Práctica Profesionalizante I

Entrega Nº 3: Modelado Predictivo

Equipo: “Data Voyagers”

BADIN, María Paula

LEDEZMA, Mariano

PERALTA, María Laura

Modelado Predictivo

Introducción

Las empresas que gestionan cadenas de suministro enfrentan desafíos continuos relacionados con la previsión de la demanda, la gestión de inventarios y la optimización de rutas de transporte. Estos problemas pueden derivar en costos operativos elevados, demoras en la entrega y una disminución en la satisfacción del cliente. DataVista Analytics, consciente de estos desafíos, ha diseñado una solución basada en ciencia de datos y machine learning para abordar estos problemas de manera eficiente y proactiva.

El presente informe describe la implementación de un modelo predictivo que permite mejorar la gestión de la cadena de suministro. A través de técnicas avanzadas de análisis de datos y algoritmos de machine learning, este proyecto busca optimizar las operaciones logísticas, mejorar la toma de decisiones y, en última instancia, reducir los costos operativos mientras se incrementa la satisfacción del cliente.

Se seleccionaron cuatro algoritmos diferentes: regresión lineal simple, árboles de decisión, regresión logística y clustering, para abordar diferentes aspectos del problema.

Resumen de las etapas anteriores

Análisis Exploratorio de Datos (EDA)

¿Qué es el Análisis de datos exploratorio?

El análisis de datos exploratorio (EDA) lo utilizan los científicos de datos para analizar e investigar conjuntos de datos y resumir sus principales características, empleando a menudo métodos de visualización de datos. Mediante EDA podemos determinar la mejor manera de manipular los orígenes de datos para obtener respuestas necesarias en cuanto a: descubrir patrones, detectar anomalías, probar una hipótesis o comprobar supuestos.

Usos del EDA

1. Se utiliza principalmente para:
2. Ver qué datos pueden revelarse más allá de la tarea de modelado formal o las pruebas de hipótesis.
3. Permite conocer mejor las variables del conjunto de datos y las relaciones entre ellas.
4. También permite determinar si las técnicas estadísticas que está considerando para el análisis de datos son apropiadas.

¿Por qué es importante el análisis exploratorio de datos en ciencia de datos?

- El principal objetivo del EDA es consultar los datos antes de hacer cualquier suposición.
- Permite identificar errores obvios.
- Comprender mejor los patrones en los datos.
- Detectar valores atípicos o sucesos anómalos.
- Encontrar relaciones interesantes entre las variables

Herramientas comunes para EDA

- Python: Lenguaje de programación interpretado y orientado a objetos cone estructuras de datos incorporadas de alto nivel.
- Librerias de procesamiento automatico: herramientas como DataPrep, IDAtaProfiling permiten un acercamiento rapido al problema brindando información relevante.

Procesos de Preparación de datos

Adquisición de Datos: El primer paso es adquirir los datos en un formato accesible, como un documento de excel o csv. El dataset utilizado consta de 180.519 filas y 53 columnas.
Exploración de Datos: Se realizó un análisis visual y un análisis de medidas estadísticas como la media, mediana y desviación estándar para evaluar la distribución y variabilidad de los datos.
Limpieza de datos: Incluye la corrección de errores de entrada, eliminación de duplicados, valores faltantes , y columnas no significativas.
Transformación de datos: Transformación de datos para convertirlos en un formato adecuado para el análisis, incluyendo la conversión de fechas y precios a formatos numéricos.

Resultados del EDA

Algunos de los resultados destacables del análisis son:

- **Días de envío:** Valores medios de 3.5 y 2.93 días para envíos reales y programados, respectivamente.
- **Beneficio por pedido**: Media de 21.97 con una alta desviación estándar, indicando gran variabilidad.
- **Riesgo de entrega Tardía:** Valor medio de 0.55 indicando que alrededor del 55% de los pedidos tienen un riesgo de entrega tardía.

Análisis Univariado

El análisis univariado se enfoca en la evaluación de una sola variable a la vez para resumir y entender sus características fundamentales. Durante la exploración inicial de los datos, se revisaron diversas variables categóricas y numéricas clave.

- **Variables categóricas:** categorías de producto, métodos de pago, modos de envío, segmento de clientes y estado de entrega.
- **Variables numéricas:** Días de envío (real y programado), beneficio por pedido, riesgo de entrega tardía y cantidad de artículos por pedido.

Transformación de datos

Precios y beneficios por pedido: se implementó una función de limpieza para eliminar caracteres no numéricos, permitiendo calcular estadísticas clave.

Visualización: se generaron histogramas y boxplots para visualizar la distribución y los valores atípicos en las siguientes variables:

- Días de envío (real y programado)
- Ventas
- Precio del producto
- Beneficio por pedido
- Cantidad de artículos por pedido

Resultados

- **Días de envio (Real):** La mayoría de los envíos se complementan en un rango de 0 a 6 días.
- **Días de envío (programado):** Los datos se concentran en torno a 4 días.
- **Ventas y precio del producto:** consistencia en los precios con pocos valores atípicos.
- **Beneficio por pedido:** alta variabilidad en los beneficios.
- **Cantidad de artículos por pedido:** distribución relativamente uniforme entre 1 y 5 artículos.

Análisis Bivariado

El análisis bivariado examina la relación entre dos variables para entender cómo una puede influir o relacionarse con otra. Este análisis permite identificar patrones, correlaciones y posibles áreas de mejora en el proceso de gestión y logística.

Ejemplos de este tipo de análisis:

- Estado de los pedidos según el País
- Estado de las entregas según el país

Visualizaciones multivariantes

Se utilizan para relacionar y comprender las interacciones entre los diferentes campos en los datos. Se incluyen:
Gráfico de dispersión: muestra como una variable afecta a otra.
Mapa de calor: representación gráfica de datos donde los valores se representan por color.

Consideraciones finales

El EDA garantiza que los resultados generados sean válidos y aplicables a las conclusiones y objetivos de negocio deseados.
Una vez completado el EDA, sus características pueden utilizarse para un análisis o modelado de datos más complejo, incluido machine learning.

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:

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

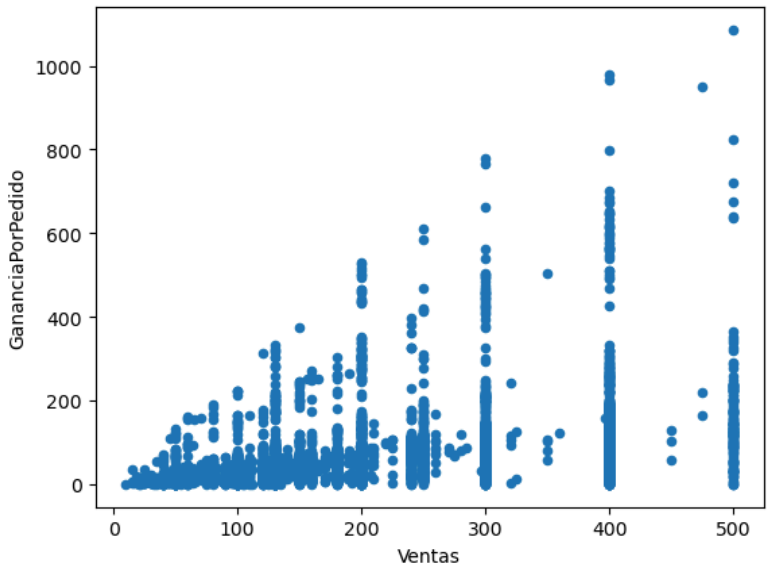
  donde:

  - 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.

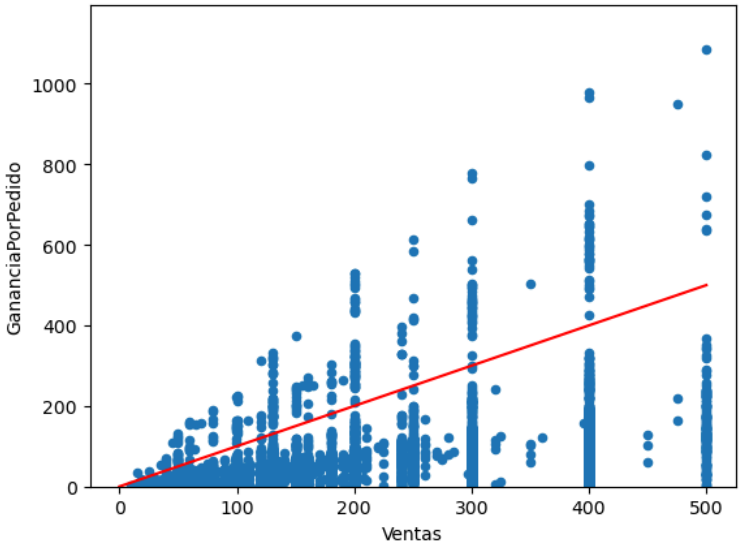
Procedimiento:

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.



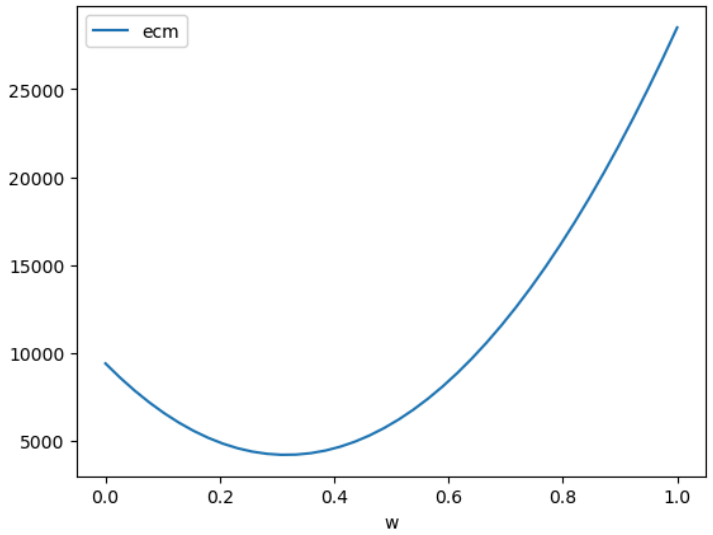
    - A través de diferentes pruebas, calculamos los coeficientes w y b que optimizan el modelo.

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.

```python
# 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

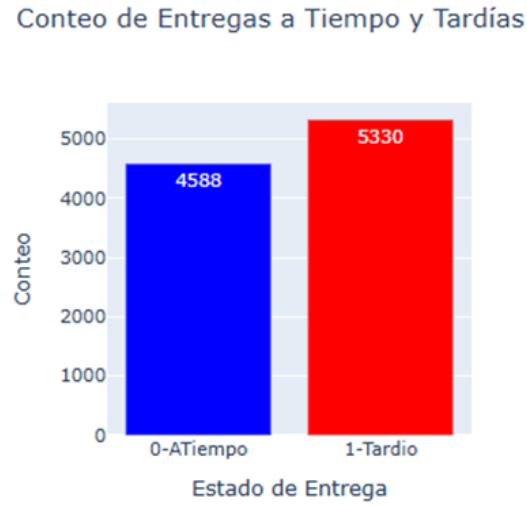
Las regresiones logísticas son utilizadas para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función

de las variables independientes o predictivas. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores.

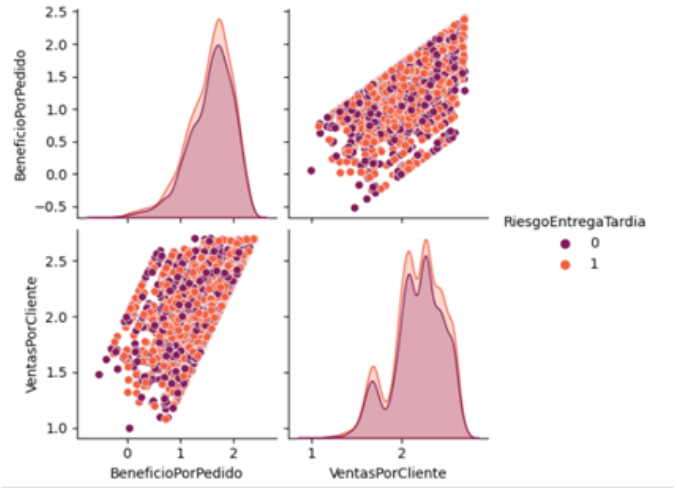
Es una extensión de la regresión lineal que se usa comúnmente para clasificación binaria. Aplica una función logística (sigmoide) a la salida de una regresión lineal para producir probabilidades entre 0 y 1.

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.

Con lo anterior mediante la regresión logística podemos modelar el problema de clasificación binaria.

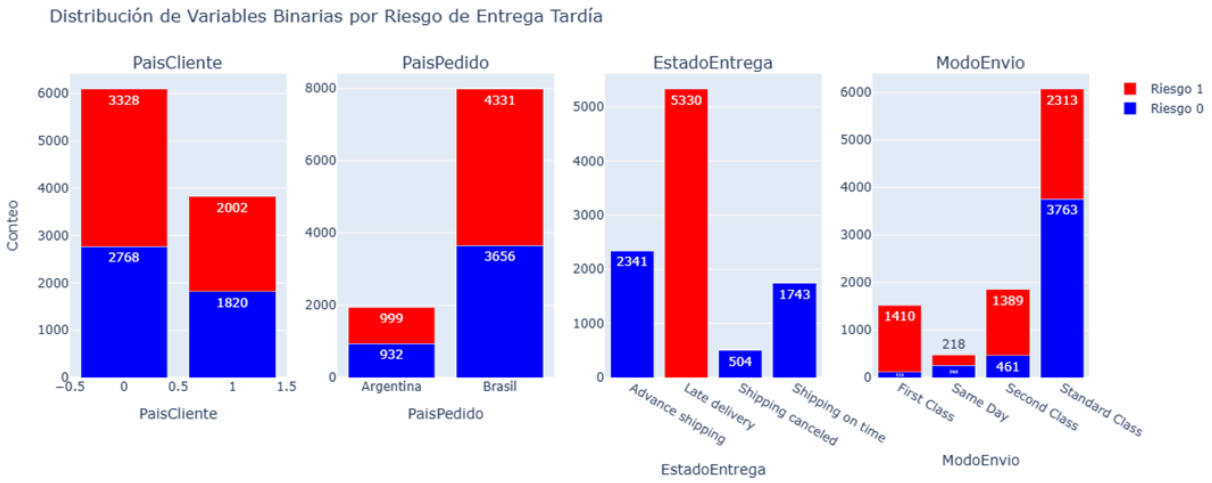


Aproximadamente el 45,17 % de las observaciones fueron marcadas como entrega a tiempo (riesgo) Aproximadamente el 54.83 % de las observaciones fueron marcadas como entrega tardía (riesgo)



Se grafican las variables continuas considerando la variable objetivo Para hacer comparables las variables continuas se obtiene el logaritmo base 10, esto también ayuda a tener más cercanos los datos

Ahora veamos la correlación que existe con las variables binarias



- A lo largo de los pedidos no hay variación significativa entre el porcentaje de riesgo positivo y negativo.

Previo a la implementación de la técnica de regresión logística se realizó un análisis exploratorio de los datos que nos permita tener un acercamiento a cómo se distribuyen con respecto a las variables de interés, adaptar valores numéricos, y convertir valores categóricos y por último graficar las correlaciones entre las distintas variables.

Desarrollo y Validación de la Regresión Logística

1- Creación de la variable X que contendrá las variables independientes: (**'Pago','EstadoEntrega','PaisCliente','Categoria','PaisPedido','SegmentoCliente','RegionPedido','DestinoPedido','CiudadCliente', 'ModoEnvio', 'Categoria', 'ModoEnvio'**)

Se seleccionan las características relevantes y se convierten en un arreglo NumPy que es fácil de procesar por algoritmos de aprendizaje automático.

```
array([[ 2,  1,  1, 29,  1,  2,  0,  4, 57,  2, 29,  2],
       [ 2,  1,  1, 23,  1,  2,  0, 45, 57,  2, 23,  2],
       [ 3,  1,  1, 23,  0,  0,  0,  5, 57,  3, 23,  3],
       [ 3,  3,  1, 23,  1,  0,  0, 45, 57,  3, 23,  3],
       [ 3,  0,  1, 29,  1,  0,  0, 45, 57,  3, 29,  3]], dtype=int64)
```

2- Se estandarizan las variables independientes, esto se hace por que variables con valores más altos pueden influir más en el modelo, de esta manera todas las variables están en términos similares
X = preprocessing.StandardScaler().fit(X).transform(X)

3- Se crea un array con la variable dependiente con los datos que luego vamos a querer predecir:

```
y = np.asarray(features['RiesgoEntregaTardia'])
y [0:5]
```

```
array([1, 1, 1, 0, 0], dtype=int64)
```

4- Separamos el conjunto de datos entre para Entrenamiento y Test (80-20)

```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2, random_state=4)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (7934, 12) (7934,)
Test set: (1984, 12) (1984,)
```

5- Se entrena la regresión

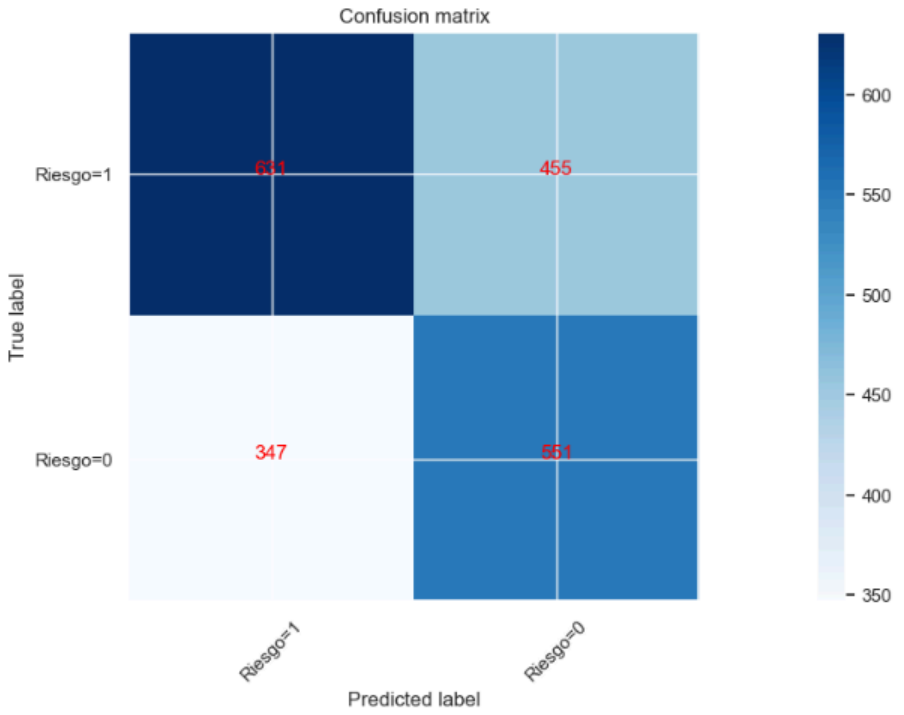
```
LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train)
LR
```

```
LogisticRegression
LogisticRegression(C=0.01, solver='liblinear')
```

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



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.

Jaccard Score

Otro Score a considerar es el Jaccard Score, y se puede obtener de la siguiente manera

```
jaccard_score(y_test, yhat,pos_label=0)
```

```
0.4072431633407243
```

Peso de las Variables en la Predicción

	Variable	Coeficiente
7	DestinoPedido	0.110000
8	CiudadCliente	0.090000
4	PaisPedido	0.020000
2	PaisCliente	0.010000
3	Categoria	0.000000
6	RegionPedido	0.000000
10	Categoria	0.000000
5	SegmentoCliente	-0.050000
0	Pago	-0.130000
1	EstadoEntrega	-0.470000
9	ModoEnvio	-0.520000
11	ModoEnvio	-0.520000

Como última validación del modelo, se revisa cuánto peso tienen las variables independientes en el resultado de la predicción

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**.

Seleccionadas las variables, comienza la etapa de “preprocesamiento”, en la que se busca preparar los datos para que sea más fácil y efectiva la implementación del modelo.

- Se convirtieron las variables categóricas a tipo 'category' y se crearon variables dummy.
- Se dividió el conjunto de datos en 80% de entrenamiento y 20% de prueba para entrenar y evaluar el modelo.

Se entrenó un modelo de Árbol de Decisión 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.

Parámetros iniciales del modelo

- Criterio de división: Entropía
- Profundidad máxima: No se delimitó (default:None).
- Muestras mínimas para dividir: 2.

El modelo se entrenó usando el conjunto de datos de entrenamiento y validación cruzada para verificar su estabilidad.

En cuanto al rendimiento, se evaluó utilizando métricas estándar como precisión, recall y f1-score, además de una matriz de confusión para identificar las clases donde el modelo podría mejorar.

Resultados del modelo inicial

- Precisión: 98.13%.
- Recall: Alto rendimiento para clases como “Entrega Tardía” (Late Delivery), y “Envío a tiempo” (Shipping on Time), pero más bajo para “Envío cancelado” (Shipping Canceled).

```
Precisión del modelo: 0.9813508064516129
                  precision    recall  f1-score   support

 Advance shipping       0.97      0.99      0.98       477
    Late delivery       1.00      1.00      1.00      1094
Shipping canceled       0.91      0.62      0.74        84
 Shipping on time       0.96      1.00      0.98       329

         accuracy                           0.98      1984
        macro avg       0.96      0.90      0.92      1984
     weighted avg       0.98      0.98      0.98      1984
```

Para mejorar el rendimiento del modelo, se realizó una ***búsqueda de hiperparámetros*** usando ‘GridSearchCV’, evaluando los siguientes parámetros:

- Criterio: Entropía y Gini.
- Profundidad máxima: Entre 5 y 20, incluyendo la opción sin límite (None).
- Muestras mínimas para dividir: Entre 2,5 y 10.

Los ***mejores hiperparámetros*** encontrados:

- Criterio: Entropìa.
- Profundidad máxima: 5.

- Muestras mínimas para dividir: 2.

Se entrenó un nuevo modelo utilizando los mejores hiperparámetros obtenidos. El resultado fue un modelo más generalizado, con menor riesgo de sobreajuste debido a la restricción en la profundidad del árbol.

Resultados del modelo optimizado

- Precisión: 98.39%
- Recall: Mejora notable en la clase “Envío cancelado”, con una mejora en el f1-score de esta clase.

```
Precisión del modelo mejorado: 0.9838709677419355
Reporte de Clasificación:
                    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
```

Se realizó validación cruzada para verificar la robustez del modelo en diferentes particiones de los datos, obteniendo una precisión promedio de 97.79%.

```
# Validación cruzada
cv_scores = cross_val_score(best_dt_model, X_train, y_train, cv=5)
print("Precisión promedio en Validación Cruzada:", cv_scores.mean())
```

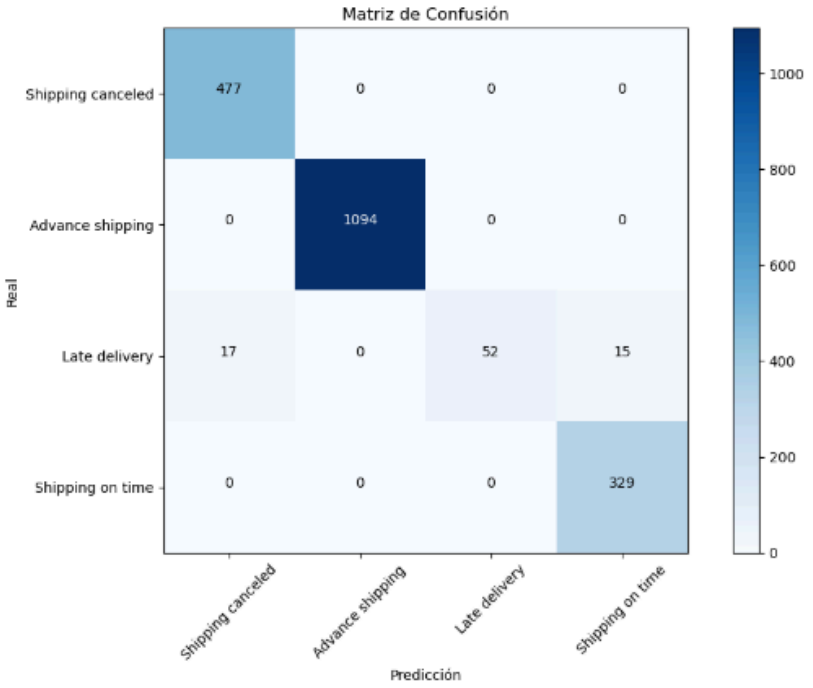
```
Precisión promedio en Validación Cruzada: 0.9779425518339027
```

El modelo se evaluó utilizando varias métricas para tener una visión integral de su rendimiento:

- Precisión: 98.39%.
- f1-score: 0.98 (promedio ponderado).

La matriz de confusión mostró 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



Resultados

El modelo de árboles de decisión ha demostrado ser una herramienta efectiva para predecir el estado de entrega de los pedidos, lo que facilita su adopción por parte de las áreas operativas de la empresa. Los resultados permiten prever con una **precisión superior al 98% si un pedido será entregado a tiempo**, lo cual es **clave para la optimización de rutas (logística de transporte) y gestión de inventarios**; contribuyendo a la reducción de costos y mejorando la satisfacción del cliente.

Aunque el modelo presenta un rendimiento excelente en general, tiene **limitaciones** como:

- La dificultad para predecir con precisión clases minoritarias, aunque el tuning mejoró el rendimiento.
- Datos futuros no incluidos en el entrenamiento pueden presentar escenarios no vistos, lo que podría reducir la precisión.

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, Jerárquicol y DBSCAN.

Tratamiento previo de los datos

Previo a la utilización de las técnicas de agrupamiento, se realizó un análisis exploratorio de los datos que nos permita tener un acercamiento a cómo se distribuyen con respecto a las variables de interés, adaptar valores numéricos, y convertir valores categóricos y por último graficar las correlaciones entre las distintas variables.
La implementación de técnicas de Agrupamiento nos permitirá ampliar esta información.

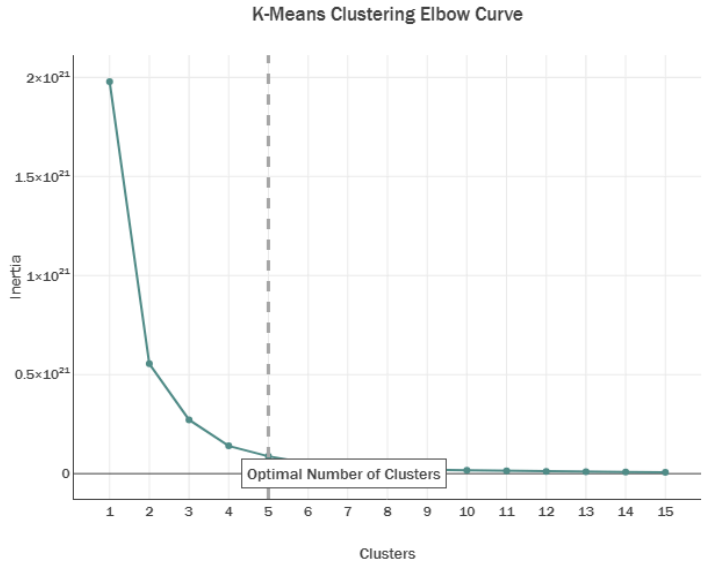
Agrupamiento de K-Means

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.

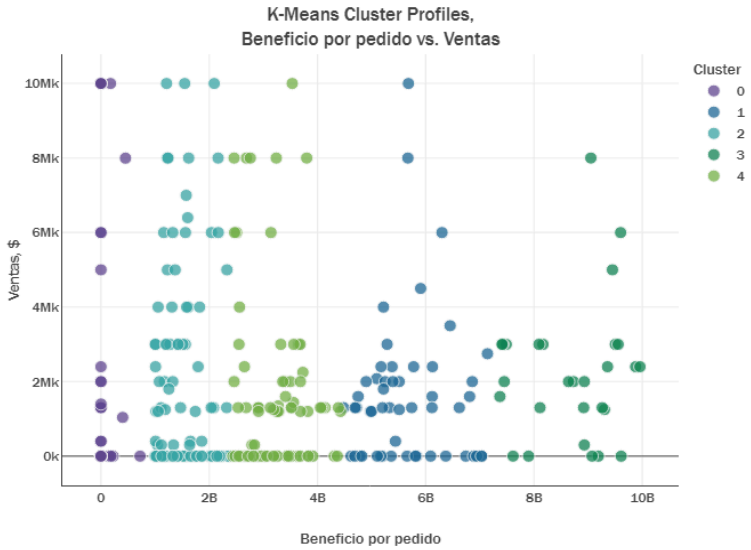
2-La inercia mide la similitud de los grupos calculando la distancia total entre los puntos de datos y su centro de grupo más cercano.

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



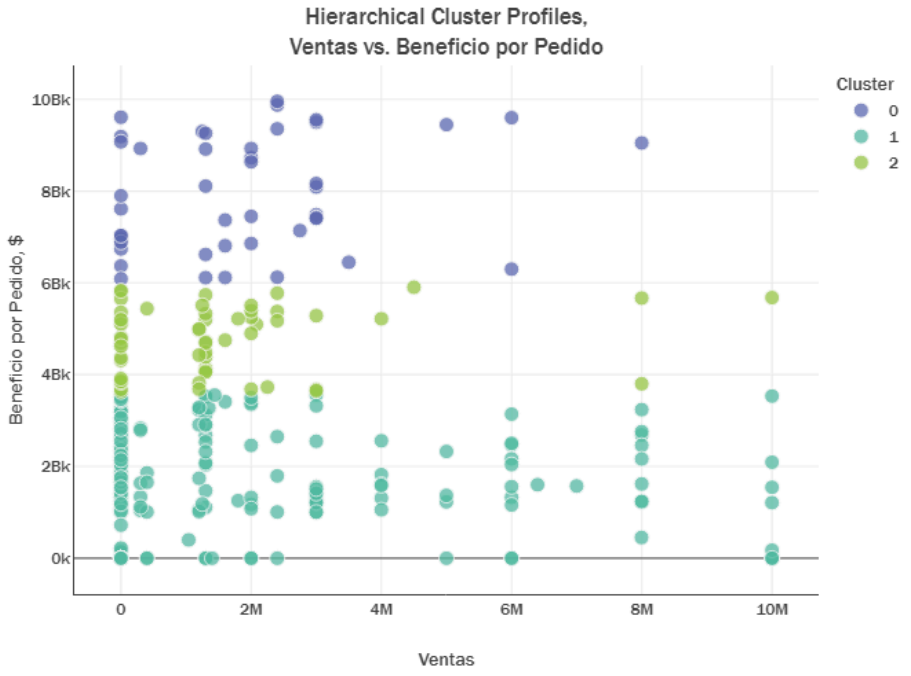
El modelo K-Means segmenta los datos en distintos grupos en función del Beneficio por pedido y las ventas.

- 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.

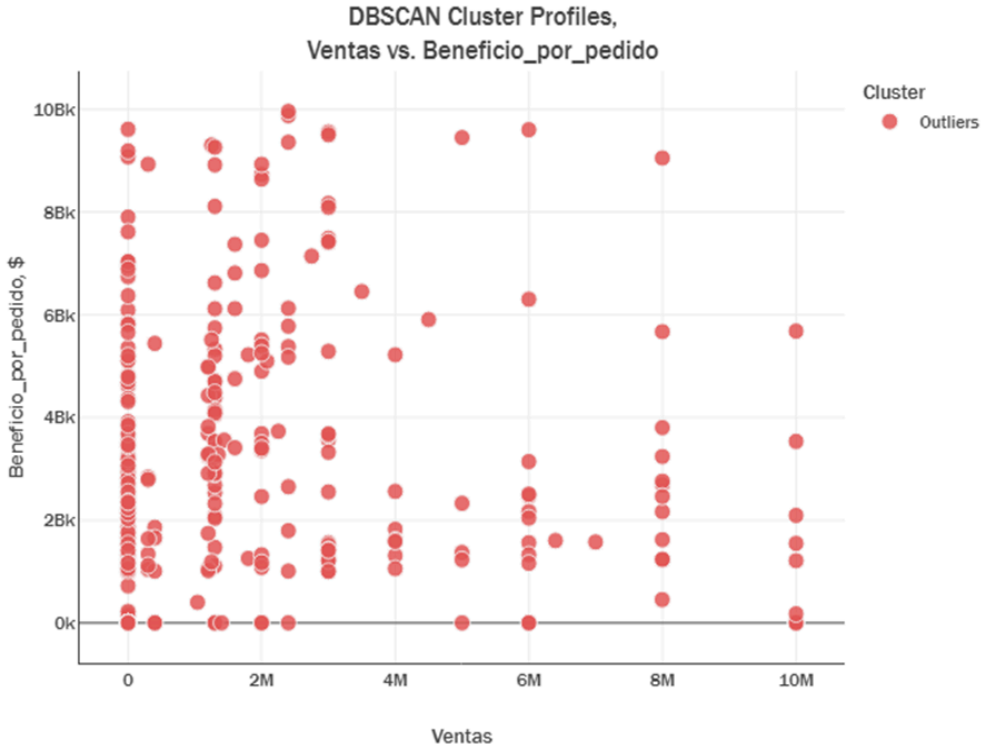


- En el centro del gráfico se encuentra el grupo 2 con los beneficios en rangos de 4 y 6 BK
- El grupo 0 en la parte superior con los valores mayores y el grupo 1 en la parte inferior con los valores menores

DBSCAN

La técnica de agrupación en clústeres espaciales basada en la densidad de aplicaciones con ruido (DBSCAN).

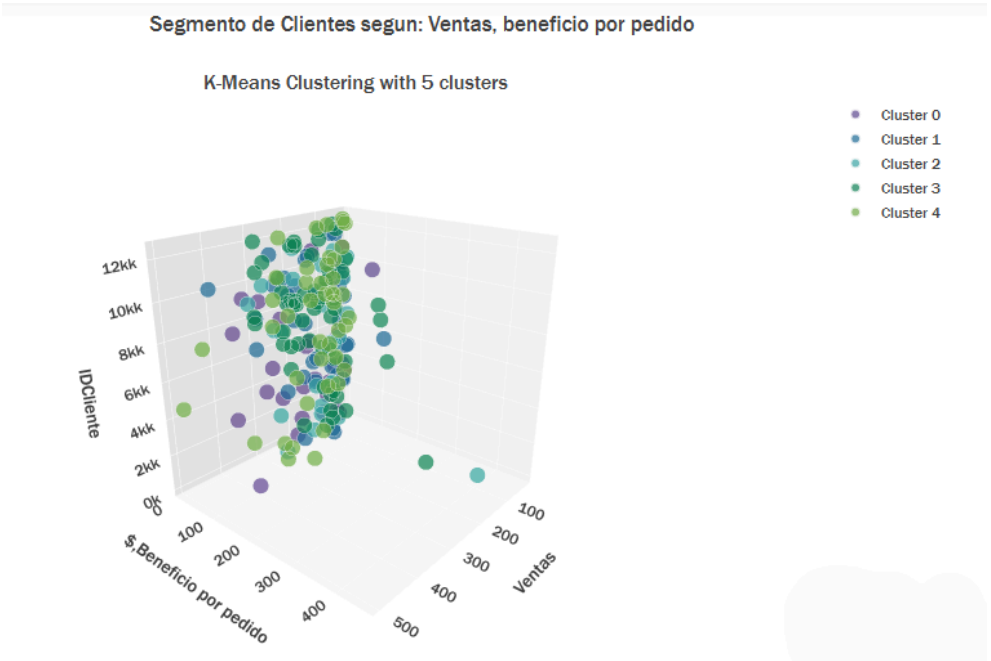
DBSCAN segmenta los datos en función de la densidad de las observaciones, donde las áreas de alta densidad se separan de las áreas de baja densidad.
El modelo también puede identificar clústeres de formas únicas y detectar valores atípicos dentro de los datos, aunque es sensible a las densidades variables de las observaciones.

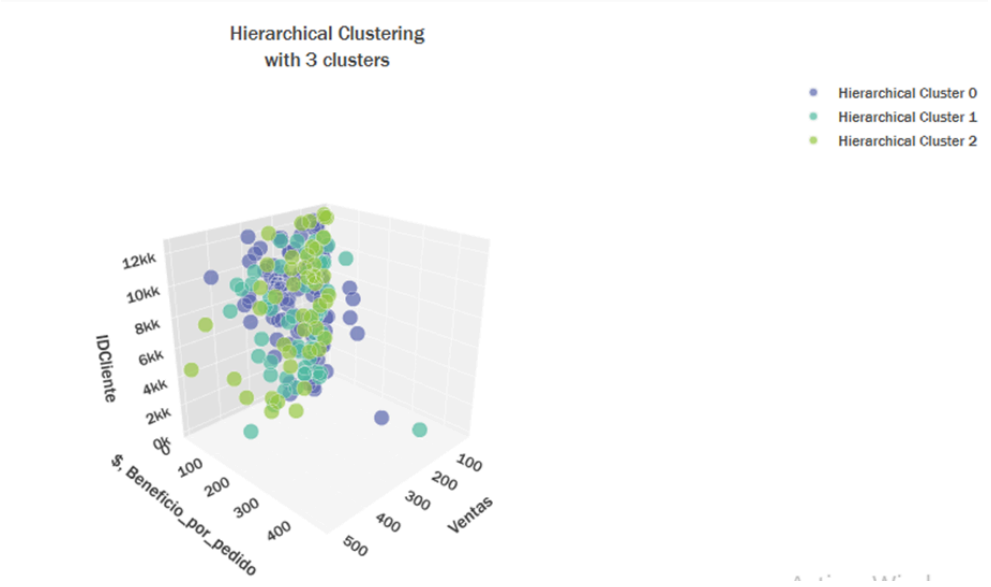


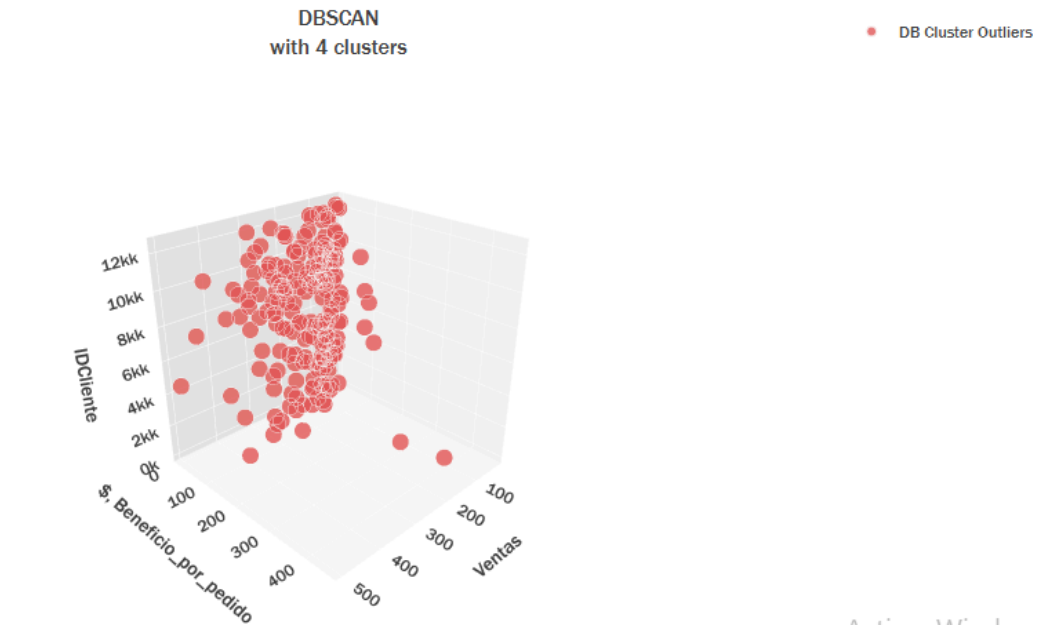
Estos segmentos se asemejan a los clústeres del modelo K-Means, con los valores atípicos identificados en rojo. En general, hay bastantes valores atípicos en el gráfico, lo que probablemente se deba a la variación en las densidades de los clústeres.

Comparación de modelos

A continuación se muestran los gráficos de los perfiles de los clientes en función de ventas y beneficios de cada modelo de agrupación.




Hierarchical Clustering
with 3 clusters
Hierarchical Cluster 0
Hierarchical Cluster 1
Hierarchical Cluster 2
IDCliente
12kk
10kk
8kk
6kk
4kk
2kk
0k
$, Beneficio_por_pedido
100
200
300
400
500
Ventas
100
200
300
400


DBSCAN
with 4 clusters
DB Cluster Outliers
IDCliente
12kk
10kk
8kk
6kk
4kk
2kk
0k
$, Beneficio_por_pedido
100
200
300
400
500
Ventas
100
200
300
400