

CAMILA PICHARDO
ALEJANDRO SOTO
CARLO HENRIQUEZ

CODIGO PYTHON



LIBRERIAS

```
9 # =====
10
11 import pandas as pd
12 import numpy as np
13 import matplotlib.pyplot as plt
14 import seaborn as sns
15
16 from sklearn.preprocessing import StandardScaler
17 from sklearn.cluster import KMeans
18 from sklearn.metrics import silhouette_score
19
20 # -----
```

PANDAS

Con pandas primero hacemos que lea el archivo con “pd.read_csv()”, luego mostramos la forma de los datos con “df.shape”, detectamos el tipo de datos con “df.types” y luego detectamos los datos faltantes con “df.isna().sum()” .

Después con las variables de ingresos mensuales y el precio de la membresía, usamos pandas para saber la media, mediana y moda de estos datos.

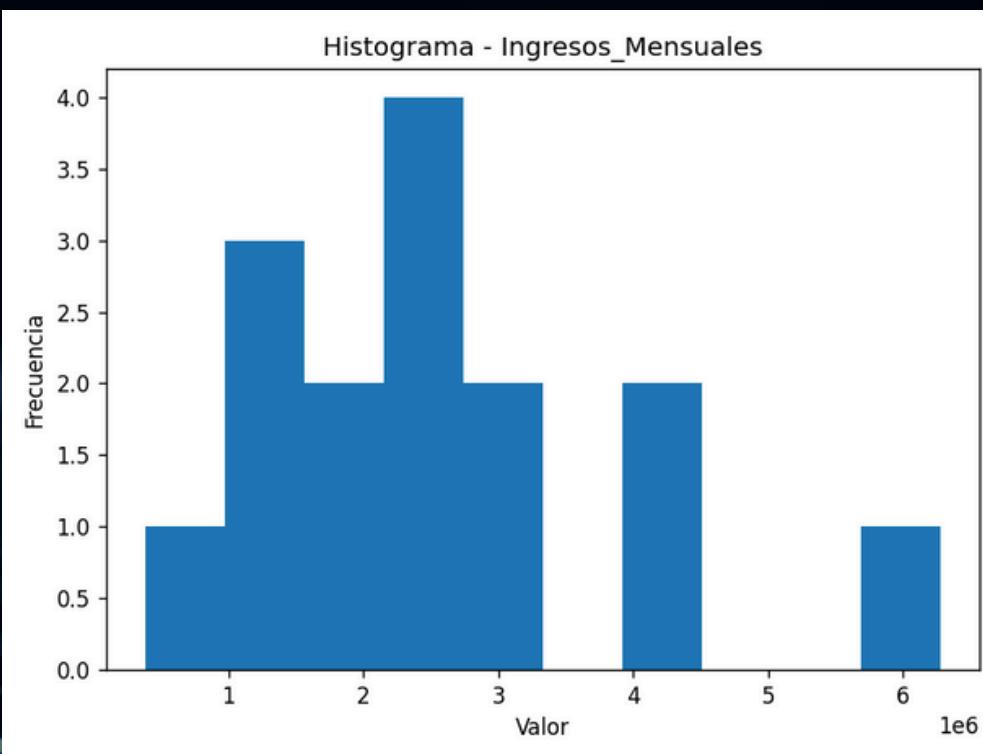
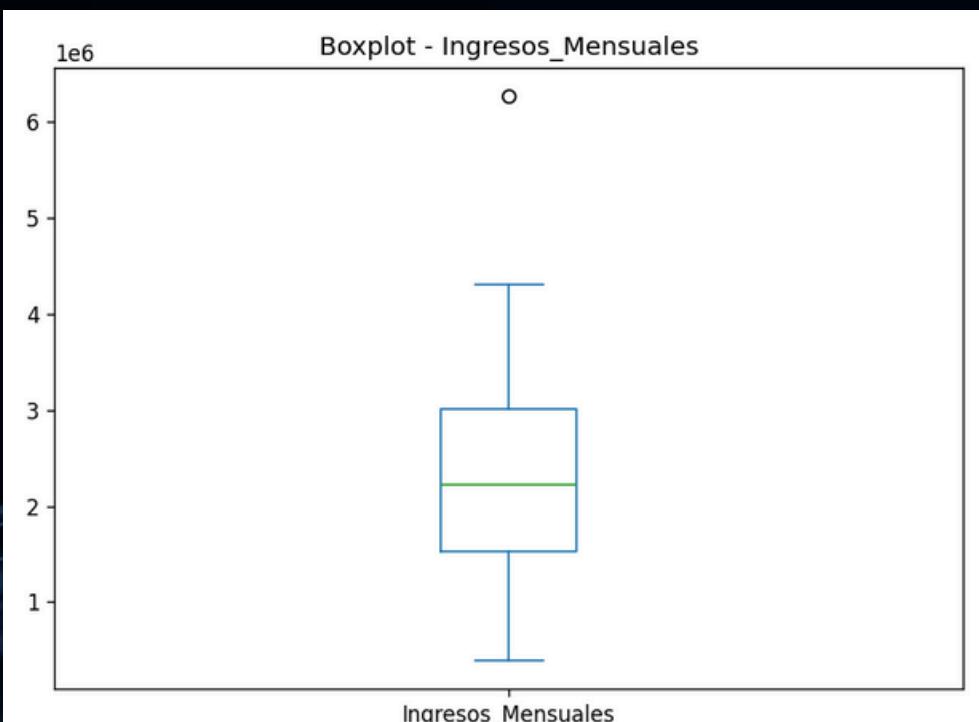
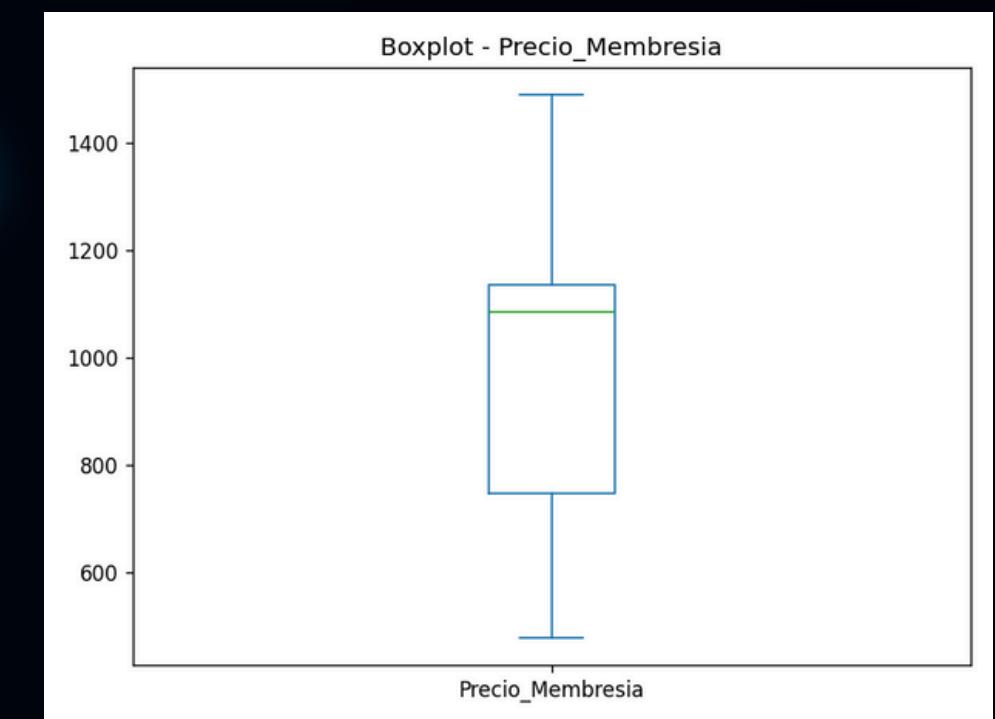
```
# ---  
# df = pd.read_csv("escenario_gimnasios_15.csv", encoding="utf-8-sig")  
print("Forma (filas, columnas):", df.shape)  
print("\nTipos de datos:\n", df.dtypes)  
print("\nValores faltantes por columna:\n", df.isna().sum())  
  
# ---  
# 2) Estadística descriptiva  
# ---  
desc = df.select_dtypes("number").describe()  
print("\nDescripción numérica:\n", desc)  
  
var1 = "Ingresos_Mensuales"  
var2 = "Precio_Membresia"  
  
print("\n==== Indicadores de tendencia central ===")  
print(f"{var1} -> media: {df[var1].mean(skipna=True):.2f}, "  
      f"mediana: {df[var1].median(skipna=True):.2f}, "  
      f"moda: {df[var1].mode(dropna=True).iloc[0] if not df[var1].mode(dropna=True).empty else None}")  
print(f"{var2} -> media: {df[var2].mean(skipna=True):.2f}, "  
      f"mediana: {df[var2].median(skipna=True):.2f}, "  
      f"moda: {df[var2].mode(dropna=True).iloc[0] if not df[var2].mode(dropna=True).empty else None}")  
  
Forma (filas, columnas): (15, 14)  
Tipos de datos:  
Gimnasio          object  
Zona              object  
Metros_Cuadrados int64  
Entrenadores      int64  
Trafico_Mensual  float64  
Precio_Membresia  float64  
Descuento_Promedio float64  
Gasto_Marketing   float64  
Satisfaccion      float64  
Equipos           int64  
Churn              float64  
Tasa_Conversion   float64  
Clases_Semana     int64  
Ingresos_Mensuales int64  
dtype: object  
  
Valores faltantes por columna:  
Gimnasio          0  
Zona              0  
Metros_Cuadrados 0  
Entrenadores      0  
Trafico_Mensual   1  
Precio_Membresia  1  
Descuento_Promedio 1  
Gasto_Marketing   1  
Satisfaccion      1  
Equipos           0  
Churn              0  
Tasa_Conversion   0  
Clases_Semana     0  
Ingresos_Mensuales 0  
dtype: int64  
  
[8 rows x 12 columns]  
  
==== Indicadores de tendencia central ====  
Ingresos_Mensuales -> media: 2508981.00, mediana: 2225298.00, moda: 386388  
Precio_Membresia -> media: 1010.34, mediana: 1085.19, moda: 476.82
```

VISUALIZACIÓN

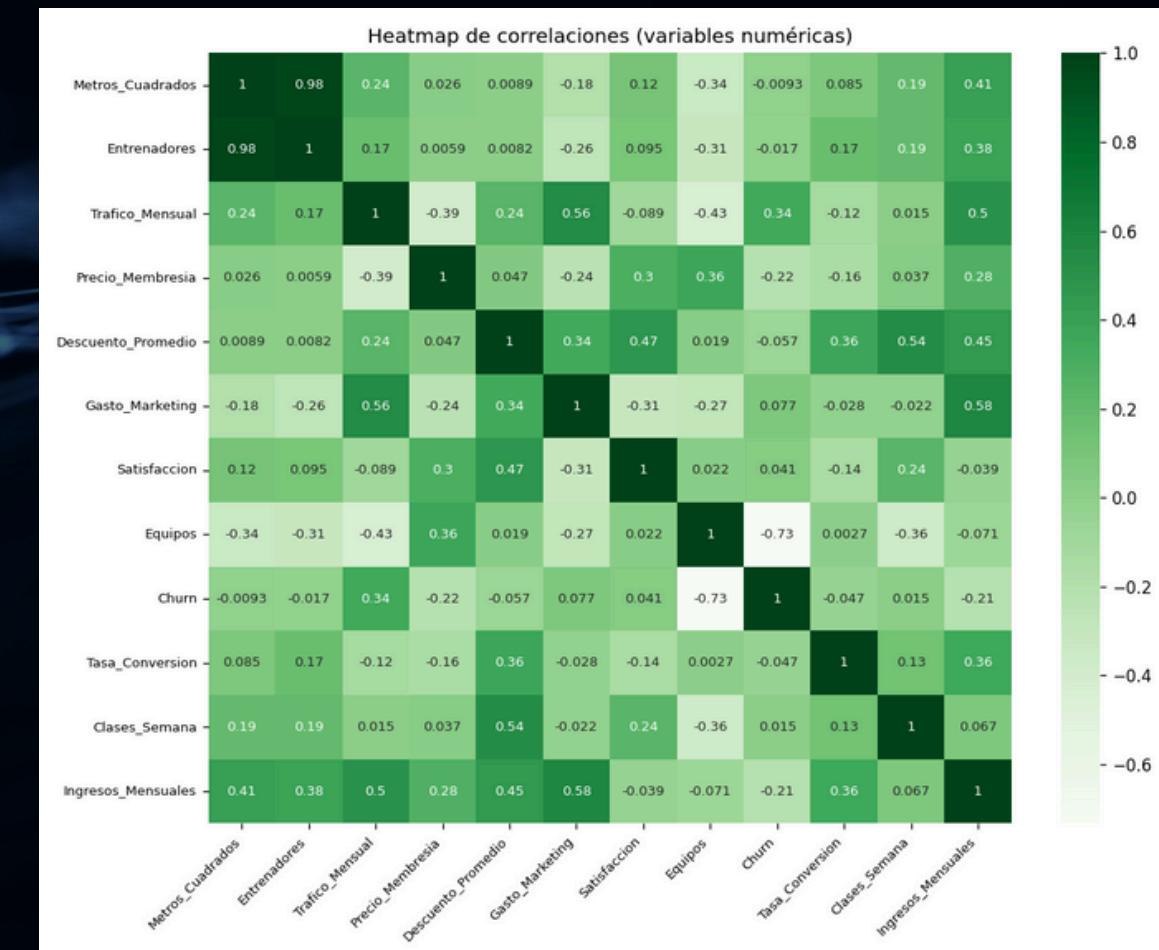
BOXPLOT
INGRESOS
MENSUALES

HISTOGRAMA

BOXPLOT
MEMBRESIA



HEATMAP



CLUSTERING

```
# -----
features_all = df.select_dtypes("number").copy()
features_all = features_all.fillna(features_all.mean(numeric_only=True))

# Eliminar variables altamente correlacionadas
high_corr_threshold = 0.90
corr_matrix = features_all.corr()
to_drop = set()
cols = corr_matrix.columns.tolist()

for i, c1 in enumerate(cols):
    for c2 in cols[i + 1:]:
        r = corr_matrix.loc[c1, c2]
        if pd.notna(r) and abs(r) >= high_corr_threshold:
            to_drop.add(c2)

X = features_all.drop(columns=list(to_drop)) if to_drop else features_all.copy()

print("\n==> Selección de variables para clustering ==")
print("Columns usadas:", list(X.columns))
if to_drop:
    print("Eliminadas por alta correlación (|r|>=0.90):", list(to_drop))
```

- Se seleccionan sólo las variables numéricas para el análisis.
- Se eliminan los valores faltantes reemplazándolos con la media de cada variable.
- Se calcula la matriz de correlación para esas variables.
- Se eliminan aquellas variables que tienen una correlación muy alta ($|r| \geq 0.90$) para evitar redundancia, esto asegura que el modelo trabaje con variables relevantes y no repetitivas.

KMEANS

- Primero, se normalizan los datos para que todas las variables tengan la misma escala. Esto es importante porque si una variable tiene números muy grandes y otra muy pequeños, el algoritmo podría darle más importancia a la primera sin razón.

- Luego, se prueban diferentes números de grupos (k) para ver cuál es el mejor. Se hace esto con dos métodos:

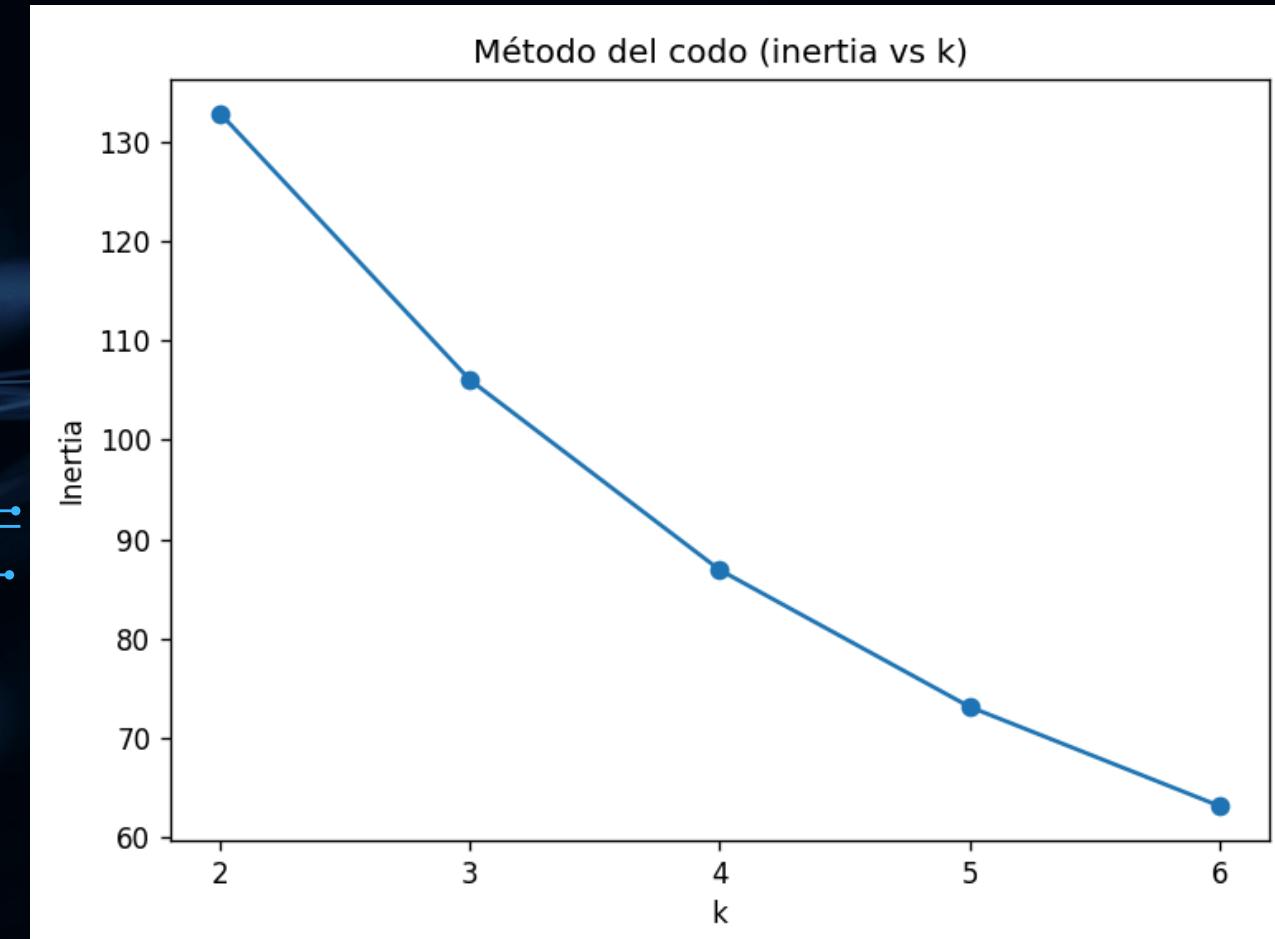
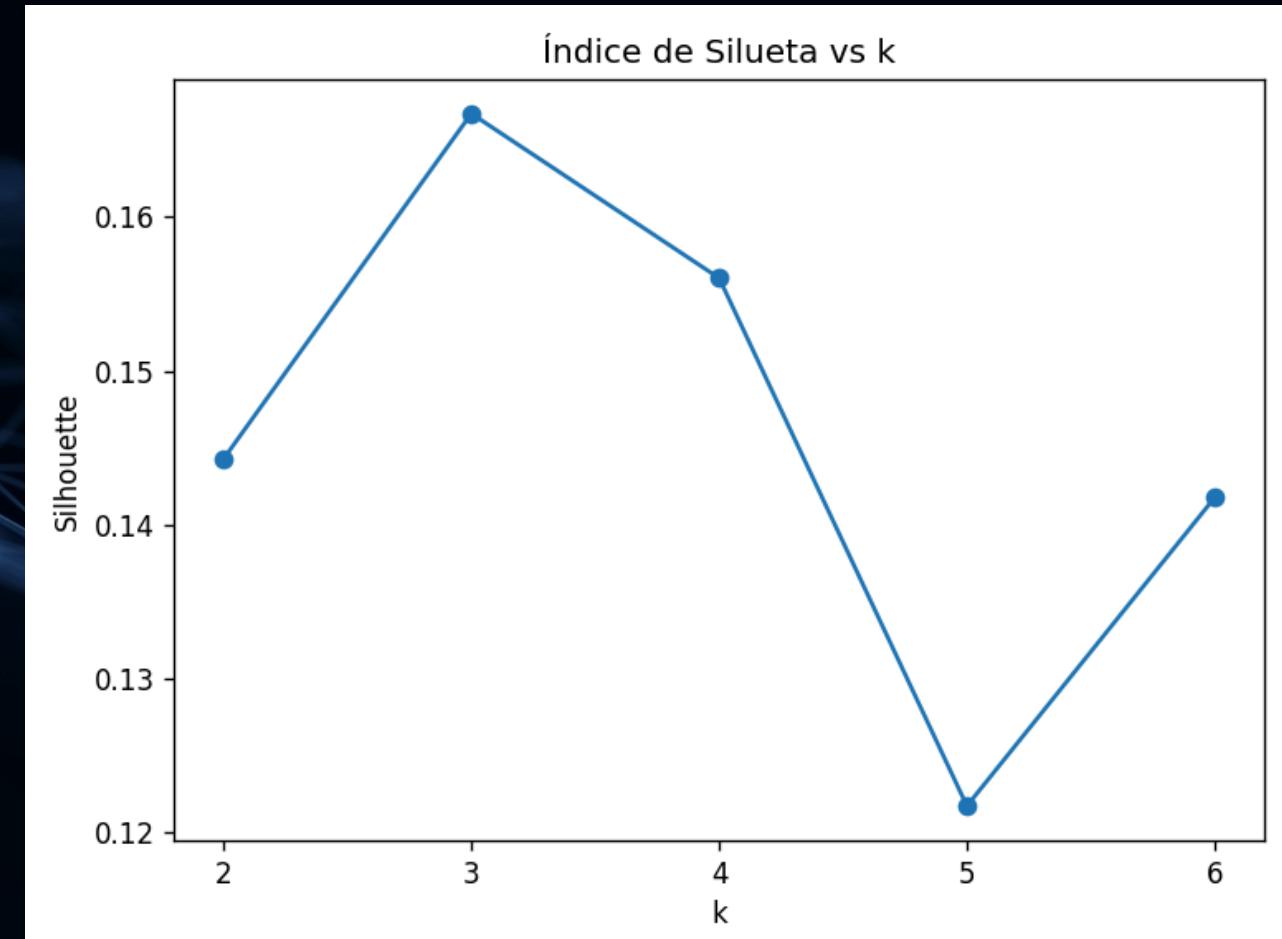
Método del

codo: mira qué tan bien los datos se agrupan para cada número de grupos y busca el punto donde mejorar mucho más al aumentar k ya no vale tanto la pena.

Índice de silueta: mide qué tan separados y compactos están los grupos; valores altos significan grupos bien definidos.

- Se elige el número de grupos que tenga el mejor índice de silueta (el que mejor separa los datos).
- Finalmente, se entrena el algoritmo K-Means con ese número de grupos para asignar cada gimnasio a un cluster.

```
#  
scaler = StandardScaler()  
Xs = scaler.fit_transform(X)  
  
k_values = range(2, 7)  
inertias = []  
sil_scores = []  
  
for k in k_values:  
    km = KMeans(n_clusters=k, random_state=42, n_init=10)  
    labels = km.fit_predict(Xs)  
    inertias.append(km.inertia_)  
    sil = silhouette_score(Xs, labels)  
    sil_scores.append(sil)
```



RESULTADOS

```
==== Centros de K-Means (escala original) ====
   Metros_Cuadrados  Trafico_Mensual ... Clases_Semana Ingresos_Mensuales
Centro_0           1713.75      36373.00 ...      48.75        2722544.75
Centro_1           1297.00      18277.55 ...      48.00        2035250.78
Centro_2           1565.00      29279.50 ...     102.00        4213639.50
[3 rows x 11 columns]

==== Distancias entre centros (escala original) ====
    Centro_0  Centro_1  Centro_2
Centro_0      0.00  687827.70 1491112.22
Centro_1  687827.70      0.00 2178521.84
Centro_2 1491112.22  2178521.84      0.00
```

- Cluster 0 – Gimnasios grandes con precios accesibles y fuerte marketing: gran tamaño y tráfico alto, precio relativamente bajo, invierten mucho en marketing, nivel de satisfacción medio, ingresos altos por volumen.
Segmento masivo, atraen clientes por visibilidad y precio.
- Cluster 1 – Gimnasios compactos, premium y con baja rotación: menor tamaño y tráfico, precio más alto, muchos equipos disponibles, baja tasa de abandono, marketing bajo.
Segmento premium/eficiente, clientes fieles aunque en menor cantidad.
- Cluster 2 – Gimnasios grandes, premium y enfocados en clases: espacios amplios y buena afluencia, Precio alto.
 - Ofrecen muchos descuentos.
 - Mayor satisfacción del cliente.
 - Altísima oferta de clases semanales.
 - Ingresos más altos de todos los clusters.
 - → Segmento premium dinámico, enfocados en clases grupales y experiencia.

PREGUNTAS

- ¿Crees que los centros obtenidos son representativos de los datos? ¿Por qué?
Sí, los centros obtenidos son representativos porque resumen las características principales de cada segmento de gimnasios. Al aplicar K-Means y verificar con el índice de silueta, se comprobó que los clusters tienen una separación razonable



- ¿CÓMO DETERMINASTE EL VALOR DE K?
OBSERVAMOS DONDE SE MUESTRAN LOS PUNTO DONDE LA INERCIA DEJA DE DECRECER SIGNIFICATIVAMENTE.
- ¿QUÉ OCURRIRÍA SI USARAS UN VALOR MÁS ALTO O MÁS BAJO DE K?
SI AUMENTAS K:
LOS CLUSTERS SE VUELVEN MÁS PEQUEÑOS Y ESPECÍFICOS.
SI REDUCES K:
LOS CLUSTERS SON MÁS GENERALES Y FÁCILES DE INTERPRETAR.

- ¿QUÉ PASARÍA CON LOS CENTROS SI HUBIERA MUCHOS OUTLIERS (SEGÚN LOS BOXPLOTS)?
LOS OUTLIERS ARRASTRAN LOS CENTROIDES, PROVOCANDO QUE REPRESENTEN PEOR AL GRUPO.
- ¿QUÉ ESTRATEGIAS DE NEGOCIO PROPONDRÍAS PARA LOS DIFERENTES SEGMENTOS DE GIMNASIOS?
CON BASE EN LAS VARIABLES CLAVE (INGRESOS, PRECIO DE MEMBRESÍA, TRÁFICO MENSUAL, GASTO EN MARKETING Y SATISFACCIÓN), SE PUEDEN DEFINIR TRES GRANDES SEGMENTOS:
GIMNASIOS PREMIUM (ALTA MEMBRESÍA, ALTOS INGRESOS, MUCHOS ENTRENADORES Y EQUIPOS)
GIMNASIOS ESTÁNDAR (PRECIOS MEDIOS, INGRESOS SÓLIDOS, BUEN EQUILIBRIO EN TRÁFICO Y EQUIPOS)
GIMNASIOS ACCESIBLES (MEMBRESÍA BAJA, ALTO TRÁFICO, INGRESOS MODERADOS, MARKETING FUERTE)