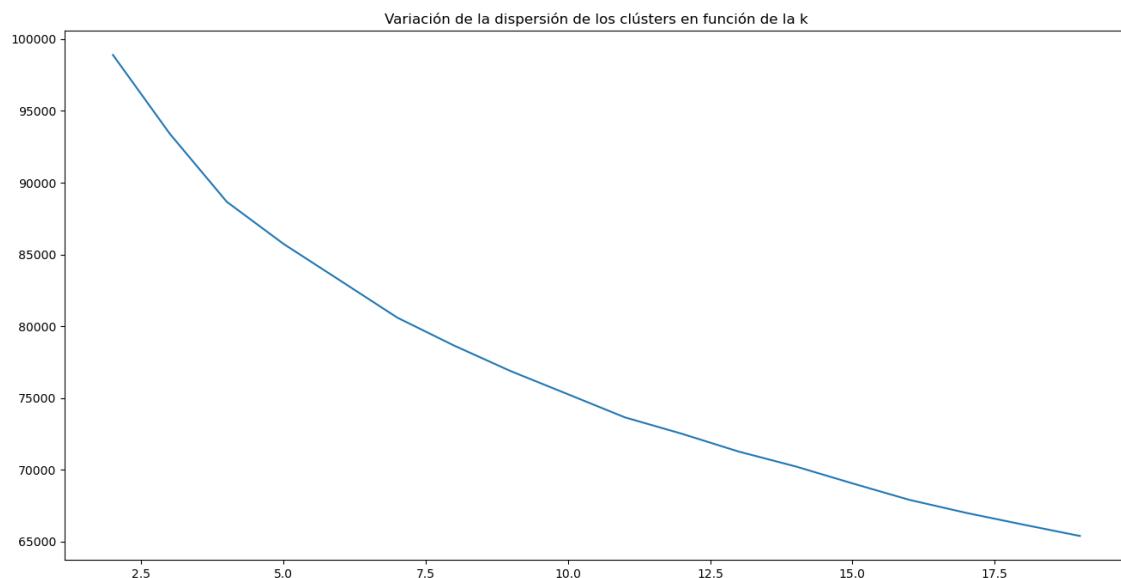
Notebook: [2_4_item_clustering_4_clusters.ipynb](#)

- Archivos origen:
 - **df_item_cluster_ready.xlsx** (Carpeta: **task_2_clustering**)
- Selección de variables:
 - **revenue_per_purchase**
 - **units_per_purchase**
 - **purchase_to_exhib_ratio**
 - **mean_pvp**

- **mean_discount**
 - **2016_vs_2015**
 - **2015_vs_2014**
 - **2014_vs_2013**
- [Escalar y ajustar los valores:](#)
 - **KNNImputer()**: Este paso se utiliza para rellenar los valores faltantes en el Data Frame utilizando el algoritmo de K-Nearest Neighbors (KNN). Reemplaza los valores faltantes con la media de los valores de los vecinos más cercanos.
 - **ArrayTypeDataFrame (columns = columns, index = index)**: Este paso convierte el numpy array (generado después de imputar los valores faltantes) de nuevo a un Data Frame, utilizando las columnas y el índice originales.
 - **OutlierFilter(q = 0.99, col_to_filter = selected_columns)**: Este paso filtra los valores atípicos (outliers) del DataFrame.
 - **StandardScaler()**: Este paso escala los valores del Data Frame para que tengan media cero y varianza unitaria.

- [Elbow curve:](#)

Hacemos un fit con el algoritmo de clustering “**k-means**” para calcular la inercia de los grupos (la dispersión de los datos al centroide), ajustando la segmentación de 2 a 19 clusters. Todo esto lo hacemos en un loop (técnica del **Elbow Curve**) porque queremos ver cuando hay un cambio brusco en la inercia, es decir, cuando aumentar el número de centroides no sale a cuenta porque la reducción del error es muy pequeña.



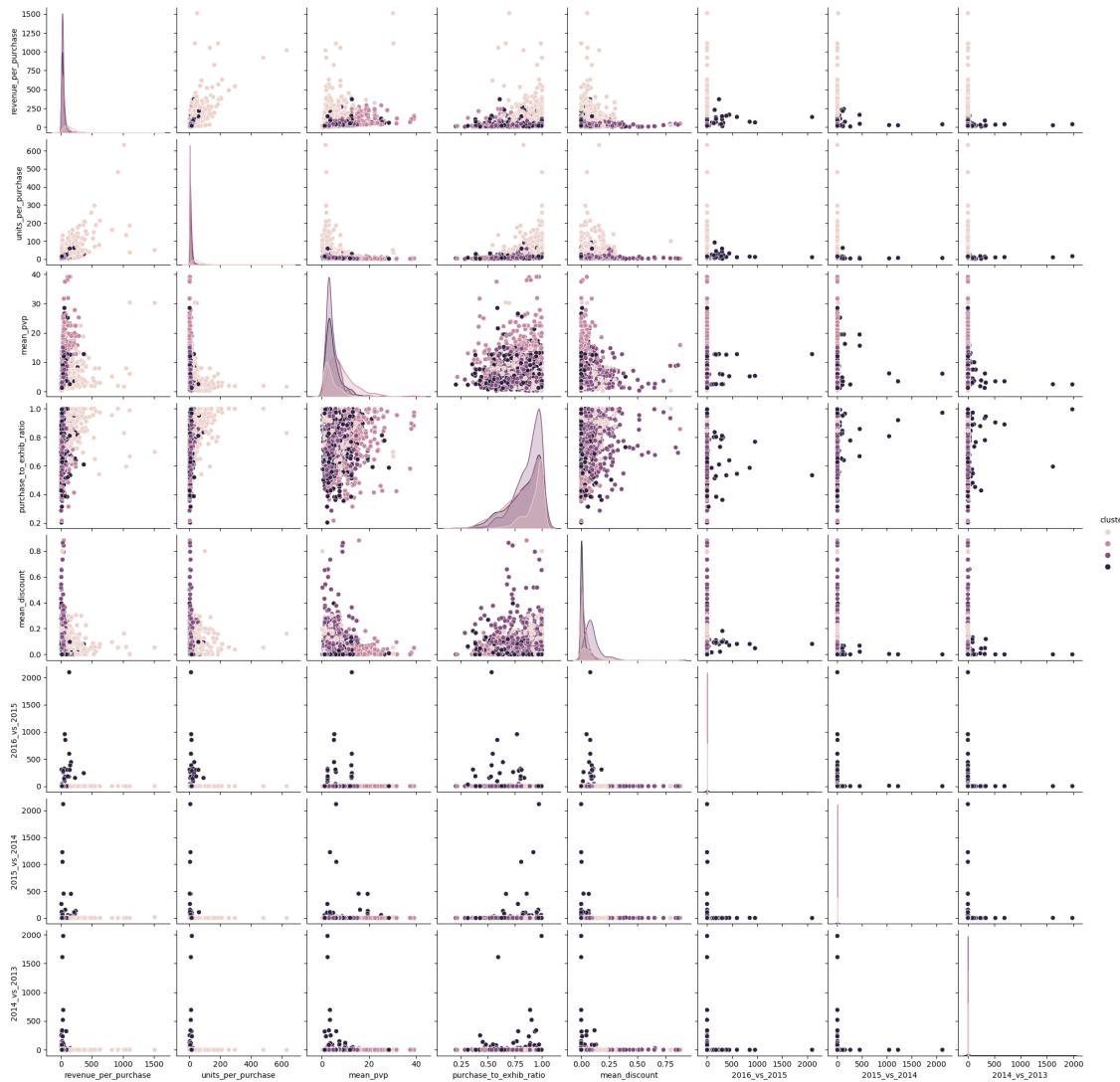
Observamos que el número óptimo de clusters para nuestros productos dadas las variables seleccionadas para su segmentación es **4**.

- [Segmentación de los productos con la “k” adecuada:](#)

Usamos el número óptimo de clusters, 4, en el **k-means** para predecir el clúster al que pertenece cada producto del Data Frame:

- Cluster 0: **361**
- Cluster 1: **819**
- Cluster 2: **1042**
- Cluster 3: **827**

Graficamos las relaciones entre las variables y la localización de cada cluster:



También generamos una ficha de productos con sus características para analizar mejor su segmentación ([item_clusters_4_report.csv](#)). Según el análisis de las métricas y características generales de cada cluster, podemos darles un nombre orientativo:

- **"Best sellers"** (Cluster 0): **361**
- **Unidades:** Es el grupo con la **mayor cantidad media de unidades por compra**.
- **Ingresos:** Este cluster muestra el **ingreso medio por compra más alto** entre los cuatro.
- **Precio:** Tiene un **precio medio por unidad más bajo** en comparación con otros clusters. Esto podría ser debido a que las compras en este cluster involucran una mayor cantidad de unidades.

- **Rendimiento de ventas:** Este grupo tiene la **tasa más alta de ventas por exhibición**, lo que indica un excelente rendimiento.
- **Descuento:** Los descuentos medios son altos en comparación con los otros grupos.
 - **"Premium"** (Cluster 1): **819**
- **Unidades:** La cantidad **media de unidades por compra es la más baja** entre todos los grupos, posiblemente debido a un precio por unidad más alto.
- **Ingresos:** Los **ingresos medios por compra son los segundos más altos**.
- **Precio:** Este cluster tiene el **precio medio por unidad más alto**.
- **Rendimiento de ventas:** El rendimiento de ventas es inferior al del cluster 0, pero sigue siendo sólido.
- **Descuento:** Tiene la **tasa de descuento media más baja**.
 - **"On Sale"** (Cluster 2): **1042**
- **Unidades:** La cantidad media de unidades por compra es la segunda más alta, pero sigue siendo muy inferior a la del Cluster 0 ("Best sellers").
- **Ingresos:** Este grupo muestra los ingresos medios por compra más bajos.
- **Precio:** Este grupo tiene un precio medio por unidad bastante bajo, lo que junto con la baja cantidad de unidades por compra podría explicar los bajos ingresos.
- **Rendimiento de ventas:** Aunque es inferior a los grupos 0 y 1, este cluster muestra un rendimiento de ventas decente.
- **Descuento:** Este cluster presenta la **mayor tasa de descuento media**.
 - **"Rising"** (Cluster 3): **827**
- **Unidades:** El número medio de unidades por compra es menor que en los clusters 0 y 2, pero mayor que en el cluster 1.
- **Ingresos:** Este grupo tiene ingresos medios por compra ligeramente superiores al cluster 2, pero menores que los clusters 0 y 1.
- **Precio:** Este grupo tiene un precio medio por unidad algo superior a los clusters 0 y 2, pero inferior al del cluster 1.
- **Rendimiento de ventas:** El rendimiento de ventas es el más bajo de todos los clusters, aunque sigue siendo decente.
- **Descuento:** La tasa de descuento media es la segunda más baja, son productos que normalmente no están rebajados.
- **Crecimiento:** Este cluster **destaca por su crecimiento extraordinariamente alto** en los últimos 3 años, de manera que podrían ser productos de nueva incorporación al mercado.
- **Archivos resultantes:**
 - **item_clusters_4.csv** (Carpeta: **task_2_clustering**): items con su correspondiente cluster.
 - **item_clusters_4_report.csv** (Carpeta: **task_2_clustering**): ficha de los productos.

2.3. Si es relevante para el análisis, considerar la agrupación de tiendas utilizando un enfoque similar al utilizado para los productos.

Mencionar que no tiene mucho sentido, ya que son pocas tiendas (10) y muchas características: maldición de la dimensionalidad.

2.4. Preparar una presentación de los resultados de la agrupación y sus implicaciones para evaluar el rendimiento de las campañas de marketing.



Hemos incorporado dos paneles interactivos en nuestro dashboard de powerBI para poder analizar las tendencias generales de ventas de cada cluster de productos, el comportamiento de estos clusters en las diferentes tiendas, y los artículos más y menos vendidos de cada grupo de productos.



3. Pronóstico de ventas

Paul, el Director Financiero, desea mejorar el pronóstico de ventas de la empresa.

Carpetas: task_3_sales_forecast

Otro de nuestros objetivos en DSMarket es abordar un desafío crucial en el mundo del comercio minorista: la predicción de ventas; aprovechando el poder de los datos y las técnicas de análisis avanzadas dejando atrás los métodos rudimentarios para prever nuestras ventas.

3.1. Desarrollar modelos predictivos para pronosticar las ventas/reposición de inventario.

Hemos escogido una **ventana de tiempo semanal**, agrupando nuestras ventas por “**year-week**”, ya que se los ha especificado que la reposición del inventario en tiendas se realiza de manera semanal.

Una vez entrenados los modelos, hemos empezado con **pronósticos a 4 semanas** (28 días), utilizando las 4 últimas semanas para evaluar su ajuste a las ventas reales:

- El dataset original incluye 274 semanas (1913 días).
- Se elimina la primera semana ya que no contiene toda la información, de manera que nos quedamos con **273 semanas completas**.
- A no ser que se especifique de otra manera, usamos las **269 primeras semanas para entrenar** los modelos, y **las últimas 4 semanas se usan para evaluar la calidad de los modelos** generados: ventas predichas vs. ventas reales.

Notebook: 3_1_item_prediction_total_units.ipynb

- Objetivo:

Primero de todo, hemos analizado de manera general nuestras ventas, utilizando la suma de ventas totales (todos los artículos) para sondear los distintos algoritmos de predicción de series temporales a nuestra disposición: ARIMA, Exponential Smoothing, Double Exponential Smoothing y Holt-Winters.

- Archivos origen:
 - **df_forecast_1.xlsx** (Carpeta: **new_files**)

Observamos que aquellos que nos ofrecen mejores resultados son **ARIMA** y **Holt-Winters**.

Notebook: 3_2_item_prediction_arima_product_store_train265_test4.ipynb

- Archivos de origen:
 - **df_forecast_1.xlsx** (Carpeta: **new_files**)
- Archivos resultantes:
 - **predictions_2023-07-22_arima_4weeks** (Carpeta: **task_3_sales_forecast**)
 - **combined_predictions_2023-07-22_arima_4weeks.csv** (Carpeta: **task_3_sales_forecast**)
 - **metrics_2023-07-22_arima_4weeks.csv** (Carpeta: **task_3_sales_forecast**)
 - **real_data_2023-07-22.csv** (Carpeta: **task_3_sales_forecast**)

Notebook: 3_3_item_prediction_hw_product_store_train265_test4.ipynb

- Archivos de origen:
 - **df_forecast_1.xlsx** (Carpeta: **new_files**)

- *Archivos resultantes:*
 - **`predictions_2023-07-22_hw_4weeks`** (*Carpeta: task_3_sales_forecast*)
 - **`combined_predictions_2023-07-22_hw_4weeks.csv`** (*Carpeta: task_3_sales_forecast*)
 - **`metrics_2023-07-22_hw_4weeks.csv`** (*Carpeta: task_3_sales_forecast*)
 - **`real_data_2023-07-22.csv`** (*Carpeta: task_3_sales_forecast*)

Tras considerar nuestras necesidades de negocio, hemos trabajado en un conjunto de modelos de **pronóstico de ventas a nivel de tienda-producto**, basándonos en los **top20 productos más vendidos de cada tienda** (200 predicciones), y probando ambos algoritmos (**ARIMA** y **Holt-Winters**) de predicción basados en series temporales.

Para ambos casos, se ha definido que el algoritmo se ajuste con los mejores parámetros a la serie temporal:

- Para el caso del ARIMA, se ha utilizado el método “**auto_arima**”, de la biblioteca “**pmdarima**”, que permite seleccionar automáticamente los hiper-parámetros p, d y q (autoregresión, diferenciación y medias móviles) del modelo ARIMA, basándose en métricas como el AIC.
- Para el caso de **Holt-Winters**, hemos implementado una búsqueda de hiper-parámetros para predecir la serie temporal y encontrar la combinación de parámetros que proporcione el mejor ajuste a la serie temporal para los diferentes artículos (la combinación que diera lugar al menor RMSE).

3.2. Desarrollar un panel interactivo para interactuar con los pronósticos de venta.

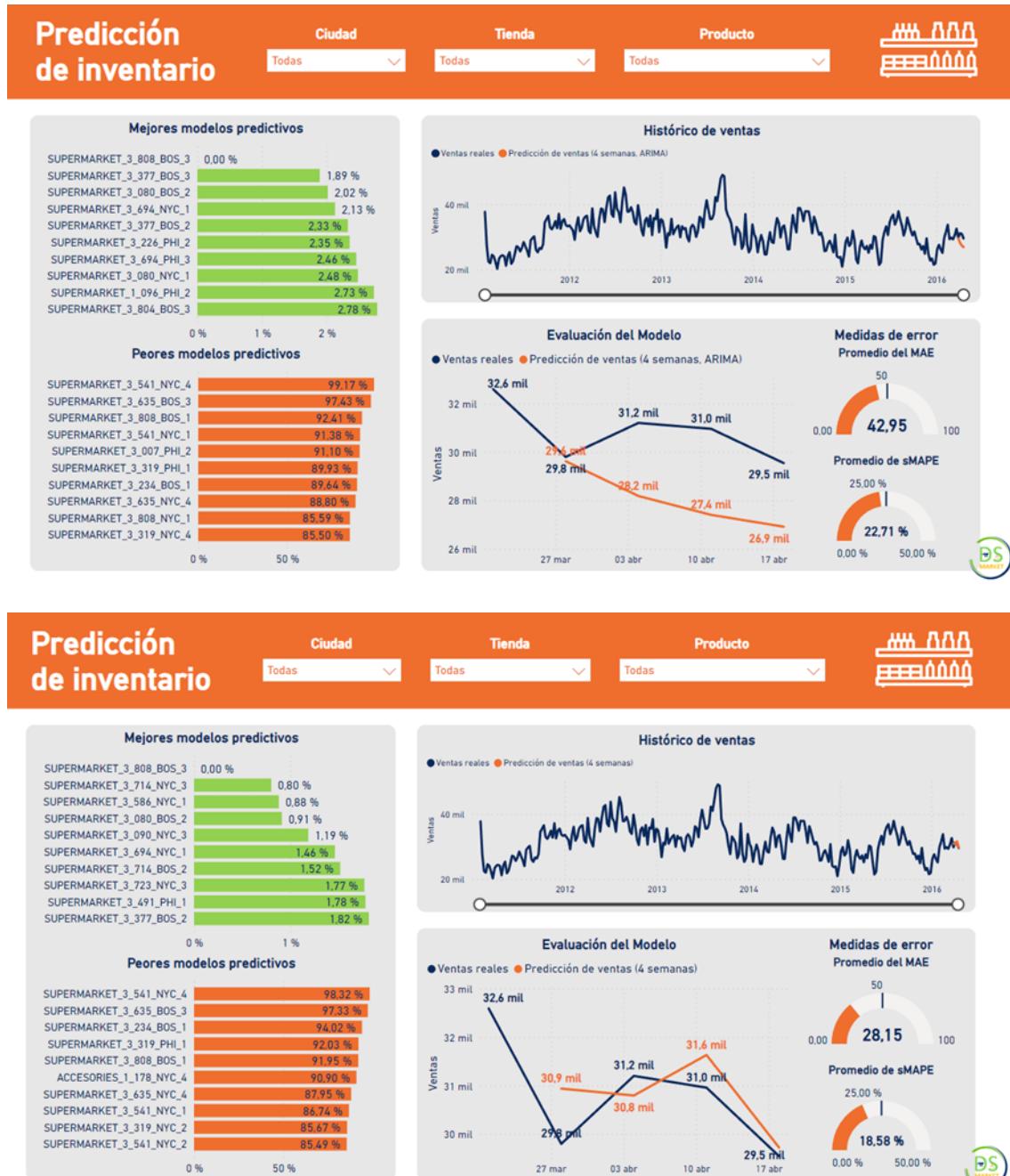
Para garantizar la transparencia y el control de la calidad, hemos incluido un sistema de informes que muestran las medidas de error de nuestros pronósticos frente a las ventas reales. En nuestro dashboard hemos incluído dos métricas:

- **Mean Average Error (MAE)**. El MAE es una métrica que mide el promedio de las diferencias absolutas entre los valores pronosticados y los valores reales. Una ventaja del MAE es que es fácil de interpretar, ya que representa la magnitud promedio de los errores de pronóstico, sin tener en cuenta su dirección (positiva o negativa).
- **Symmetrical Mean Average Percentage Error (sMAPE)**. El sMAPE es otra métrica comúnmente utilizada en pronósticos. Mide el porcentaje promedio de diferencia absoluta entre los valores pronosticados y los valores reales en relación con la suma de los valores reales. El sMAPE tiene la ventaja de ser simétrico, lo que significa que penaliza de manera equitativa tanto las sobreestimaciones como las subestimaciones en los pronósticos. De todos modos, no responde bien a la presencia de valores nulos.

Vamos a seguir la estrategia de la empresa **agregando los pronósticos independientes** para obtener las predicciones a nivel de ciudad/tienda, y poder evaluar de manera

general los modelos. Para ello, hemos generado un **dashboard** en powerBI que nos permite evaluar el promedio de los errores de los modelos, e interactuar de manera individual o agrupada con las predicciones.

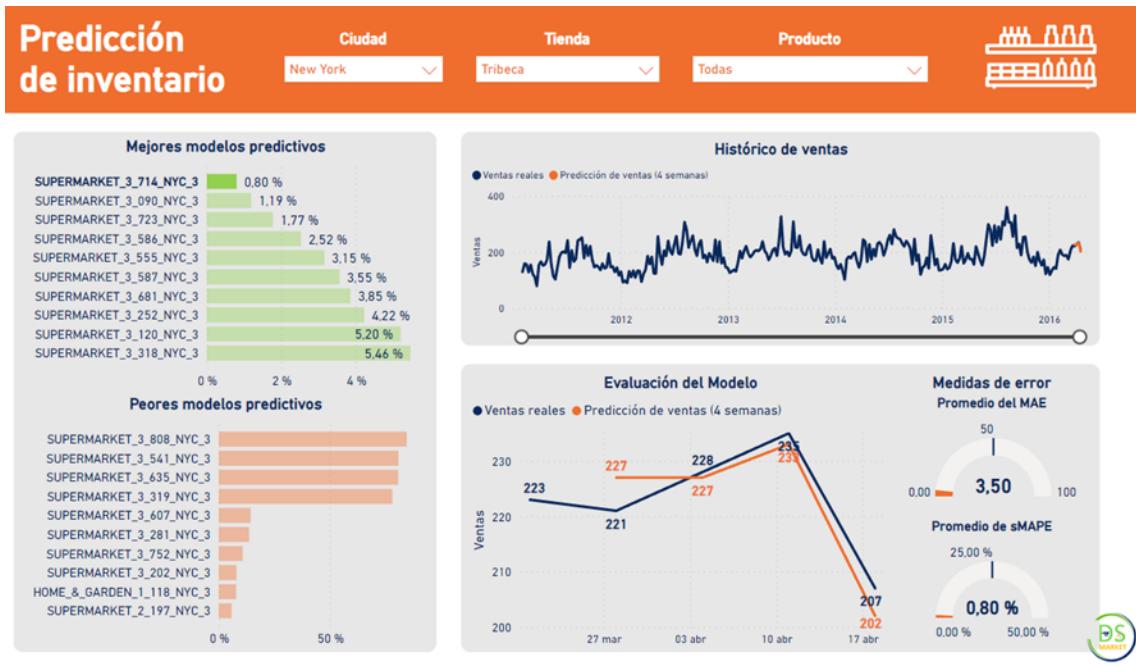
Tras generar los modelos, observamos como el **promedio de error de los modelos basados en ARIMA es superior al de los basados en Holt-Winters**, por lo que nos quedaremos con estos últimos.



Podemos ofrecer este panel interactivo como forma de interacción de los usuarios finales con la predicción de ventas futuras a nivel de tienda, p.e.: de los ítems de la ciudad de New York, y de la tienda de Tribeca:



También podemos observar nuestros mejores y peores modelos, para determinar si hace falta un reentrenamiento en base a sus métricas:



Estos modelos están diseñados para ser flexibles y escalables, de manera que podamos ajustar los intervalos de pronóstico según las necesidades de nuestro negocio. Sin embargo, es importante destacar que los modelos predictivos no son estáticos, y su precisión puede disminuir con el tiempo si no se actualizan.

Notebook: [3_4_item_prediction_hw_product_store_train261_test8.ipynb](#)

- Archivos de origen:
 - `df_forecast_1.xlsx` (Carpeta: `new_files`)
- Archivos resultantes:
 - `predictions_2023-07-22_hw_8weeks` (Carpeta: `task_3_sales_forecast`)

- [combined_predictions_2023-07-22_hw_8weeks.csv](#) (Carpeta: task_3_sales_forecast)
- [metrics_2023-07-22_hw_8weeks.csv](#) (Carpeta: task_3_sales_forecast)
- [real_data_2023-07-22.csv](#) (Carpeta: task_3_sales_forecast)

Hemos testeado los modelos basados en Holt Winters para que realizaran una predicción a más tiempo, analizando su rendimiento a 8 semanas. En este caso usamos las **261 primeras semanas para entrenar** los modelos, y las **8 últimas semanas para el test** (*forecasted vs. real*). Vemos como las métricas agregadas empeoran, en comparación con los modelos entrenados a 4 semanas vista.



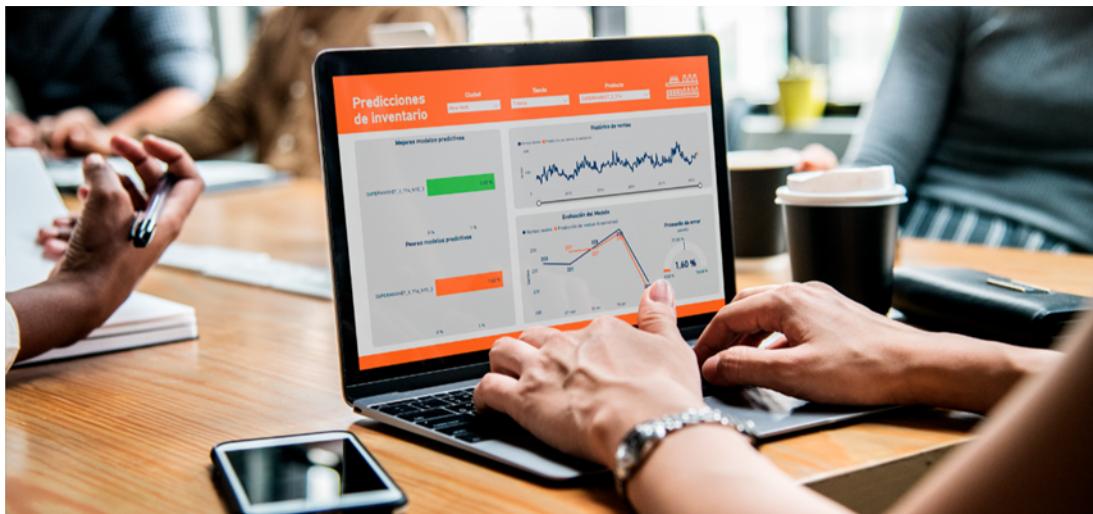
4. Caso de Uso: Reposición de Tiendas

El Departamento de Operaciones está interesado en utilizar nuestros modelos de pronóstico de ventas para la reposición de tiendas.

Para que puedan utilizarlos presentamos la propuesta de la realización e implementación de una API para utilizar a nivel interno. La API tendrá el objetivo de dar respuesta al problema del abastecimiento por medio de una solución escalable que se actualizará fácilmente en una primera etapa y automáticamente en etapas avanzadas del proceso de digitalización de DS Market.

4.1. Operacionalización de la API: herramientas necesarias

El Departamento de Operaciones busca utilizar los modelos de pronóstico de ventas para la reposición de tiendas, y para ello, se propone la realización e implementación de una API de uso interno. La API brindará una solución escalable que se actualizará fácilmente en una primera etapa y automáticamente en etapas avanzadas del proceso de digitalización de DS Market.



Para la operacionalización de la API en el proceso de reposición de tiendas, se utilizará **Python** con el **Flask framework**. Los usuarios finales, como el director financiero, accederán y utilizarán las predicciones del modelo a través de un panel de control interactivo con PowerBI, el cual se conectará a la base de datos que almacenará las predicciones y presentará la información de manera clara y comprensible.

La API se conectará con el dashboard de **PowerBI** mediante la función “**Web.Contents()**” para enviar nuevos datos de entrada y obtener nuevas predicciones semanales/mensuales. Además, se configurará un proceso en segundo plano que llamará a la API para obtener las últimas predicciones y enviará un informe semanal por correo electrónico utilizando la biblioteca `smtplib`.

Para llevar a cabo la API, se incorporarán las herramientas necesarias, como **Azure**, **DVC**, **Git**, **Docker** y **mlflow**. **Azure** fue elegido en lugar de AWS debido a su integración nativa con las tecnologías y aplicaciones de Microsoft ya existentes, lo que permitirá una transición más fluida a la nube y una mejor cohesión, además de evitar competencia con DS Market.



En la implementación de la API, se utilizará **DVC** y **Git** para gestionar eficientemente tanto las versiones de código como los datos. **Git** proporcionará un sólido control de versiones y colaboración, permitiendo un desarrollo seguro y organizado del código, mientras que **DVC** garantizará una gestión óptima de los conjuntos de datos, asegurando su coherencia y reproducibilidad a lo largo del proyecto. La combinación de ambas herramientas asegurará un control riguroso de cambios en el

desarrollo y los datos, facilitando la colaboración en equipo y brindando confianza en la calidad y trazabilidad de los resultados de la API.

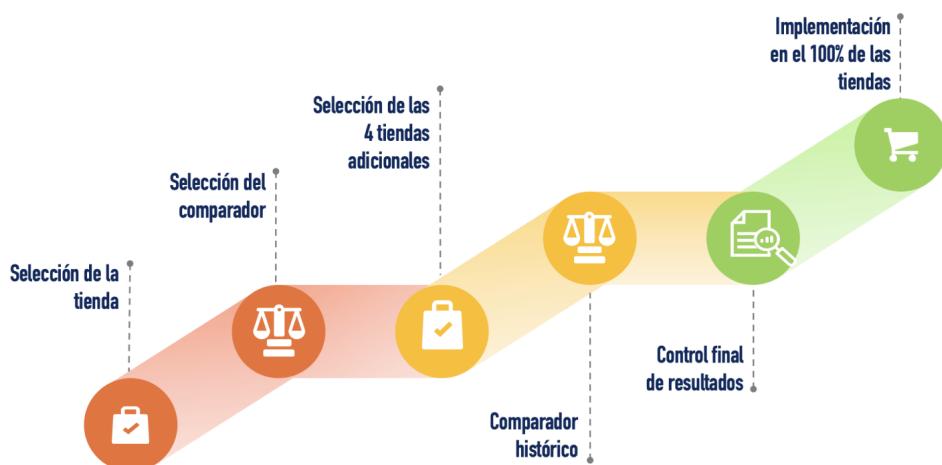
Se creará un archivo **Dockerfile** que describirá el entorno necesario para ejecutar la API y el modelo. El **Dockerfile** especificará la imagen base, copiará el código de la API y el modelo dentro del contenedor e instalará todas las dependencias requeridas. Una vez construida la imagen de **Docker**, se empaquetará el modelo entrenado y la API dentro de ella, asegurándose de incluir todo el código, datos y dependencias necesarias. Luego, el contenedor Docker se desplegará en el entorno de producción y se establecerán mecanismos de monitorización para evaluar el rendimiento y disponibilidad de la API y el modelo, documentando todo el proceso para facilitar el mantenimiento y futuras actualizaciones. Esto proporcionará un entorno coherente y portátil, facilitando el despliegue y mantenimiento de la solución en diferentes entornos de producción.

Además, se utilizará **MLflow** para rastrear las métricas de rendimiento de los modelos y se configurarán alertas para notificar al equipo si el rendimiento de alguno de ellos no cumple con los umbrales establecidos. Asimismo, la API lanzará un mensaje de "revisar por el supervisor" para predicciones que superen ciertos porcentajes en el MAE y sMAPE, previniendo situaciones disruptivas en las tendencias de ventas, como el quiebre de stock.



En la propuesta de la API, se considera un "best case scenario" con recopilación y actualización semanal de los datos de ventas para entrenar el modelo y generar predicciones precisas. No obstante, se entiende la complejidad de este proceso y se plantea un "worst case scenario" donde la recopilación se realice de manera mensual. Esta decisión se basa en pruebas previas de los modelos y su coherencia con el conocimiento de mercado.

4.2. Despliegue de los modelos

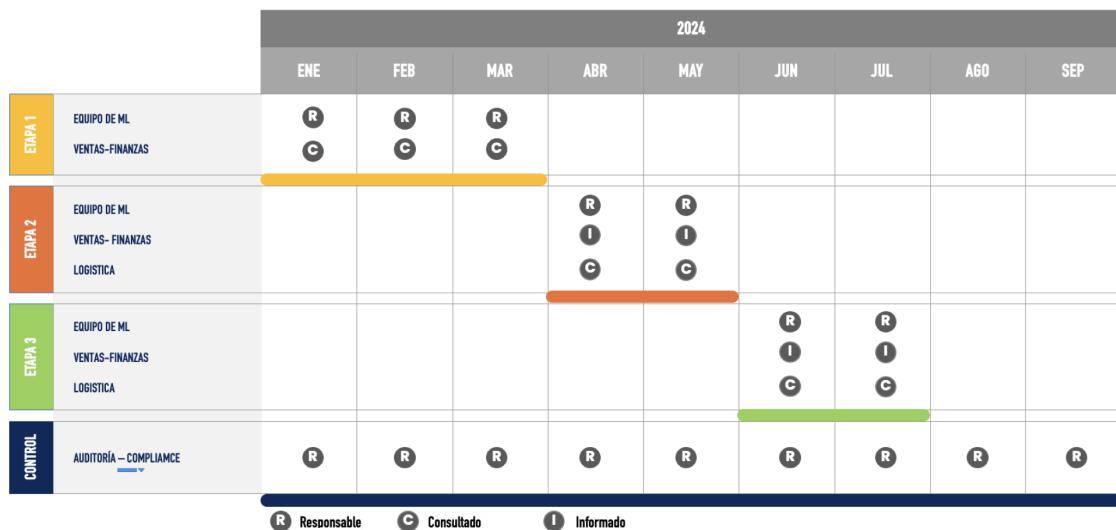


La implementación de la API no se realizará utilizando el método tipo "Big Bang", sino que se optará por un enfoque de "phase roll out". Estratégicamente, se seleccionará una tienda dentro de todas las existentes para la primera etapa. Los criterios de selección serán que la tienda posea al menos 1 producto de cada cluster, para verificar que las predicciones funcionan sin importar la clasificación de los productos, y que sea una tienda de tamaño medio para evitar impactos significativos en caso de errores iniciales. La tienda seleccionada será Back-Bay. Para evaluar las predicciones realizadas por la API, se escogerá la tienda South-End, que tendrá características similares a Back-Bay.

Una vez finalizada la primera etapa, se evaluará tanto el stock remanente en cada tienda como una prueba piloto de satisfacción del cliente para verificar posibles desabastecimientos. Si los resultados de la API igualan o mejoran el rendimiento en la tienda South-End, se procederá a la segunda etapa. En esta etapa, se seleccionarán 4 tiendas adicionales en las cuales se implementará la utilización de la API. En lugar de tiendas similares del mismo periodo, se utilizarán las ventas realizadas en las 4 tiendas seleccionadas durante el año anterior como comparador. Esto permitirá evaluar el desempeño del modelo predictivo considerando eventos como Ramadán, que afectan las tendencias en las ventas de DS Market.

En caso de que el desempeño del modelo predictivo implementado por medio de la API sea igual o mejor que el comparador, se avanzará a la tercera etapa del proceso. En esta etapa, se utilizará la predicción del modelo para realizar todas las reposiciones del stock, monitoreando continuamente los errores y controlando el stock remanente y la satisfacción del cliente para identificar posibles casos de desabastecimientos.

La implementación de esta API representa un nuevo proceso para DS Market. El equipo de ML será el responsable de la coordinación del mismo, y se incorporarán otros equipos, como ventas, finanzas y logística, para asegurar la colaboración y participación de todos los actores en el cambio hacia la utilización de la API. Además, se establecerá una cuarta etapa de control que será transversal a todo el proceso. Los equipos de auditoría y compliance de DS Market serán responsables de supervisar y garantizar el correcto funcionamiento y cumplimiento del proceso en todas las etapas, acompañando al equipo de ML para evitar posibles desviaciones. Esta transición representa una evolución importante en la metodología de DS Market y requerirá la implicación y seguimiento de múltiples equipos para lograr una implementación exitosa y un flujo de funcionamiento adecuado.



Los tiempos estimados en la imagen consideran un periodo de aprobación previo al lanzamiento del proyecto, iniciando la etapa 1 en Enero de 2024. No obstante, los plazos serán adaptados a las necesidades de DS Market para brindar una propuesta realista de implementación que determine plazos adecuados para cada una de las etapas del proceso.

En este contexto, es importante destacar que la implementación de la API para solucionar el problema de abastecimiento en DS Market se alinea con el proceso de digitalización en curso. Nuestro equipo se encuentra trabajando en la implementación de la API, siguiendo un enfoque de "continuous integration," "continuous delivery," y finalizando con la etapa de "continuous deployment." Esto permitirá que el código se ponga en funcionamiento sin necesidad de intervención humana, avanzando hacia la digitalización completa de DS Market.

Con esta propuesta, DS Market obtendrá una solución escalable y automatizada que mejorará significativamente la gestión del stock en sus tiendas. La adaptación de los plazos garantizará una transición exitosa hacia la utilización de la API, y el enfoque en la digitalización posicionará a DS Market para enfrentar desafíos futuros. Nuestro equipo se compromete a acompañar y guiar a DS Market en este proceso de transformación, asegurando una implementación exitosa y una mejora continua en la eficiencia operativa de la empresa.

5. PowerBI

Además de las tareas específicas, Michelle quiere que desarrollemos un servicio de inteligencia empresarial (BI) que proporcione actualizaciones regulares sobre los principales resultados de tu análisis.

Carpeta: task_1_powerbi

“CapstoneDSMarket.pbix”