

Proyecto Final Diplomatura Universitaria en Ciencia de Datos

Horacio Antonio Zini

12 de abril de 2025

Resumen

Este informe presenta un análisis exploratorio y predictivo del dataset de ventas. El estudio incluye el preprocesamiento de datos, visualización de patrones y construcción de modelos predictivos utilizando técnicas de aprendizaje profundo.

1. Introducción

El objetivo de este trabajo es analizar un conjunto de datos de ventas proporcionado por una tienda minorista que contiene variables demográficas de los clientes, características de los productos y el importe total de compra registrado por cada cliente durante el mes considerado.

A partir de los datos, se busca desarrollar un modelo predictivo que permita estimar el monto de compra de los clientes en función de los distintos productos. Esta herramienta resultará clave para diseñar ofertas personalizadas, optimizando así la experiencia del cliente y fortaleciendo las estrategias comerciales.

1.1. Dataset

Las variables del conjunto de datos se pueden dividir en dos secciones principales:

- Información sobre el cliente
 - User ID: identificación del usuario.
 - Gender: sexo del usuario.
 - Age: edad en intervalos.
 - Occupation: ocupación (enmascarada).
 - Marital Status: estado civil.
 - City Category: categoría de la ciudad (A,B,C).
 - Stay In Current City Years: número de años que lleva en la ciudad actual.
- Información sobre el producto

- Product ID: identificación del producto.
- Product Category 1: categoría 1 del producto.
- Product Category 2: categoría 2 del producto.
- Product Category 3: categoría 3 del producto.
- Purchase: importe total de la compra del mes pasado.

La variable purchase es la variable que se quiere predecir (target).

2. Preprocesamiento

2.1. Análisis exploratorio inicial

Dimensiones del dataset:

- Número de observaciones: 550068
- Número de variables: 12

Sólo las columnas Product Category 2 y Product Category 3 tienen valores nulos, 173.638 y 383.247, lo que representa un 32 % y un 70 % de valores nulos sobre el total de los datos, respectivamente.

Cuadro 1: Tipos de columnas y cantidad de valores únicos.

Columna	Tipo	Valores únicos
User ID	int64	5891
Product ID	object	3631
Gender	object	2
Age	object	7
Occupation	int64	21
City Category	object	3
Stay In Current City	object	5
Marital Status	int64	2
Product Category 1	int64	20
Product Category 2	float64	17
Product Category 3	float64	15
Purchase	int64	18105

2.2. Transformación de variables

En primer lugar se analizaron las variables categóricas. A continuación, se presenta el proceso que se llevó a cabo para la codificación de variables categóricas a valores numéricos.

Para la variable Purchase, se definieron grupos de categorización en base a los cuartiles.

Cuadro 2: Codificación de variables categóricas a numéricas.

Variable	Categoría	Valor codificado
Gender	F	0
	M	1
Age	0–17	0
	18–25	1
	26–35	2
	36–45	3
	46–50	4
	51–55	5
	55+	6
City Category	A	0
	B	1
	C	2
Stay in Current City	0	0
	1	1
	2	2
	3	3
	4+	4

2.3. Tratamiento de valores faltantes

Para iniciar el proceso de imputación de valores nulos, en primer lugar se analizó gráficamente la distribución de las columnas Product Category 2 y Product Category 3. Dado que las columnas no tienen una distribución normal, se realizó el proceso de imputar los valores nulos respecto a la mediana.

3. Modelado predictivo

3.1. División del dataset

El conjunto de datos fue dividido en tres subconjuntos utilizando la función *train-test-split* de la librería *scikit-learn*: entrenamiento (440,054 muestras), validación (55,007 muestras) y prueba (55,007 muestras), con el objetivo de entrenar, ajustar y evaluar el desempeño del modelo de manera adecuada.

3.2. Elección de métrica

Para evaluar el desempeño de los modelos, se optó por la métrica *F1 score* debido a que proporciona un equilibrio entre la precisión y la recuperación (recall), dos métricas fundamentales en la evaluación de modelos de clasificación multiclase.

3.3. Aprendizaje automático

Se optó por aplicar inicialmente un modelo de aprendizaje automático debido a su simplicidad, menor demanda computacional y facilidad de interpretación. En particular, se utilizó Regresión Logística como modelo base para establecer un punto de comparación con métodos más complejos como los modelos de Deep Learning.

3.4. Aprendizaje Profundo

Se entrenó el modelo *MultilayerPerceptron* (MLP) con una arquitectura compuesta por cuatro capas completamente conectadas. La red recibe como entrada un vector de dimensión igual al número de variables predictoras y produce una salida con tantas unidades como clases presentes en el conjunto de entrenamiento. La arquitectura incluye capas lineales con 256, 128 y 64 neuronas respectivamente, cada una seguida por una función de activación *ReLU*, y una capa final lineal sin activación. El modelo fue implementado utilizando PyTorch y entrenado en el dispositivo disponible (CPU o GPU).

Para el entrenamiento, se utilizó la función de pérdida *CrossEntropyLoss*, ponderada según la distribución de clases, y el optimizador *SGD* con una tasa de aprendizaje de 0.01 y una penalización L2 (parámetro weight decay) de $1e - 4$.

4. Resultados

A continuación se detallan los resultados obtenidos con los modelos anteriormente presentados. Los valores corresponden a los datos de validación.

Cuadro 3: Regresión Logística.

Métrica	Valor
Accuracy	0.3889
Precisión	0.3639
Recall	0.3889
F1-score	0.3660

Cuadro 4: Desempeño del modelo MLP.

Métrica	Valor
Epoch final	39
Pérdida (Train / Val)	0.983 / 0.973
F1-score (Train / Val)	0.500 / 0.491

Clase real \ Clase predicha	0	1	2	3
0	10645	1346	291	1473
1	6141	5601	487	1530
2	3132	3727	1142	5744
3	252	21	364	13111

Cuadro 5: Matriz de confusión con colores

5. Conclusiones

Al analizar la matriz de confusión del modelo, se pueden realizar algunas observaciones, mientras que el modelo realiza buenas predicciones para las clases extremas, para las clases intermedias se presenta una alta confusión con otras clases, lo que indica que el modelo tiene dificultades para distinguir montos medios. Estas confusiones ocurren principalmente con clases vecinas. Por ejemplo, la clase 2 se predice como 1 en 3,727 casos y como 3 en 5,744 casos. Esto sugiere que la representación de los montos medios no está bien diferenciada en el espacio de características.

Teniendo en cuenta que la discretización de la variable continua Purchase se realizó definiendo los límites entre las clases utilizando los cuartiles, las categorías de la variable Purchase y sus rangos quedaron de la siguiente manera:

- Bajo (0): 12 a 5.823
- Medio-bajo (1): 5.823 a 8.047
- Medio-alto (2): 8.047 a 12.054
- Alto (3): 12.054 a 23.961

Al observar el tamaño de los intervalos, se nota que las clases intermedias son más pequeñas que las categorías extremas, que abarcan rango mayores. Además, las diferencias entre los valores en las clases 1 y 2 son relativamente pequeños, lo que podría estar dificultando la diferenciación entre ellas durante el entrenamiento del modelo.

Esto puede explicar la alta confusión en la clasificación de las clases intermedias, ya que los valores en estos rangos están muy próximos y no presentan una separación clara en el espacio de características del modelo. También explica el buen desempeño del modelo para clasificar elementos de la categoría 3, ya que esta es la clase con el rango más amplio.

Es importante notar también que la categorización de la variable objetivo se realizó asignando etiquetas numéricas de menor a mayor según el monto de compra.

Dado que estos valores representan niveles de compra en una escala lineal sin considerar la diferencia de escala entre los rangos de las clases, el modelo aprende que existe un orden inherente entre las clases pero no necesariamente aprende que la diferencia entre 1 y 2 es menor que la diferencia entre 2 y 3.

Esta falta de información sobre las escalas de las clases puede ser una de las razones por las que el modelo tiene dificultades en la distinción de los montos intermedios, y explicaría la confusión observada en la matriz de confusión.