Práctica Profesionalizante I

Entrega Nº 4: Implementación de soluciones

Equipo: “Data Voyagers”

BADIN, María Paula

LEDEZMA, Mariano

PERALTA, María Laura

Introducción

Como conclusión de los análisis y modelados previos, DataVista Analytics busca implementar una solución integral que optimice la cadena de suministro a través de modelos de machine learning. Este informe detalla el desarrollo y la implementación de los modelos seleccionados en un entorno controlado y su arquitectura, desde el procesamiento de datos hasta la entrega de predicciones en tiempo real.

Tratamiento previo de los datos

Previo a la utilización de las técnicas de agrupamiento, se realizó un análisis exploratorio de los datos que nos permitió tener un acercamiento a cómo se distribuyen los mismos con respecto a las variables de interés, adaptar valores numéricos, convertir valores categóricos y por último graficar las correlaciones entre las distintas variables.

Selección y Preparación de Modelos Predictivos

Se investigaron y seleccionaron los siguientes algoritmos de modelado predictivo: Regresión Lineal Simple , Árboles de Decisión, Regresión Logística, Clustering.

Regresión Lineal Simple

Conceptos Básicos:

- **Regresión Lineal Simple**: Este método estadístico permite modelar la relación entre una variable independiente (predictora) y una variable dependiente (respuesta).

El objetivo principal de este análisis es predecir las ganancias por pedido en función de las ventas realizadas, considerando las ventas como variable independiente y las ganancias como variable dependiente.

- **Ecuación de la Regresión**: La ecuación básica de la regresión lineal simple se expresa como:

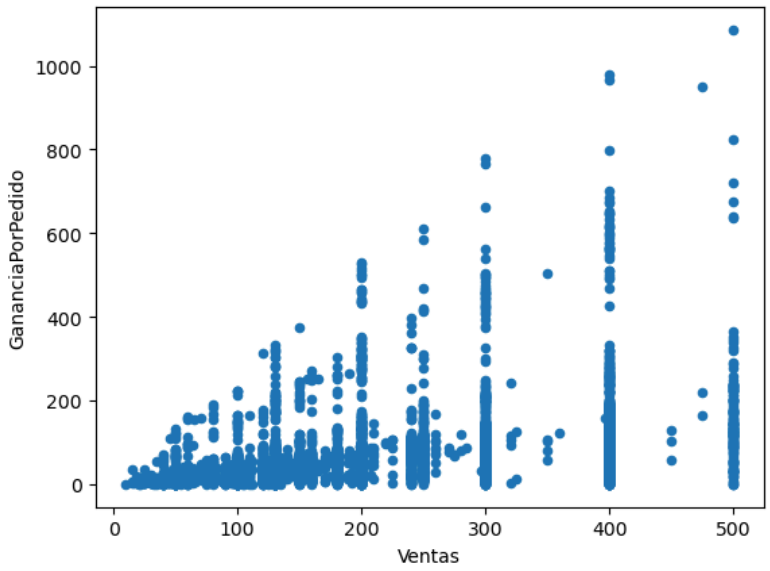
  $$Y = b + wXY = b + wXY = b+wX$$

  donde:

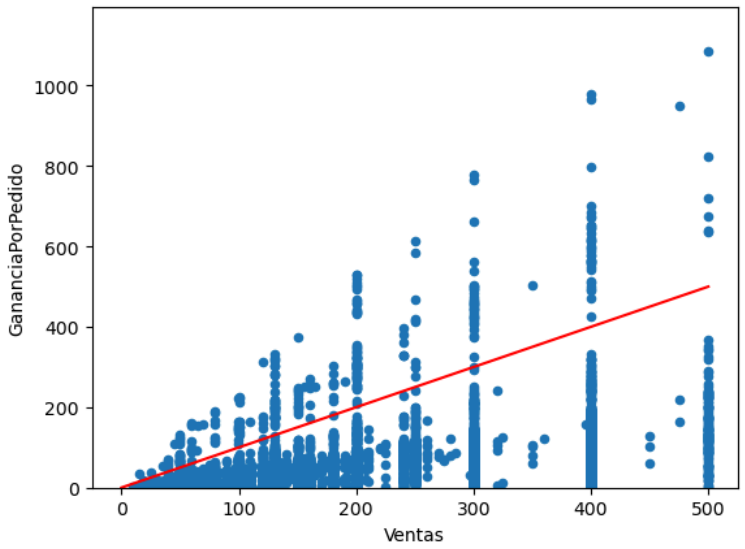
  - Y es la variable dependiente (ganancias).
  - X es la variable independiente (ventas).
  - b representa la intersección de la línea con el eje Y.
  - w es la pendiente de la línea de regresión.

Procedimiento:

1. **Recopilación de Datos**:
    - Filtramos los datos de ventas y ganancias para establecer su relación.
    - Eliminamos valores vacíos o ceros.
    - Estructuramos las columnas para facilitar la visualización.
2. **Exploración de Datos**:
    - Visualizamos los datos mediante un gráfico de dispersión, lo que nos permite observar la relación entre ventas y ganancias y evaluar si la regresión lineal es apropiada.



3. **Ajuste del Modelo**:
    - Utilizamos la recta de regresión para establecer la linealidad deseada.



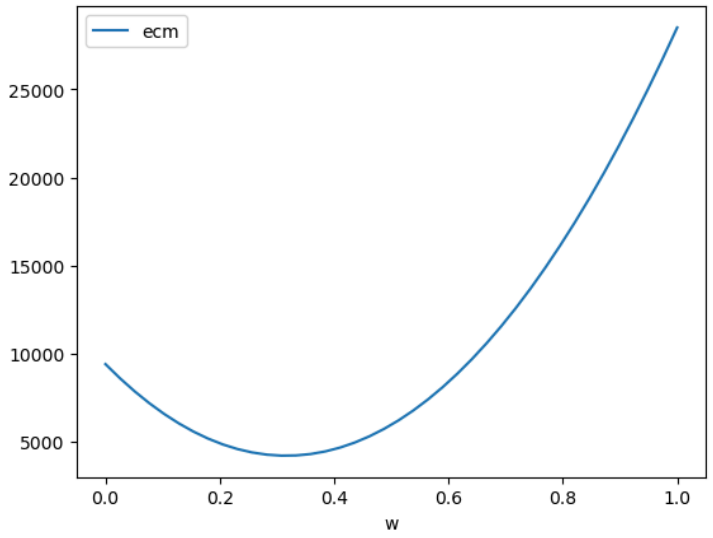
    - A través de diferentes pruebas, calculamos los coeficientes w y b que optimizan el modelo.

Luego de realizar pruebas con diferentes valores de w (entre 0 y 2.5) y comparar los valores del ecm obtenido, podemos optimizar w calculando el mínimo promedio. Para ello realizamos una función cuadrática entre w y ecm y calculamos el vértice de la parábola (o valor mínimo)

```
#grid del error:
w = np.linspace(0, 1, num=40) #trabajaremos con 20 valores diferentes dentro del rango
grid_error = pd.DataFrame(w, columns=['w'])
grid_error.head()
```

- Aunque es posible usar librerías de Python para automatizar estos cálculos, en este ejemplo optamos por realizarlo manualmente para entender cada paso del proceso.

4. **Evaluación del Modelo**:
   - Analizamos los errores de predicción y calculamos el error cuadrático medio.
   - Comparamos estos errores para obtener la función cuadrática que minimiza el valor de error.



- Además, utilizamos una librería de Python para confirmar que los resultados obtenidos mediante ambos métodos coincidían.

```
# usando sklear para saber los valores optimos
from sklearn.linear_model import LinearRegression

# definiendo input y output
X_train = np.array(df_lineal['Ventas']).reshape((-1, 1))
Y_train = np.array(df_lineal['GananciaPorPedido'])

# creando modelo
model = LinearRegression(fit_intercept=False)
model.fit(X_train, Y_train)

# imprimiendo parametros
print(f"intercepto (b): {model.intercept_}")
print(f"pendiente (w): {model.coef_}")
```

```
intercepto (b): 0.0
pendiente (w): [0.31621416]
```

5. **Interpretación**:
    - Un valor positivo para w indica que, a medida que las ventas aumentan, las ganancias también tienden a incrementarse.
    - Esto permite hacer predicciones sobre las ganancias con base en nuevas cifras de ventas.
6. **Validación**:
    - Se sugiere utilizar un conjunto de datos diferente para validar el modelo y asegurar que generaliza correctamente.

Conclusión:

El modelo final de regresión lineal simple demostró ser efectivo en la predicción de ganancias por pedido a partir de las ventas. Esto confirma la validez de la metodología empleada y sugiere su implementación en futuros análisis de datos similares.

Clasificación Binaria

Consideraciones Generales

La clasificación binaria es un tipo de aprendizaje supervisado donde el objetivo es predecir una de dos clases posibles. Es decir, es la tarea de clasificar los elementos de un conjunto en dos grupos sobre la base de una regla de clasificación.

A modo descriptivo mencionamos en diferentes categorías, las diferentes alternativas que existen actualmente para desarrollar este tipo de análisis, luego en particular abordaremos como más detalles las técnicas utilizadas para nuestro caso elegidos en función de las necesidades de análisis de nuestra empresa.

- **Técnicas Clásicas:** Regresión Logística, Árboles de Decisión, Máquinas de Vectores de Soporte (SVM), K-Nearest Neighbors (KNN).
- **Técnicas de Ensemble:** Random Forest, Gradient Boosting, AdaBoost.
- **Redes Neuronales Artificiales:** Perceptrón Multicapa (MLP), Redes Neuronales Convolucionales (CNN), Redes Neuronales Recurrentes (RNN)

Consideraciones: La elección del algoritmo de machine learning depende de varios factores. Para grandes conjuntos de datos, los algoritmos de ensemble y las redes neuronales son eficientes. Si la interpretabilidad es importante, los árboles de decisión y la regresión logística son mejores opciones. El tiempo de entrenamiento y las métricas de evaluación también influyen en la decisión final.

En el contexto de la logística y los pedidos, abordaremos análisis con las técnicas clásicas: Árboles de decisión y Regresión logística

Regresión Logística

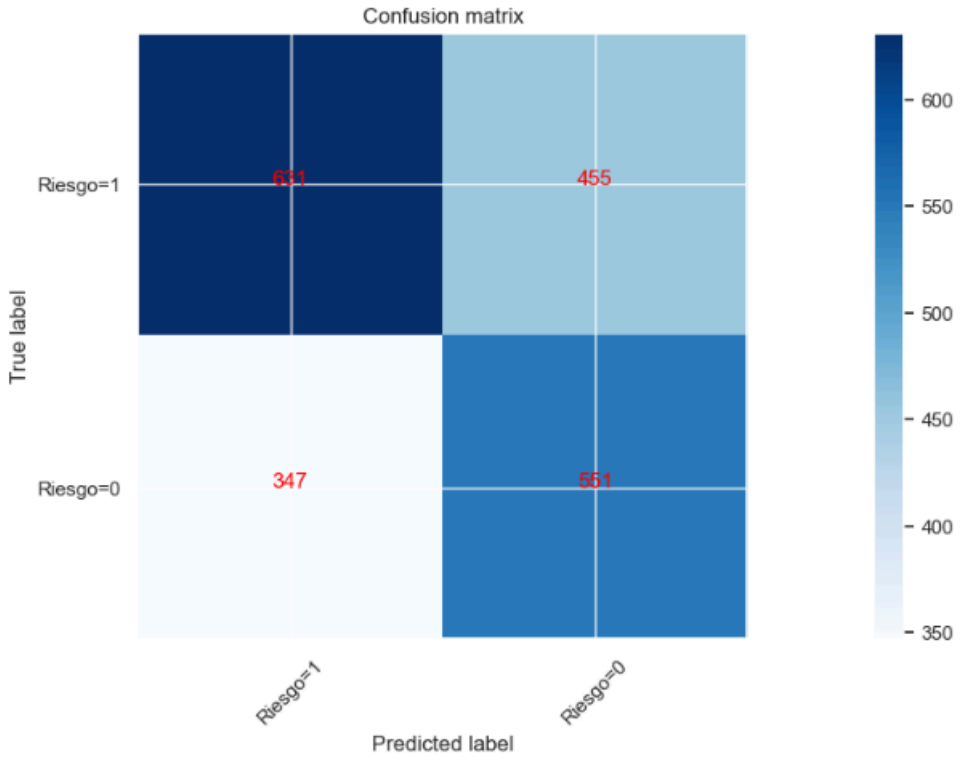
Las regresiones logísticas son utilizadas para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictivas. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores.

Es una extensión de la regresión lineal que se usa comúnmente para clasificación binaria. Aplica una función logística (sigmoide) a la salida de una regresión lineal para producir probabilidades entre 0 y 1.

La Empresa pretende obtener estimaciones sobre el riesgo de que un pedido no sea entregado a tiempo basándose en la información asociada a cada pedido en la columna llamada **" riesgo de entrega tardía "**, una variable binaria (0: entrega a tiempo, 1: entrega tardía), en conjunto con las variables que puedan influir en el resultado.

Para esto utilizamos los diferentes valores de un conjunto de variables que consideramos independientes junto a los valores de una variable dependiente para procesar por algoritmos de aprendizaje automático. En proceso consistió en:

1- Creación de una variable independiente y estandarización
2- Creación de la variable dependiente
3- Separar el conjunto de datos entre para Entrenamiento y Test (80-20)
5- Entrenar la regresión



Desarrollo y Validación de la Regresión Logística

La regresión logística arroja una probabilidad de que el resultado sea 1 y una probabilidad de que sea cero

- se asigna el número 1 o 0 según sea mayor la probabilidad

- Se utiliza la regresión entrenada para que arroje las probabilidades

6- Mostramos los resultados

Resultados:
Se tienen 561 casos marcados como 0 y que efectivamente eran 0.

1. De todos los casos que eran efectivamente cero (347 + 551 = 898) el 0,613% (551/898) fue correctamente predecido -> Esta métrica es conocida como calidad

1. De todos los casos que el modelo marco como 0 (455 + 551 = 1006) el 0,547% (551/1006) eran efectivamente cero -> Esta métrica es conocida como precisión

1. Se tienen 631 casos marcados como 1 y que efectivamente eran 1.

1. De todos los casos que eran efectivamente uno (631 + 455 = 1086) el 0,581% - (631/1086) fue correctamente predecido

1. De todos los casos que el modelo marco como 1 (631 + 347 = 978) el 0,645% - (631/978) eran efectivamente uno

Reporte con Python

```
print (classification_report(y_test, yhat))
```

	precision	recall	f1-score	support
0	0.55	0.61	0.58	898
1	0.65	0.58	0.61	1086
accuracy			0.60	1984
macro avg	0.60	0.60	0.60	1984
weighted avg	0.60	0.60	0.60	1984

observaciones:

- F1 Score es una combinación de la precisión y la calidad, indica con que tanta certeza el modelo está funcionando, se puede decir que en general el modelo tiene un 60% de certeza
- Análisis por Clase:

Clase 0:
Precisión: 0.55 (moderada): El modelo tiene algunas falsas positivas.
Recall: 0.61 (moderada): El modelo identifica algunas muestras negativas incorrectamente.
F1-score: 0.58 (moderada): Desequilibrio entre precisión y recall.
Soporte: 898 (menor soporte): La clase 0 es menos frecuente.

Clase 1:
Precisión: 0.65 (buena): El modelo tiene pocas falsas positivas.
Recall: 0.58 (moderada): El modelo identifica algunas muestras positivas incorrectamente.
F1-score: 0.61 (moderada): Desequilibrio entre precisión y recall.

Soporte: 1086 (mayor soporte): La clase 1 es más frecuente.

Resumen General:

Desempeño moderado: El modelo tiene un rendimiento moderado en general, con precisión y recall similares para ambas clases.

Desequilibrio de clases: La clase 1 tiene un soporte mayor que la clase 0, lo que puede influir en las métricas globales.

Áreas de mejora: Se podrían explorar técnicas para mejorar la precisión y recall en ambas clases, especialmente en la clase 0.

Árboles de Decisión

Un árbol de decisión es un modelo de aprendizaje automático que toma decisiones basadas en una serie de preguntas (nodos) para clasificar o predecir datos. Cada pregunta divide los datos en ramas, lo que lleva a decisiones finales (hojas) que representan las predicciones. Es una técnica de aprendizaje supervisado utilizada en clasificación y regresión.

Este modelo busca predecir el **estado de entrega** de los pedidos basándose en diversas características, como los **días de envío reales, el riesgo de entrega tardía y la categoría del producto**.

El proceso consistió en seleccionar como variable objetivo: estado de entrega y como características descriptivas: días de envío reales, el riesgo de entrega tardía y la categoría del producto.

Se dividió el dataset en un conjunto de entrenamiento, 80% y un conjunto de prueba, 20%.

Se procedió al entrenamiento del modelo, utilizando la métrica de entropía como criterio para las divisiones de nodos.

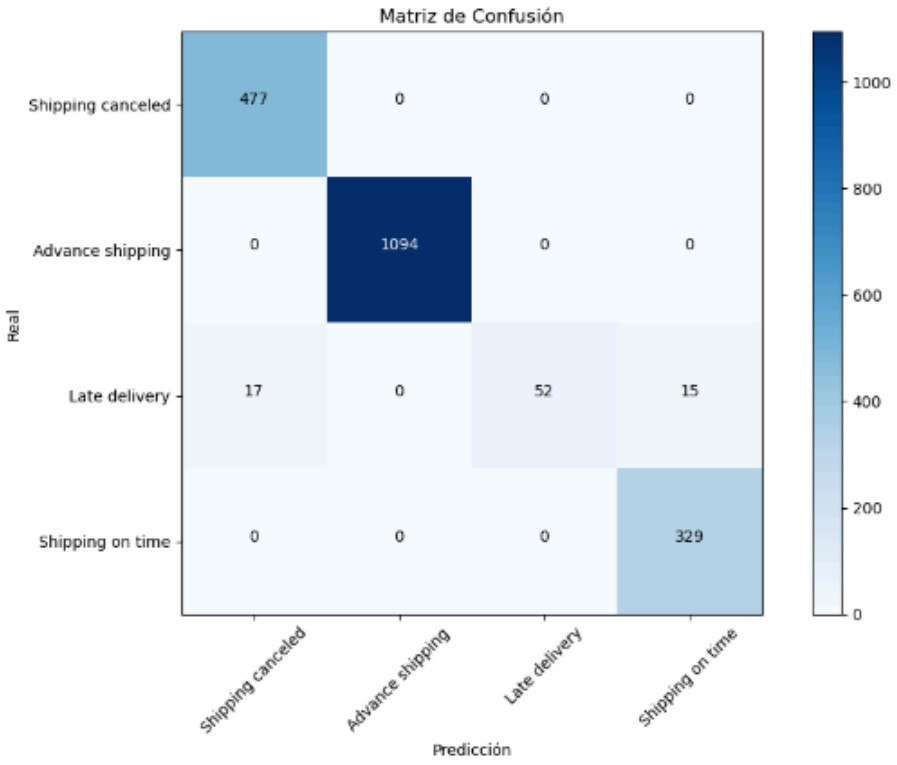
Este modelo inicial mostró una buena precisión (98.13%), pero con espacio para optimización.

Para mejorar el rendimiento del modelo, se realizó una búsqueda de hiperparámetros, resultando en los mejores valores:

- Criterio: 'entropy'
- Profundidad Máxima: 5
- Mínimo de Muestras para Dividir: 2

Concluimos que el modelo tiene un excelente rendimiento para predecir correctamente las clases mayoritarias como “Entrega tardía” y “Envío demorado”. Sin embargo, la clase minoritaria “Envío cancelado” presentó un recall ligeramente inferior, aunque mejorando en comparación con el modelo inicial.

	precision	recall	f1-score	support
Advance shipping	0.97	1.00	0.98	477
Late delivery	1.00	1.00	1.00	1094
Shipping canceled	1.00	0.62	0.76	84
Shipping on time	0.96	1.00	0.98	329
accuracy			0.98	1984
macro avg	0.98	0.90	0.93	1984
weighted avg	0.98	0.98	0.98	1984



Resumen

El modelo optimizado de árboles de decisión ha demostrado ser una herramienta efectiva para predecir el estado de entrega de los pedidos con una precisión del 98.39%. Esto facilita su adopción por las áreas operativas de la empresa, permitiendo prever con alta precisión si un pedido será entregado a tiempo. Esto es clave para la optimización de rutas logísticas y la gestión de inventarios, contribuyendo a la reducción de costos y mejorando la satisfacción del cliente. Sin embargo, existe un área de mejora en la identificación de las cancelaciones, a pesar de los buenos resultados obtenidos.

Agrupamiento - Clustering

Introducción

El agrupamiento, o clustering, es una técnica de aprendizaje no supervisado que busca identificar grupos naturales en un conjunto de datos. Su objetivo es agrupar objetos que sean similares entre sí en función de ciertas características para poder revelar patrones ocultos y tendencias que no son evidentes a simple vista.

A modo descriptivo mencionamos en diferentes categorías, las diferentes alternativas que existen actualmente para desarrollar este tipo de análisis, luego en particular abordaremos como más detalles las técnicas utilizadas para nuestro caso elegidos en función de las necesidades de análisis de nuestra empresa.

Técnicas Actuales de Clustering

- **Técnicas Clásicas:** K-means, Jerárquico, DBSCAN
- **Técnicas Basadas en Densidad:** OPTICS, DBSCAN-based algorithms, .
- **Técnicas Basadas en Modelos:** Gaussian Mixture Models (GMM), Density-based clustering.
- **Técnicas Basadas en Redes Neuronales:** Autoencoders, Deep Clustering

Consideraciones al Seleccionar una Técnica:

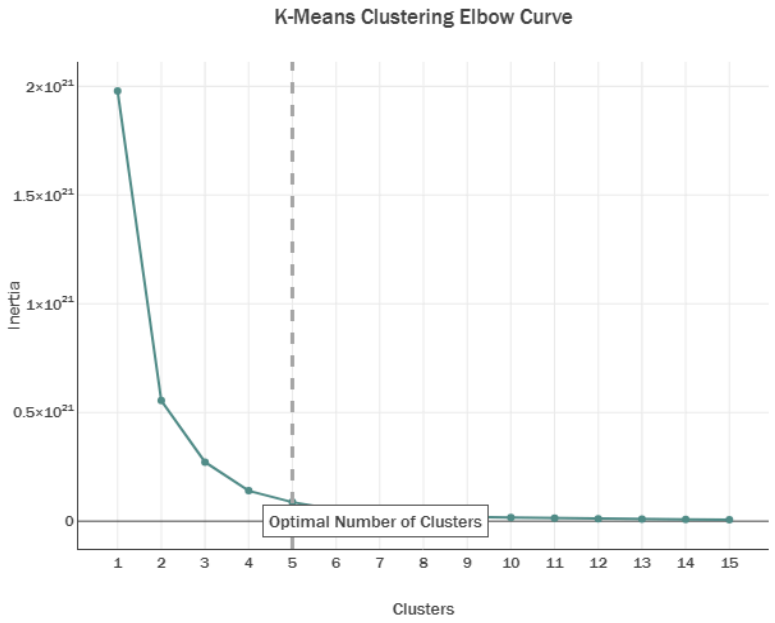
- A la hora de elegir un algoritmo, tendremos en cuenta la forma y distribución de los datos, además de tener en cuenta que algunos algoritmos requieren especificar el número de clusters, mientras que otros lo determinan automáticamente.
- Con respecto a los clusters, se tendrá en cuenta la forma que adoptan: pueden ser esféricos, elípticos o de formas más complejas. Su robustez frente al ruido, su escalabilidad y la eficiencia computacional.

En el contexto de la logística y los pedidos, el agrupamiento permite a nuestra empresa segmentar a sus clientes en grupos homogéneos, lo que facilita la toma de decisiones estratégicas.
Para efectuar nuestro análisis implementaremos técnicas clásicas: K-Means, Jerárquico y DBSCAN.

Agrupamiento de K-Means

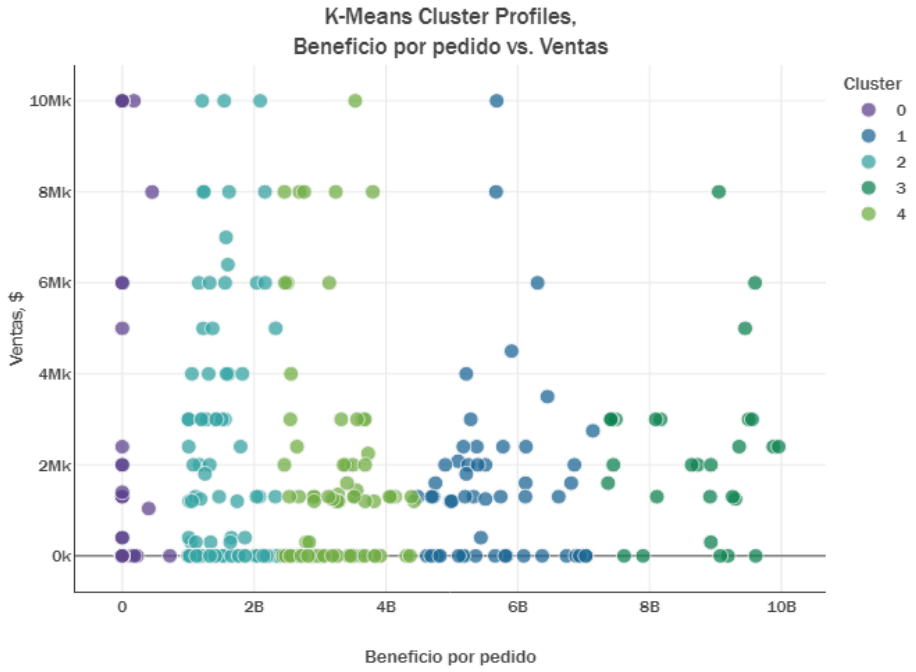
El agrupamiento de K-Means es un método de agrupamiento simple pero poderoso que crea ‘k’ segmentos distintos de los datos donde la variación dentro de los grupos es lo más pequeña posible.
1- Para encontrar el número óptimo de grupos, se prueban diferentes valores de k y se calcula la inercia, o el puntaje de distorsión, para cada modelo.
2-La inercia mide la similitud de los grupos calculando la distancia total entre los puntos de

datos y su centro de grupo más cercano.
3- Los grupos con observaciones similares tienden a tener distancias más pequeñas entre ellos y un puntaje de distorsión más bajo en general.



El gráfico anterior muestra los valores de inercia para cada modelo K-Means con clústeres entre 1 y 15. El punto de inflexión en el gráfico se produce en unos 5 grupos, donde la inercia comienza a estabilizarse. Esto indica que el número óptimo de clústeres, k es igual a 5.

Gráfico de los clústeres en función de Beneficio por Pedio y ventas



El modelo K-Means segmenta los datos en distintos grupos en función del Beneficio por pedido y las ventas.

- Se puede apreciar los 5 grupos, que reportan beneficios de menor a mayor en el siguiente orden: 0, 2, 4, 1 y 3.
- La variación en todos los grupos en el segmento más bajo de ventas 0k -4mk se concentra la mayoría de los pedidos, teniendo algunas diferencias en los extremos, es decir los grupos 0 y 3
- En el segmento superior de ventas, 4mk-10mk prevalecen los grupos 2 y 4

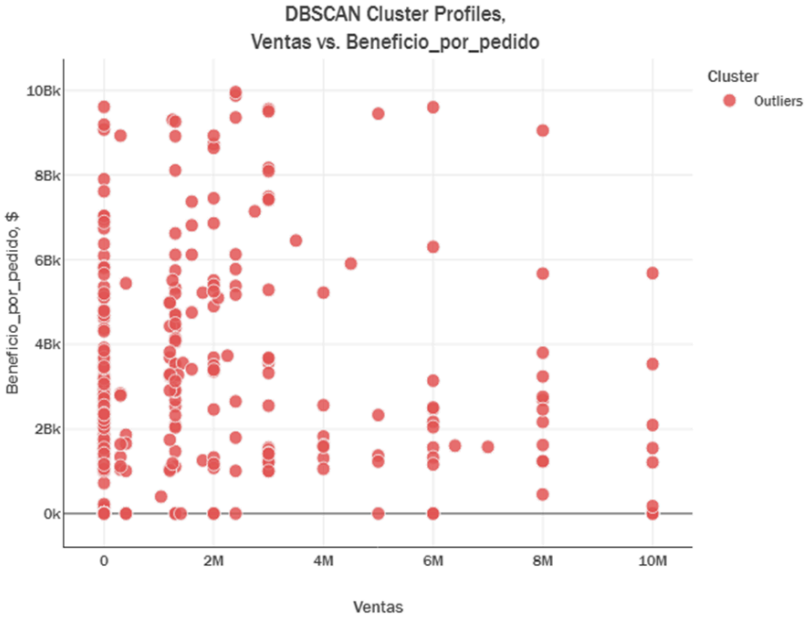
Agrupamiento Jerárquico

El siguiente método de agrupación en clústeres es el agrupamiento jerárquico. Utilizando un enfoque aglomerativo, la agrupación jerárquica une grupos de observaciones de abajo hacia arriba, y cada observación comienza su propio agrupamiento.
Luego, el modelo une pares de observaciones que son más similares entre sí en función de su distancia euclidiana y combina iterativamente pares de clústeres en función de las distancias entre los grupos hasta que se hayan fusionado todas las observaciones.

- En el centro del gráfico se encuentra el grupo 2 con los beneficios en rangos de 4 y 6 BK
- El grupo 0 en la parte superior con los valores mayores y el grupo 1 en la parte inferior con los valores menores

DBSCAN

La técnica de agrupación en clústeres espaciales basada en la densidad de aplicaciones con ruido (DBSCAN).
DBSCAN segmenta los datos en función de la densidad de las observaciones, donde las áreas de alta densidad se separan de las áreas de baja densidad.
El modelo también puede identificar clústeres de formas únicas y detectar valores atípicos dentro de los datos, aunque es sensible a las densidades variables de las observaciones.



Estos segmentos se asemejan a los clústeres del modelo K-Means, con los valores atípicos identificados en rojo. En general, hay bastantes valores atípicos en el gráfico, lo que probablemente se deba a la variación en las densidades de los clústeres.