

Nombre: Aldo Tena García

Matrícula: A01275222

```
# Importamos las librerías que usaremos
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

import plotly.express as px

# Carga el conjunto de datos al ambiente de Google Colab y muestra los primeros
# 6 renglones.
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
    print('User uploaded file "{name}" with length {length} bytes'.format(
        name=fn, length=len(uploaded[fn])))

Elegir archivos bestsellers...egories.csv
• bestsellers with categories.csv(text/csv) - 51161 bytes, last modified: 8/5/2022 - 100% done
Saving bestsellers with categories.csv to bestsellers with categories (1).csv
User uploaded file "bestsellers with categories.csv" with length 51161 bytes

df = pd.read_csv('bestsellers with categories.csv')
df.head(6)
```

	Name	Author	User Rating	Reviews	Price
0	10-Day Green Smoothie Cleanse	JJ Smith	4.7	17350	8
1	11/22/63: A Novel	Stephen King	4.6	2052	22
2	12 Rules for Life: An Antidote to Chaos	Jordan B. Peterson	4.7	18979	15
3	1984 (Signet Classics)	George Orwell	4.7	21424	6
.	5.000 Awesome Facts (About Everything!)	National Geographic	4.8	11000	10

Name: Nombre del libro.

Author: Autor.

User Rating: Calificación promedio que los usuarios asignaron al libro (1-5).

Reviews: Número de reseñas.

Price: Precio del libro.

Year: Año de publicación.

Genre: Género literario (ficción/no ficción).

▼ Analisis estadistico

#Verifica la cantidad de datos que tienes, las variables que contiene cada vector de datos e
`len(df)`

```
550
```

`df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550 entries, 0 to 549
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Name            550 non-null   object
1   Author          550 non-null   object
2   User Rating     550 non-null   float64
3   Reviews         550 non-null   int64
4   Price           550 non-null   int64
5   Year            550 non-null   int64
6   Genre           550 non-null   object
dtypes: float64(1), int64(3), object(3)
memory usage: 30.2+ KB
```

`df.describe()`

	User Rating	Reviews	Price	Year
count	550.000000	550.000000	550.000000	550.000000
mean	4.618364	11953.281818	13.100000	2014.000000
std	0.226980	11731.132017	10.842262	3.165156
min	3.300000	37.000000	0.000000	2009.000000
25%	4.500000	4058.000000	7.000000	2011.000000
50%	4.700000	8580.000000	11.000000	2014.000000
75%	4.800000	17253.250000	16.000000	2017.000000
max	4.900000	87844.000000	105.000000	2019.000000

Antes de llevar a cabo análisis más profundos con los datos que fueron entregados se puede apreciar desde la descripción de los mismos que la mayor parte fueron escritos alrededor del año 2014, con un precio de 13 y una calificación de 4.6 como promedio. Los libros que se incluyen dentro de esta base de datos ya han estado una buena cantidad de tiempo en el mercado, esto repercute directamente en la cantidad de reviews que estos tienen y en su user rating respectivamente, no se tiene claro si los precios han cambiado con el tiempo o si se han mantenido estáticos.

```
#Calcula la correlación de las variables que consideres relevantes.
dfwy = df.drop(columns='Year')
dfwyc = dfwy.corr()
dfwyc
```

	User Rating	Reviews	Price
User Rating	1.000000	-0.001729	-0.133086
Reviews	-0.001729	1.000000	-0.109182
Price	-0.133086	-0.109182	1.000000

¿Cuáles son las variables relevantes e irrelevantes para el análisis?

Dependiendo del análisis que se quiere llevar cabo todas las variables pueden resultar útiles, pero se pueden descartar más fácilmente aquellas que no contengan datos numéricos (como el título o el autor) o incluso variables como el año, o el número de reviews de un libro en específico.

▼ Analisis gráfico

¿Hay alguna variable que no aporta información? Si tuvieras que eliminar variables, ¿cuáles quitarías y por qué?

Considero que todas pueden ser útiles dependiendo del análisis que se quiera llevar a cabo, pero una variable que podría llegar a ser eliminada sería el número de reviews que tiene el libro, este dato tiene un rango muy amplio y podría ser reemplazado por el user rating. Para este análisis en específico se puede disponer fácilmente del autor del libro en cuestión.

¿Existen variables que tengan datos extraños?

Considero que la variable de género está muy limitada en los datos que contiene y que se podría especificar la moneda en la que están expresados los precios. Aparte de esto se puede ver que el campo de reviews tiene una gran diferencia de valores dentro de sí mismo.

Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte?

De cierta forma todas las variables numéricas tienen un rango dentro de su propia categoría, con excepción de reviews, pero cuando se comparan las 3 se observa que hay escalas muy distintas entre ellas, esto puede afectar al análisis general.

¿Puedes encontrar grupos que se parezcan? ¿Qué grupos son estos?

Por el momento no parece haber grupos que compartan características en específico, después del análisis posterior a esta sección se espera llegar a identificar y describir adecuadamente estos grupos.

```
#1 gráfico de caja (boxplot)
fig = plt.figure(figsize=(7,5))
sns.boxplot(data=df, x='User Rating', y = 'Genre')
plt.title('Histograma de la distribución de User Ratings por genero')
```

```
Text(0.5, 1.0, 'Histograma de la distribución de User Ratings por genero')
```

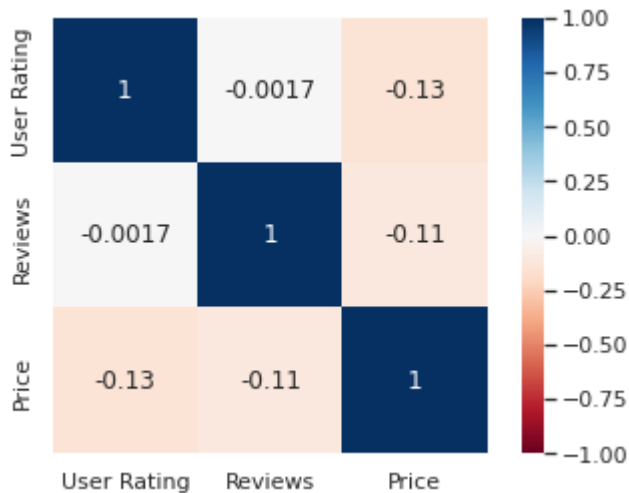
Histograma de la distribución de User Ratings por genero



```
#1 mapa de calor
```

```
sns.heatmap(data=dfwyc, vmin=-1, vmax=1, cmap = 'RdBu', annot=True, square = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f49a660eb50>
```



```
#1 gráfico de dispersión
```

```
fig = plt.figure(figsize=(6, 4))
```

```
sns.scatterplot(data=df, x='User Rating', y='Price', hue='Year')
```

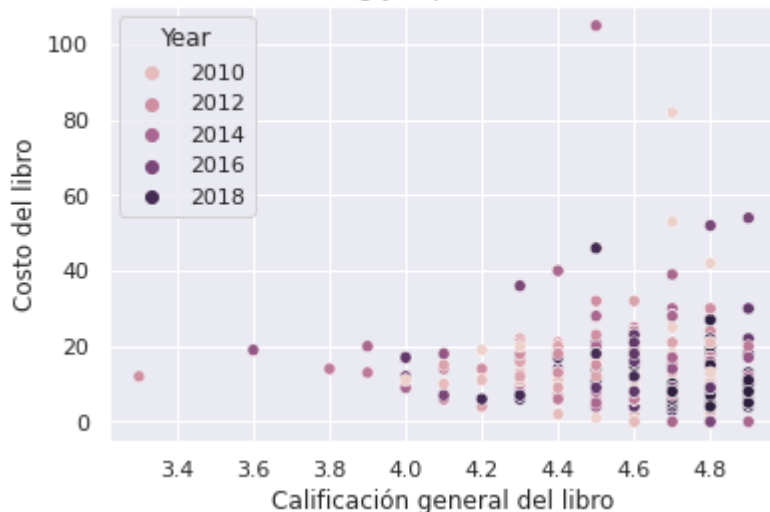
```
plt.title('Relación entre el User Rating y el precio del libro a traves de los años')
```

```
plt.xlabel('Calificación general del libro')
```

```
plt.ylabel('Costo del libro')
```

```
Text(0, 0.5, 'Costo del libro')
```

Relación entre el User Rating y el precio del libro a traves de los años



Describe brevemente las conclusiones:

Algunos hechos que salen a la vista despues de este breve analisis es el hecho de que las libros con genero de NO-ficción suelen tener una calificación general más baja pero se mantiene constante (4.6), mientras que los libros de ficción suelen tener más datos atipicos, es decir suelen variar en la calificación de los usuarios y posiblemente en su calidad. Tambien se puede apreciar que a pesar de que los precios se matengan dentro de una rango definido, los libros con mejores calificaciones suelen tener más casos de precios superiores a lós demas.


▼ Clústering

Implementa el algoritmo de kmeans y justifica la elección del número de clusters. Usa las variables numéricas

```
# Seleccionamos las variables a normalizar
numeric_cols = ['User Rating', 'Reviews', 'Price',]
X = dfwy.loc[:, numeric_cols]

# Hacemos el escalamiento.
scaler = StandardScaler()
X_norm = scaler.fit_transform(X)

# El escalador nos genera una matriz de numpy. Vamos a convertirlo en DF
X_norm = pd.DataFrame(X_norm, columns=numeric_cols)
X_norm.head()
```

	User Rating	Reviews	Price	
0	0.359990	0.460453	-0.470810	
1	-0.080978	-0.844786	0.821609	
2	0.359990	0.599440	0.175400	
3	0.359990	0.808050	-0.655441	
4	0.800958	-0.365880	-0.101547	

```
# Declaramos algunos arreglos. Los usaremos para guardar los valores de la WCSS
# y la silhouette score
kmax = 10
grupos = range(2, kmax)
wcss = []
sil_score = []
# Ciclo para calcular K-Means para diferentes k
for k in grupos:
```

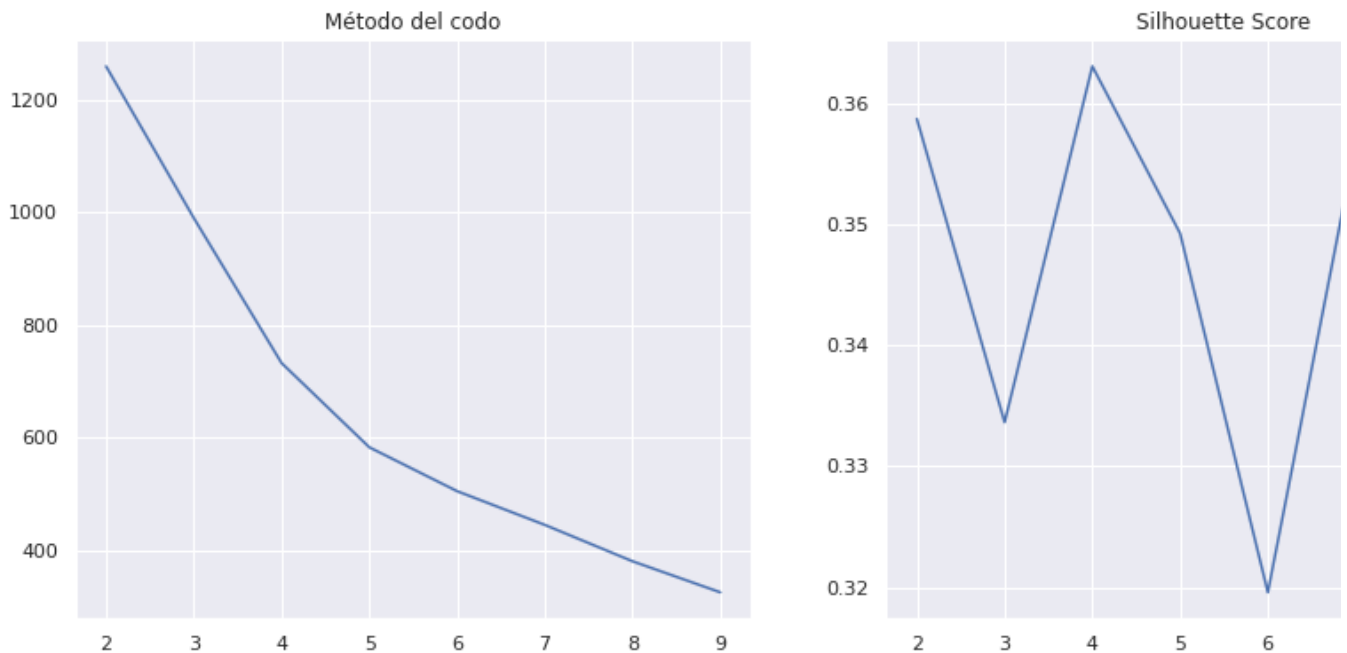
```
# Clustering
model = KMeans(n_clusters=k)
# Obtener las etiquetas
clusters = model.fit_predict(X_norm)
# Guardar WCSS
wcss.append(model.inertia_)
# Guardar Silhouette Score
sil_score.append(silhouette_score(X_norm, clusters))

# Graficaremos el codo y silhouette score en la misma gráfica. Recorda que
# subplots nos permite tener más gráficas en la misma figura.
fig, axs = plt.subplots(1, 2, figsize=(15, 6))

# Primera figura es el codo
axs[0].plot(grupos, wcss)
axs[0].set_title('Método del codo')

# La segunda es el Silhouette Score
axs[1].plot(grupos, sil_score)
axs[1].set_title('Silhouette Score')
```

```
Text(0.5, 1.0, 'Silhouette Score')
```



Después de llevar a cabo el análisis por ambos métodos, en especial por el método de Silhouette Score, se puede apreciar que los datos normalizados arrojan a 4 como el número óptimo para los clusters, 2 podría ser considerado como la siguiente opción más viable.

```
# Generamos los 4 grupos
model = KMeans(n_clusters=4)
clusters = model.fit_predict(X_norm)
```

```
clusters = model.fit_predict(X_norm)
```

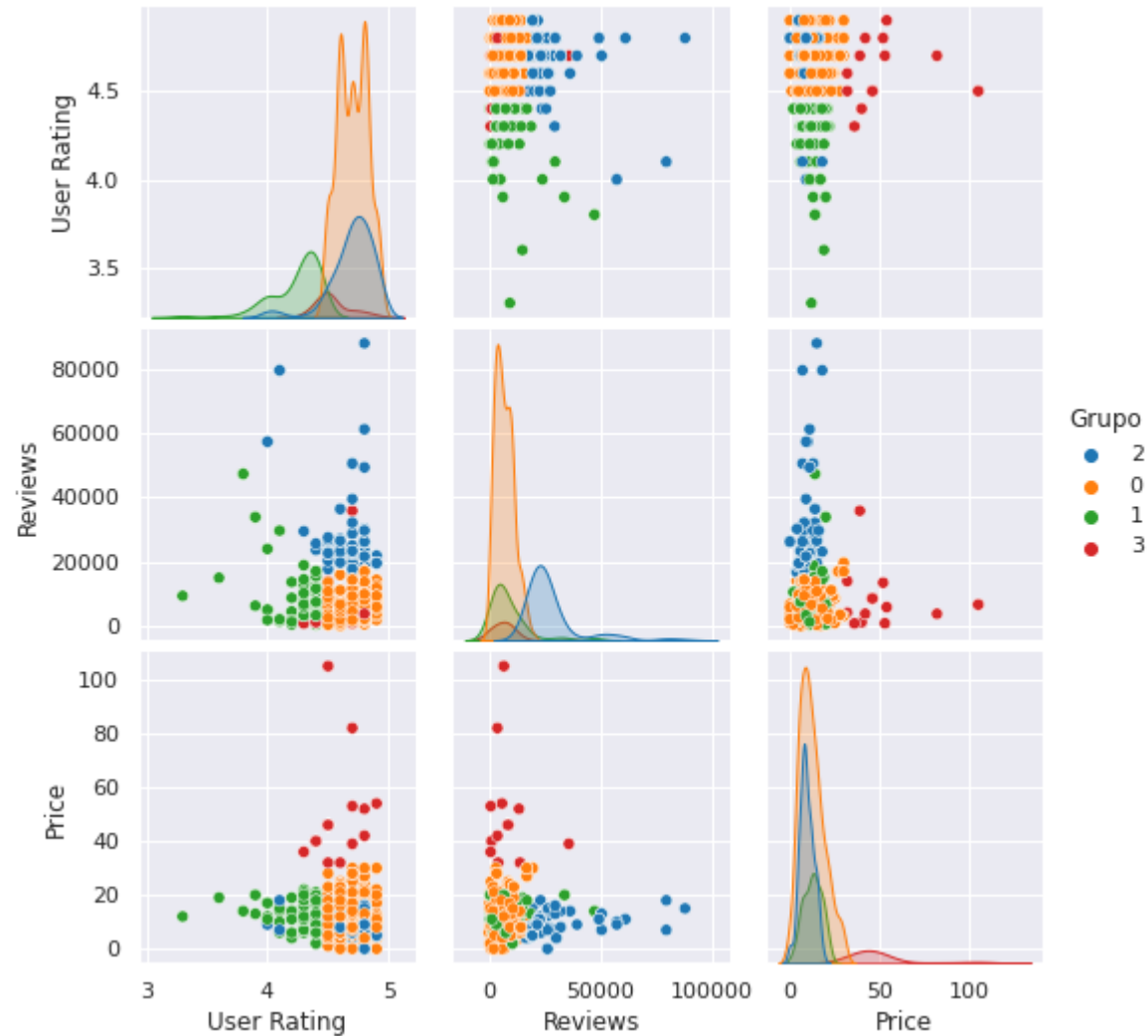
```
# Agregamos los clusters a nuestros DATOS ORIGINALES
dfwy['Grupo'] = clusters.astype('str')
dfwy.head()
```

	Name	Author	User Rating	Reviews	Price
0	10-Day Green Smoothie Cleanse	JJ Smith	4.7	17350	8
1	11/22/63: A Novel	Stephen King	4.6	2052	22
2	12 Rules for Life: An Antidote to Chaos	Jordan B. Peterson	4.7	18979	15
3	1984 (Signet Classics)	George Orwell	4.7	21424	6

```
sns.pairplot(data=dfwy, hue='Grupo', palette='tab10')
plt.suptitle('4 grupos dentro de todos los libros', y=1.05)
```

Text(0.5, 1.05, '4 grupos dentro de todos los libros')

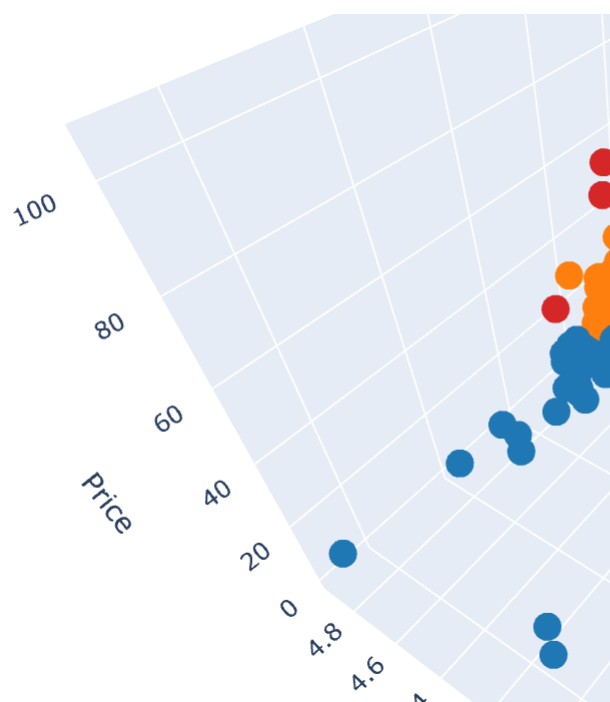
4 grupos dentro de todos los libros




```
# Creamos la figura donde graficaremos
fig = px.scatter_3d(dfwy, x = 'User Rating', y = 'Reviews',
                    z = 'Price',
                    title='4 grupos dentro de todos los libros',
                    color='Grupo',
                    color_discrete_sequence=px.colors.qualitative.D3)

# mostramos la imagen
fig.show()
```

4 grupos dentro de todos los libros



```
# Veamos las características de cada grupo (i.e. los centros)
dfwy.groupby('Grupo').mean()
```

User Rating

Reviews

Price

**Grupo**

Analiza las características de cada grupo. ¿Qué nombre le pondrías a cada segmento?

4 1 222112 9621 666667 12 116667

Grupo 1: Libros bien calificados pero poco conocidos

Libros que tienen un costo aproximado de 11, una baja cantidad de reviews comparado a los demas grupos y un User Rating de aproximadamente 4.7

Grupo 2: Libros medianamente conocidos con un precio razonable ('comunes')

Libros que tienen un costo aproximado de 12.5, una cantidad 'media' de reviews y un User Rating de aproximadamente 4.2

Grupo 3: Libros con un alto impacto y bien calificados

Libros que tienen un costo aproximado de 9, una gran cantidad de reviews y un User Rating de aproximadamente 4.7. Es muy probable que los libros dentro de estas categoría sean libros que han tenido un gran exito desde un punto de vista literario y comercial.

Grupo 4: Libros caros/de colección

Libros con un precio superior a los demas, de aproximadamente 49,5. Estos libros tienen un user rating bueno, pero no excelente dentro de los parametros de la categoría, es muy probable que estos libros sean ediciones coleccionables, especializados o parte de sagas consolidadas.

✓ 0 s completado a las 12:45

×