01MIAR - Python 101, Data Types

Luis Pilaguano

Examen del Tercer Parcial

Regresión lineal

Dado el archivo **movie_genre_classification_final.csv** se encuentra un base de datos con información de **50000 películas** que incluyen variables como:

- Título,
- Año de estreno,
- Duración,
- Puntuación promedio,
- Número de votos,
- Presupuesto,
- Ingresos en taquilla,
- Número de premios obtenidos,
- Entre otras caracterisiticas. Como parate del análisis de predicción, se desea estudiar la relación entre el:
- Presupuesto de una película y sus caracteristicas cuantitativas, para estimar la **recaudación en taquilla (BoxOffice_USD)**. Para ello, se plantea entrenar un modelo de regresión lineal.

Objetivo

Predecir el valor de la variable **BoxOffice_USD** a partir de variables numéricas independientes disponibles en el dataset, tales como:

- Budget_USD (Presupuesto en dólares)

- Duration (Duración en minutos)
- Ratting (Calificiación Promedio"
- Votes (Número de votos)
- Num_Awards (Número de premios obtenidos)
- Critic_reviews (Números de reseñas de criticos)

El dataset fue extraido desder DataSet

Entrega esperada

- Código bien estructurado y comentado
- Gráficos claros en los pasos 2, 4 y 5.
- Breve análisis escrito de lso los resultados obtenidos.

In [12]:
```
# Lectura de datos
import pandas as pd
# Cargar el archivo CSV
df = pd.read_csv("res/movie_genre_classification_final.csv")
# Mostrar las primeras 5 filas
df.head(20)
```

Out[12]:

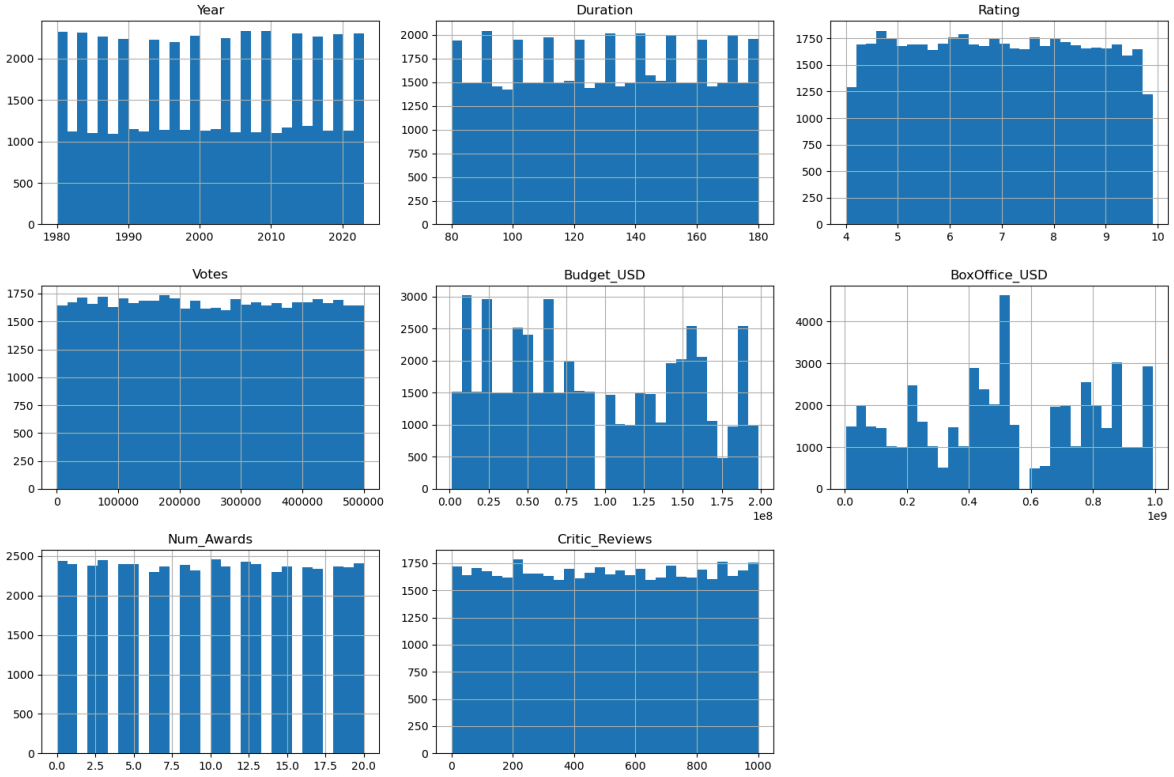
	Title	Year	Director	Duration	Rating	Votes	Description	Language	Coun
0	Winds of Fate 4	1980	R. Lee	167	4.1	182425	A touching love story with heartwarming moments.	Spanish	Ch
1	Firestorm 11	2014	S. Chen	166	4.1	449351	A fast-paced thriller with intense action scenes.	Korean	Ch
2	Silent Echo 2	2016	A. Khan	170	4.1	363328	A fast-paced thriller with intense action scenes.	Korean	Jap
3	City Lights 4	1982	L. Zhang	170	9.9	62371	An emotional journey exploring complex charact...	Japanese	Jap
4	Broken Truth 1	1990	L. Zhang	91	5.3	4600	An imaginative world filled with magic and won...	Korean	U
5	Broken Truth 12	1985	D. Patel	116	4.4	268620	An emotional journey exploring complex charact...	English	Fra
6	Crimson Sky 20	2004	R. Lee	117	5.3	38970	A light-hearted comedy that guarantees laughter.	Spanish	Ch
7	Eternal Love 4	2016	T. Johnson	111	5.5	86559	A light-hearted comedy that guarantees laughter.	English	U
8	Broken Truth 7	1982	L. Zhang	129	4.3	288173	An emotional journey exploring complex charact...	French	U
9	Winds of Fate 16	1993	S. Chen	169	7.5	355096	An emotional journey exploring complex charact...	Hindi	U
10	Winds of Fate 6	1990	M. Brown	112	4.6	327982	A suspenseful plot filled	English	

	Title	Year	Director	Duration	Rating	Votes	Description	Language	Coun
							with unexpected twists.		
11	Ocean Call 20	2002	J. Smith	135	7.9	171468	A suspenseful plot filled with unexpected twists.	Spanish	Jap
12	Frozen Whisper 16	2006	M. Brown	96	9.9	426978	A spine-chilling tale that evokes fear and dread.	English	So Ko
13	Eternal Love 17	1985	N. Roy	168	4.4	370579	A suspenseful plot filled with unexpected twists.	French	Jap
14	Ocean Call 19	1980	R. Lee	123	5.5	34337	An imaginative world filled with magic and won...	Japanese	
15	Midnight Sun 1	1993	A. Khan	162	4.3	50482	A fast-paced thriller with intense action scenes.	Korean	Jap
16	Crimson Sky 2	2020	M. Brown	89	4.6	487738	An imaginative world filled with magic and won...	English	Ch
17	Lost World 8	2021	L. Zhang	179	7.7	15429	A spine-chilling tale that evokes fear and dread.	Mandarin	
18	Quiet Heart 11	2006	P. Adams	155	4.4	118700	A spine-chilling tale that evokes fear and dread.	Mandarin	So Ko
19	Midnight Sun 15	1988	N. Roy	86	5.3	378137	An imaginative world filled with magic and won...	Japanese	U

In [10]:
```
# Visualización del conjunto de datos
# Importo las librerías necesarias
```

```
import matplotlib.pyplot as plt  #   gráficos de forma manual
import seaborn as sns  #  gráficos  personalizados

#  Histograma de todas las columnas numéricas del DataFrame, en este caso son 8
df.hist(bins=30, figsize=(15, 10))  # Crea un histograma para cada columna numér
plt.tight_layout()  # Ajusta automáticamente el espacio entre los gráficos
plt.show()  # Muestra todos los histogramas
```



In [22]:
```
# Modificación del conjutno de datos
# señecciono solo las columnas necesarias para el analisis en este caso son 7
columnas_relevantes = ['Budget_USD','Duration','Rating','Votes','Num_Awards','Cr
                       'BoxOffice_USD'   # esta ba ser mi variable a predecir
]

# Revisar las columnas del DataFrame para confirmar nombres exactos
print("--------------Estos son todas las columnas de mi archivo csv antes de la
print(df.columns)
# creo un nuevo DataFrame solo con esas columnas
df_model = df[columnas_relevantes].copy()
# Mostrar el tamaño final
print(f"Tamaño del DataFrame tras limpiar: {df_model.shape}")
# Mostrar las primeras filas del nuevo DataFrame limpio
df_model.head()
```

```
--------------Estos son todas las columnas de mi archivo csv antes de la seleccio
n de columnas a utilizar-----------------
Index(['Title', 'Year', 'Director', 'Duration', 'Rating', 'Votes',
       'Description', 'Language', 'Country', 'Budget_USD', 'BoxOffice_USD',
       'Genre', 'Production_Company', 'Content_Rating', 'Lead_Actor',
       'Num_Awards', 'Critic_Reviews'],
      dtype='object')
Tamaño del DataFrame tras limpiar: (50000, 7)
```

Out[22]:

	Budget_USD	Duration	Rating	Votes	Num_Awards	Critic_Reviews	BoxOffice_USD
0	39979615	167	4.1	182425	8	229	179936008
1	116404774	166	4.1	449351	20	466	802121619
2	166261330	170	4.1	363328	16	539	225526871
3	28861315	170	9.9	62371	15	606	69813738
4	43890403	91	5.3	4600	6	330	375136716

In [23]:
```
#Construcción del modelo de regresión lineal

# Budget_USD (Presupuesto en dólares)
# Duration (Duración en minutos)
# Ratting (Calificiación Promedio"
# Votes (Número de votos)
# Num_Awards (Número de premios obtenidos)
# Critic_reviews (Números de reseñas de criticos)

from sklearn.linear_model import LinearRegression

# Definir variables independientes (X) y dependiente (y)
X = df_model.drop('BoxOffice_USD', axis=1)
y = df_model['BoxOffice_USD']

# Crear el modelo
modelo = LinearRegression()
# Entrenar el modelo con todos los datos
modelo.fit(X, y)
# Mostrar coeficiente (pendientes) y intercepto (constante)
print("Intercepto (b0):", modelo.intercept_)
print("Coeficientes (b1, b2, ...):")
for var, coef in zip(X.columns, modelo.coef_):
    print(f"  {var}: {coef}")
```

```
Intercepto (b0): 529537671.2698003
Coeficientes (b1, b2, ...):
  Budget_USD: 0.004876447897468708
  Duration: -71599.5862332624
  Rating: -996538.5469178511
  Votes: -8.357011648244224
  Num_Awards: 483656.8223504322
  Critic_Reviews: 2207.2402677343975
```

In [28]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pandas as pd

## POR TIEMPO Y CLASE DE SIGUIENTE HORA ----------------datas generados por la
# Variables independientes (X) y objetivo (y)
X = df_model.drop('BoxOffice_USD', axis=1)
y = df_model['BoxOffice_USD']

# División 80% entrenamiento, 20% prueba
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
# Crear y entrenar el modelo
modelo = LinearRegression()
modelo.fit(X_train, y_train)

# Predecir en conjunto de prueba
y_pred = modelo.predict(X_test)

# Crear DataFrame con resultados reales y predicciones, formateando números con
resultados = pd.DataFrame({
    'Real': y_test.values,
    'Predicción': y_pred
})

# Formatear números para mejor lectura
resultados['Real'] = resultados['Real'].apply(lambda x: f"${x:,.0f}")
resultados['Predicción'] = resultados['Predicción'].apply(lambda x: f"${x:,.0f}"

# Mostrar las primeras 10 filas
print(resultados.head(10))
```

```
          Real    Predicción
0  $200,330,529  $517,533,775
1  $488,839,661  $520,799,513
2  $428,612,829  $514,359,671
3  $802,121,619  $510,530,996
4  $765,377,261  $524,760,739
5  $856,605,219  $515,216,590
6  $719,946,753  $513,235,545
7  $980,966,690  $515,138,661
8  $477,372,302  $515,448,445
9  $403,157,039  $510,798,174
```

GIT HUB

In [ ]: