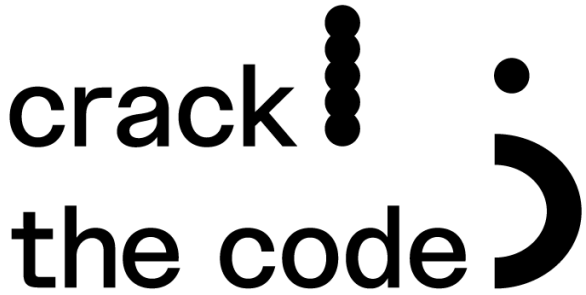
“Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho”



Proyecto 1

Análisis de Películas y Series de TV en IMDB

DOCENTE:

Carlos Eduardo Vásquez Roque

INTEGRANTES:

- Egoavil Cárdenas, Giovanny Angelo

TURNO: Mañana

LIMA – PERÚ

2024

SCI: Inteligencia artificial - Proyecto 1

Análisis de Películas y Series de TV en IMDB

1. Introducción:

En la era de la información, la industria del entretenimiento, específicamente el cine y la televisión, genera una cantidad masiva de datos. Estos datos no solo incluyen aspectos financieros, como ingresos en taquilla, sino también información sobre el reparto, directores, géneros, y la recepción del público a través de votaciones y críticas. Analizar estos datos puede ofrecer valiosas perspectivas sobre las tendencias y factores de éxito en esta industria.

El presente proyecto tiene como objetivo explorar y analizar datos de películas y series de televisión para identificar patrones y relaciones significativas entre diversas variables. Utilizando el conjunto de datos de IMDb de las 1000 mejores películas y series de televisión de Kaggle, que contiene información detallada sobre títulos, años de lanzamiento, certificaciones, duraciones, géneros, calificaciones en IMDb, resúmenes, puntuaciones en Metacritic, directores, actores principales, número de votos, recaudación bruta y conformación de equipos, se busca entender mejor la dinámica de esta industria.

Para llevar a cabo este análisis, se emplearán herramientas y bibliotecas de Python como Numpy, Pandas y Matplotlib en un entorno de Colab Notebook. Se realizarán diversas etapas de análisis que incluyen la limpieza básica de datos, análisis exploratorio, manejo de valores atípicos, creación de características, normalización y modelado matemático mediante regresión lineal simple.

1.1. Objetivos

- **Realizar la limpieza básica de datos:** Manejar valores nulos y tipos de datos para asegurar la integridad y consistencia del conjunto de datos.

- **Explorar tendencias y distribuciones:** Analizar las tendencias, distribuciones y relaciones entre variables clave utilizando técnicas de análisis exploratorio de datos (EDA).

- **Identificar y manejar valores atípicos:** Filtrar y manejar valores atípicos para mejorar la calidad del análisis y evitar sesgos en los resultados.

- **Crear nuevas características:** Generar nuevas variables a partir de las existentes para enriquecer el conjunto de datos y obtener mejores insights.

- **Normalización y escalamiento:** Normalizar y escalar los datos para prepararlos para el modelado matemático.

- **Utilizar regresión lineal simple:** Manejar el modelo y entender las relaciones entre diferentes características de películas y series de televisión.

- **Responder preguntas clave:** A través del análisis, responder preguntas específicas sobre los directores, actores y géneros que influyen en el éxito de las películas y series, tanto en términos de ingresos en taquilla como de calificaciones y votos en IMDb. Las preguntas incluyen:

✓ ¿Qué directores tienden a generar mayores ingresos en taquilla?
✓ ¿Qué actores están asociados con películas que generan mayores ingresos?

- ✓ ¿Qué directores tienden a recibir más votos en IMDb?
- ✓ ¿Qué actores están asociados con películas que reciben más votos?
- ✓ ¿Qué géneros de películas son los preferidos por los actores?
- ✓ ¿Qué combinación de actores (Star1, Star2, Star3 y Star4) está obteniendo buenas calificaciones en IMDb la mayor parte del tiempo?
- ✓ ¿Qué combinación de actores (Star1, Star2, Star3 y Star4) está obteniendo buenos ingresos en taquilla?

Estos objetivos buscan proporcionar una comprensión integral de los factores que influyen en el éxito de películas y series de televisión, ofreciendo insights valiosos para estudios académicos, estrategias de marketing y decisiones de producción.

2. Carga y Limpieza de Datos:

2.1. Data Set

El conjunto de datos "IMDb Dataset of Top 1000 Movies and TV Shows" de Kaggle contiene información detallada sobre las 1000 películas y series de televisión mejor calificadas según IMDb. Este conjunto de datos es una valiosa fuente de información para analizar y entender diversas características y tendencias dentro de la industria del entretenimiento. Las principales características del conjunto de datos incluyen:

- **Título:** El nombre de la película o serie de televisión.
- **Año de lanzamiento:** El año en que la película o serie fue lanzada.
- **Certificación:** La clasificación de edad asignada al contenido (por ejemplo, PG-13, R).
- **Duración:** La duración de la película o serie en minutos.
- **Género:** Los géneros a los que pertenece la película o serie (por ejemplo, Drama, Comedia, Acción).
- **Calificación en IMDb:** La puntuación promedio de la película o serie en IMDb, basada en las votaciones de los usuarios.
- **Resumen:** Una breve descripción de la trama de la película o serie.
- **Puntuación en Metacritic:** La calificación otorgada por Metacritic, que agrega críticas de varias fuentes.

- **Director:** El director de la película o serie.
- **Actores principales:** Los actores principales que participaron en la película o serie.
- **Número de votos:** La cantidad de votos que la película o serie ha recibido en IMDb.
- **Recaudación bruta:** Los ingresos totales generados por la película en taquilla.
- **Conformación de Equipos:** Información sobre el equipo de producción y otros detalles relevantes.

Por lo cual cargamos el conjunto de datos "IMDb Dataset of Top 1000 Movies and TV Shows"

	Poster_Link	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_score	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross
0	https://m.media-amazon.com/images/M/MV5BMDFkYT...	The Shawshank Redemption	1994	A	142 min	Drama	9.3	Two imprisoned men bond over a number of years...	80.0	Frank Darabont	Tim Robbins	Morgan Freeman	Bob Gunton	William Sadler	2343110	28,341,469
1	https://m.media-amazon.com/images/M/MV5BM2MyNj...	The Godfather	1972	A	175 min	Crime, Drama	9.2	An organized crime dynasty's aging patriarch t...	100.0	Francis Ford Coppola	Marlon Brando	Al Pacino	James Caan	Diane Keaton	1620367	134,966,411
2	https://m.media-amazon.com/images/M/MV5BMTMxNT...	The Dark Knight	2008	UA	152 min	Action, Crime, Drama	9.0	When the menace known as the Joker wreaks havo...	84.0	Christopher Nolan	Christian Bale	Heath Ledger	Aaron Eckhart	Michael Caine	2303232	534,858,444
3	https://m.media-amazon.com/images/M/MV5BMWMwMG...	The Godfather: Part II	1974	A	202 min	Crime, Drama	9.0	The early life and career of Vito Corleone in ...	90.0	Francis Ford Coppola	Al Pacino	Robert De Niro	Robert Duvall	Diane Keaton	1129952	57,300,000
4	https://m.media-amazon.com/images/M/MV5BMWU4N2...	12 Angry Men	1957	U	96 min	Crime, Drama	9.0	A jury holdout attempts to prevent a miscarria...	96.0	Sidney Lumet	Henry Fonda	Lee J. Cobb	Martin Balsam	John Fiedler	689845	4,360,000

2.2. Limpieza de Datos:

El proceso de limpieza de datos es esencial para asegurar la calidad y precisión de los análisis. A continuación, se describen brevemente los pasos seguidos para la limpieza de datos en este proyecto:

2.2.1 Detección de Valores Faltantes:

- Se examinaron todas las columnas del conjunto de datos para identificar aquellas que contienen valores faltantes.

```
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Poster_Link    1000 non-null   object
 1   Series_Title   1000 non-null   object
 2   Released_Year  1000 non-null   object
 3   Certificate    899 non-null    object
 4   Runtime        1000 non-null   object
 5   Genre          1000 non-null   object
 6   IMDB_Rating    1000 non-null   float64
 7   Overview       1000 non-null   object
 8   Meta_score     843 non-null    float64
 9   Director       1000 non-null   object
 10  Star1          1000 non-null   object
 11  Star2          1000 non-null   object
 12  Star3          1000 non-null   object
 13  Star4          1000 non-null   object
 14  No_of_Votes    1000 non-null   int64
 15  Gross          831 non-null    object
dtypes: float64(2), int64(1), object(13)
```

- Se utilizó la función isnull() de Pandas para detectar y contar los valores nulos en cada columna.

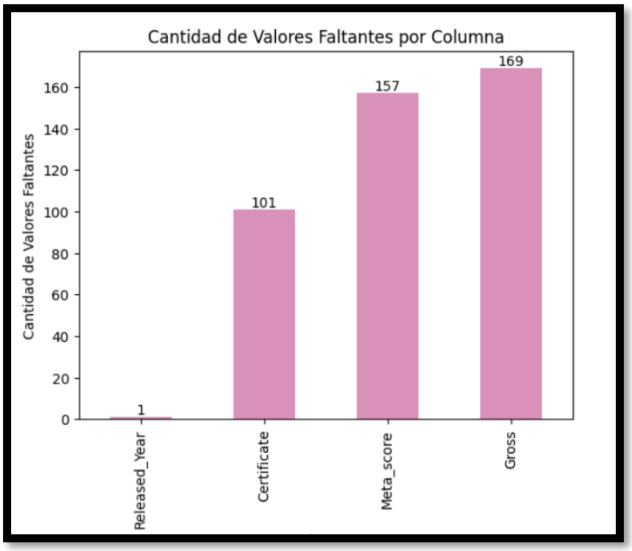
```
Poster_Link        0
Series_Title       0
Released_Year      0
Certificate      101
Runtime            0
Genre              0
IMDB_Rating        0
Overview           0
Meta_score       157
Director           0
Star1              0
Star2              0
Star3              0
Star4              0
No_of_Votes        0
Gross            169
dtype: int64
```

- Filtramos valores faltantes que sean mayores a 0.

```
Released_Year      1
Certificate      101
Meta_score       157
Gross            169
dtype: int64
```

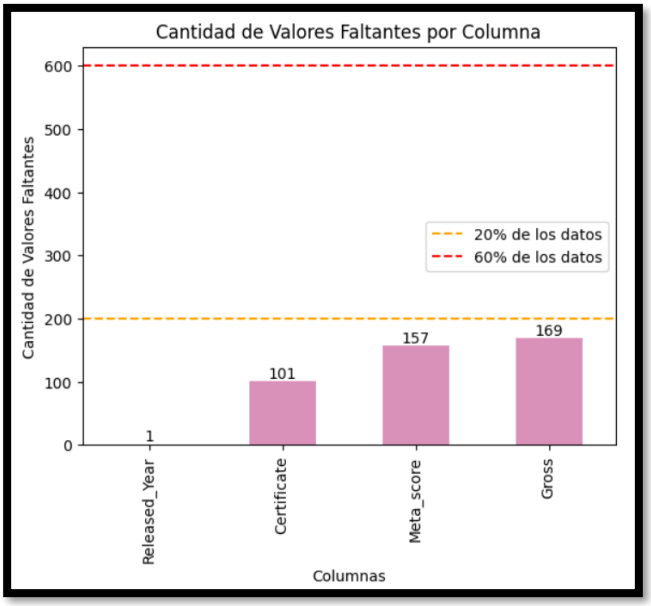
2.2.2 Reporte de Valores Faltantes:

Se generó un informe detallado que muestra la cantidad y el porcentaje de valores faltantes en cada columna del conjunto de datos. Esto ayudó a priorizar las columnas que requerían mayor atención en el proceso de limpieza.



2.2.3 Análisis de Valores Faltantes:

Se realizó un análisis para entender las posibles razones detrás de los valores faltantes y se evaluó el impacto de estos valores faltantes en los análisis posteriores y se decidió realizar un gráfico para visualizar la cantidad de valores faltantes.



Según a la gráfica, ninguna variable sobrepasa el umbral del 60%, por lo que no hace falta borrar ninguna columna.

2.2.4 Reemplazo de Valores Nulos:

Para la variable **Certificación**, se reemplazaron los valores nulos con la moda (el valor más frecuente), ya que esta variable es categórica y la moda es representativa de la mayoría de los datos.

```
#Almacenamos la moda en una variable
mode_certificate = data['Certificate'].mode()[0]

#Rellenamos los NAN con la moda
data['Certificate'].fillna(mode_certificate, inplace=True)

#Comprobamos si ya no tenemos NANS
data['Certificate'].isnull().sum()
```

Para la variable **Recaudación bruta**, se utilizó la mediana para reemplazar los valores nulos. La mediana es más robusta frente a valores atípicos y proporciona una representación central más precisa para los datos de ingresos.

```
data['Gross'].fillna(data['Gross'].median(), inplace=True)

# comprobamos que ya no tengan valores faltantes
data['Gross'].isnull().sum()
```

Para la variable **Puntuación en Metacritic**, se utilizó la media para reemplazar los valores nulos, ya que esta variable es numérica y la media proporciona una buena medida de tendencia central para datos normalmente distribuidos.

```
data['Meta_score'].fillna(data['Meta_score'].median(), inplace=True)

#C Comprobamos el cambio
data['Meta_score'].isnull().sum()
```

Para la variable **Año de Lanzamiento**, se utilizó el dropna() para eliminar los datos de la variable, ya que solo era un valor faltante.

```
data = data.dropna(subset=['Released_Year'])
```

Estos pasos aseguran que el conjunto de datos esté limpio y preparado para los análisis posteriores, reduciendo el sesgo y aumentando la fiabilidad de los resultados obtenidos.

3. Análisis Exploratorio de Datos (EDA):

El Análisis Exploratorio de Datos (EDA) es un paso crucial en el proceso de análisis de datos que permite comprender la estructura, patrones y relaciones dentro del conjunto de datos. A continuación, se detallan los pasos realizados para convertir datos de tipo objeto a numérico y las razones detrás de estos cambios, seguidos de una breve descripción de las estadísticas descriptivas y visualizaciones creadas:

3.1. Conversión de Datos de Tipo Objeto a Numérico

3.1.1 Conversión de la Variable "Duración" a Entero:

- La variable "Duración" contiene valores que representan el tiempo de duración de las películas y series en minutos.
- Se convirtió a tipo entero (int) para permitir cálculos y análisis estadísticos precisos.

```
# Antes de la limpieza de Runtime
Tipo de dato:  object
5 primeras Variables:  ['142 min' '175 min' '152 min' '202 min' '96 min']

************************************************************

# Después de la limpieza de Runtime
Tipo de dato:  int64
5 primeras Variables:  [142 175 152 202  96]

************************************************************
```

3.1.2 Conversión de la Variable "Año de Lanzamiento" a Tipo Float:

La variable "Año de Lanzamiento" inicialmente está en formato de cadena (objeto).

Se convirtió a tipo flotante (float) para manejar valores nulos de manera efectiva y para facilitar la visualización y el análisis de tendencias a lo largo del tiempo.

```
# Antes de la limpieza de Released_Year
Tipo de dato:  object
5 primeras Variables:  ['1994' '1972' '2008' '1974' '1957']

************************************************************

# Después de la limpieza de Released_Year
Tipo de dato:  float64
5 primeras Variables:  [1994. 1972. 2008. 1974. 1957.]

************************************************************
```

3.1.3 Conversión de la Variable "Recaudación Bruta" a Tipo Float:

La variable "Recaudación Bruta" representa los ingresos totales generados por las películas en taquilla y estaba en formato de cadena (objeto).

Se convirtió a tipo flotante (float) para permitir cálculos financieros y análisis de ingresos.

```
# Antes de la limpieza de Gross
Tipo de dato:  object
5 primeras Variables:  ['28,341,469' '134,966,411' '534,858,444' '57,300,000' '4,360,000']

************************************************************

# Después de la limpieza de Gross
Tipo de dato:  float64
5 primeras Variables:  [2.83414690e+07 1.34966411e+08 5.34858444e+08 5.73000000e+07
 4.36000000e+06]

************************************************************
```

- Finalmente, gracias a las conversiones de variables tenemos una tabla actualizada.

ANTES

	Poster_Link	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_score	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross
0	https://m.media-amazon.com/images/M/MV5BMDFkYT...	The Shawshank Redemption	1994	A	142 min	Drama	9.3	Two imprisoned men bond over a number of years...	80.0	Frank Darabont	Tim Robbins	Morgan Freeman	Bob Gunton	William Sadler	2343110	28,341,469
1	https://m.media-amazon.com/images/M/MV5BM2MyNj...	The Godfather	1972	A	175 min	Crime, Drama	9.2	An organized crime dynasty's aging patriarch t...	100.0	Francis Ford Coppola	Marlon Brando	Al Pacino	James Caan	Diane Keaton	1620367	134,966,411
2	https://m.media-amazon.com/images/M/MV5BMTMxNT...	The Dark Knight	2008	UA	152 min	Action, Crime, Drama	9.0	When the menace known as the Joker wreaks havo...	84.0	Christopher Nolan	Christian Bale	Heath Ledger	Aaron Eckhart	Michael Caine	2303232	534,858,444
3	https://m.media-amazon.com/images/M/MV5BMWMwMG...	The Godfather: Part II	1974	A	202 min	Crime, Drama	9.0	The early life and career of Vito Corleone in ...	90.0	Francis Ford Coppola	Al Pacino	Robert De Niro	Robert Duvall	Diane Keaton	1129952	57,300,000
4	https://m.media-amazon.com/images/M/MV5BMWU4N2...	12 Angry Men	1957	U	96 min	Crime, Drama	9.0	A jury holdout attempts to prevent a miscarria...	96.0	Sidney Lumet	Henry Fonda	Lee J. Cobb	Martin Balsam	John Fiedler	689845	4,360,000

DESPUÉS

	Released_Year	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross
count	999.000000	1000.000000	1000.000000	843.000000	1.000000e+03	8.310000e+02
mean	1991.217217	122.891000	7.949300	77.971530	2.736929e+05	6.803475e+07
std	23.297025	28.093671	0.275491	12.376099	3.273727e+05	1.097500e+08
min	1920.000000	45.000000	7.600000	28.000000	2.508800e+04	1.305000e+03
25%	1976.000000	103.000000	7.700000	70.000000	5.552625e+04	3.253559e+06
50%	1999.000000	119.000000	7.900000	79.000000	1.385485e+05	2.353089e+07
75%	2009.000000	137.000000	8.100000	87.000000	3.741612e+05	8.075089e+07
max	2020.000000	321.000000	9.300000	100.000000	2.343110e+06	9.366622e+08

3.2. Estadísticas Descriptivas Básicas

Se realizaron estadísticas descriptivas básicas para las variables numéricas, que incluyen:

- **Media:** La medida promedio de los datos.
- **Mediana:** El valor central de los datos ordenados.
- **Desviación Estándar:** La medida de la dispersión de los datos alrededor de la media.
- **Asimetría:** Una medida de la simetría de la distribución de los datos.
- **Curtosis:** Una medida de la "puntiagudez" de la distribución de los datos.

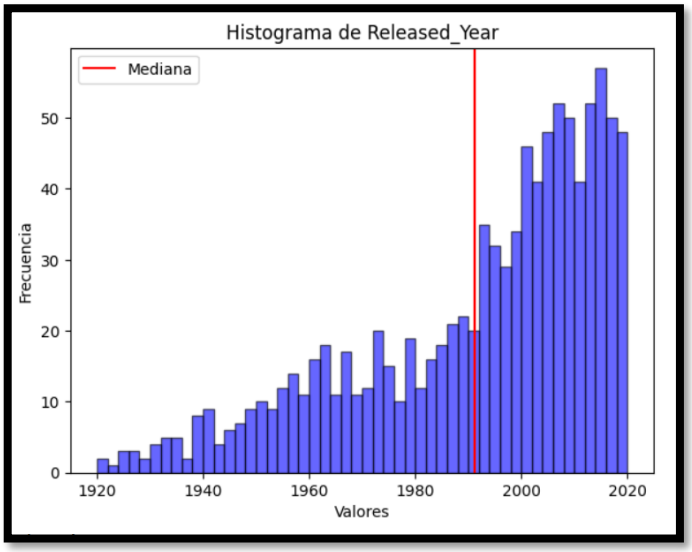
3.2.1. Visualizaciones

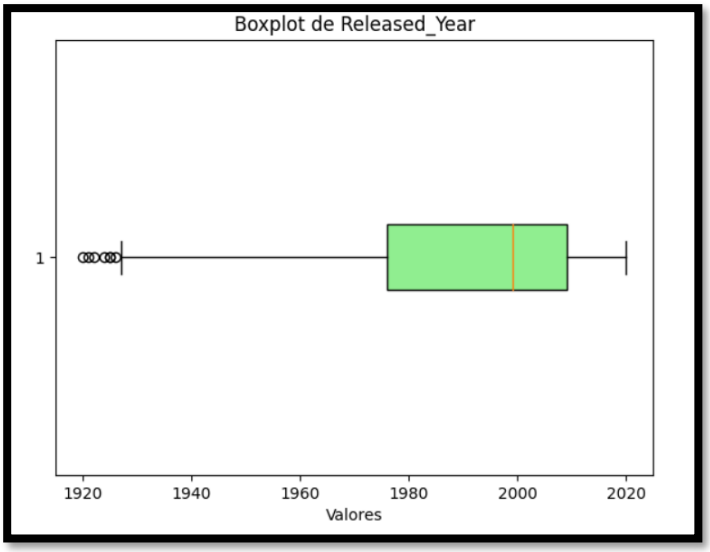
1. **Histograma:**
    - Se crearon histogramas para las variables numéricas: "Año de Lanzamiento", "Duración", "IMDB_Rating", "Puntuación en Metacritic", "Número de votos", y "Recaudación Bruta".
    - Los histogramas muestran la frecuencia de los valores en intervalos específicos, ayudando a entender la distribución de los datos.
    -
2. **Boxplot:**
    - Se crearon boxplots para las mismas variables numéricas.
    - Los boxplots visualizan la distribución de los datos a través de sus cuartiles, destacando los valores atípicos (outliers).

3.2.2. Análisis de Estadísticas Descriptivas

Para cada variable numérica:

- **Año de Lanzamiento:**
    - **Asimetría:** Puede indicar la tendencia de lanzamientos en ciertos periodos.
    - **Curtosis:** Indica la concentración de lanzamientos en ciertos años.
    - **Media, Mediana, Desviación Estándar:** Muestran la tendencia central y dispersión de los años de lanzamiento.





Asimetría: -0.939346641352243

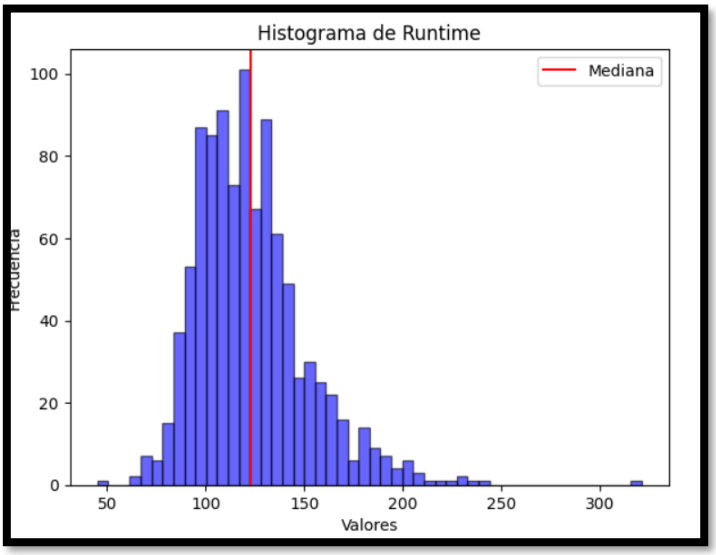
Curtosis: -0.021431891218695487

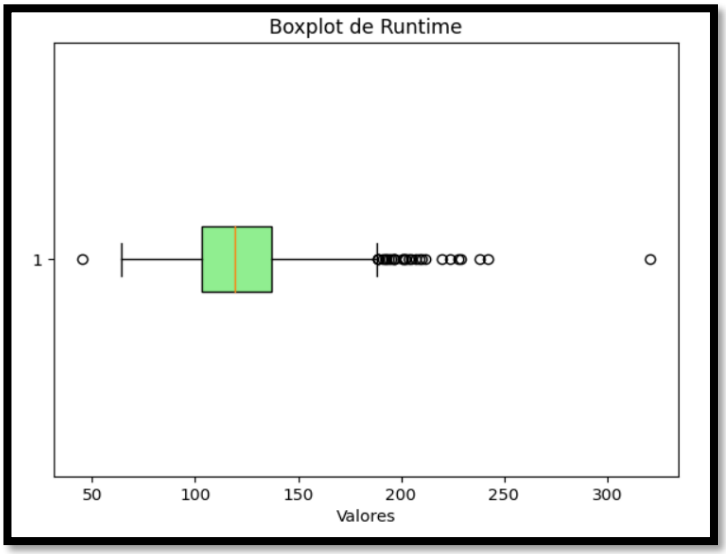
Media: 1991.2172172172172

Mediana: 1999.0

Desviación estándar: 23.297024844324177

- **Duración:**
  - **Asimetría y Curtosis:** Indican la distribución y concentración de duraciones.
  - **Media, Mediana, Desviación Estándar:** Muestran la tendencia central y variabilidad en las duraciones.





Asimetría: 1.2079088917390541

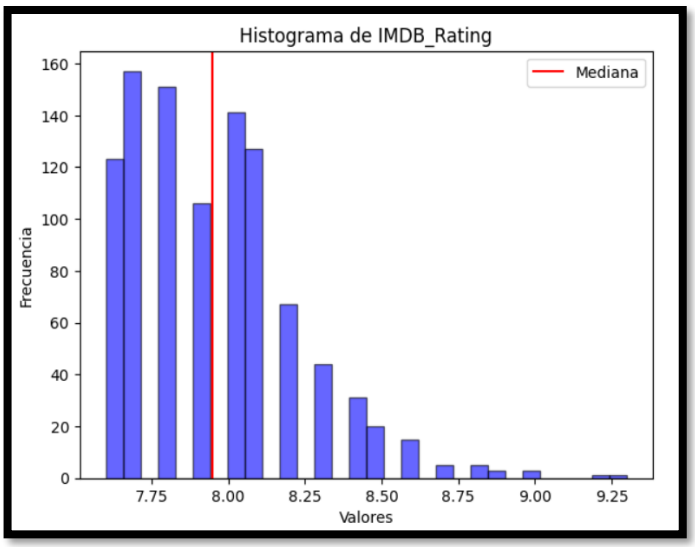
Curtosis: 3.4262648520304624

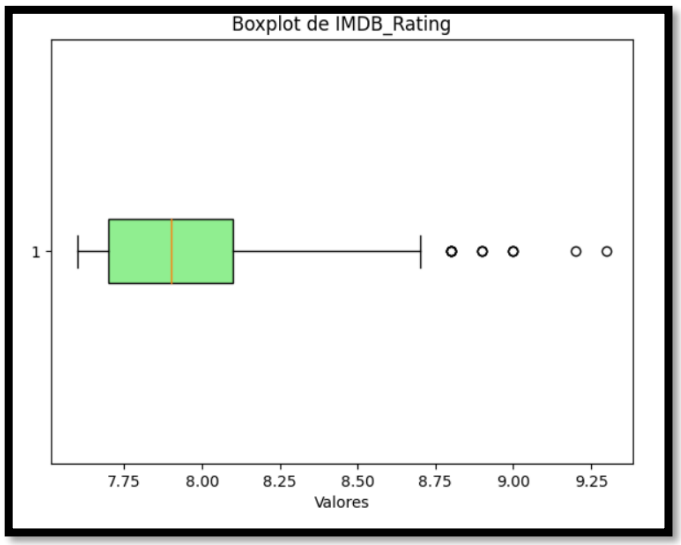
Media: 122.891

Mediana: 119.0

Desviación estándar: 28.09367141142954

- **IMDB_Rating:**
    - **Asimetría y Curtosis:** Indican la distribución de las calificaciones en IMDb.
    - **Media, Mediana, Desviación Estándar:** Proveen una visión de la tendencia central y variabilidad de las calificaciones.





Asimetría: 1.016964453611272

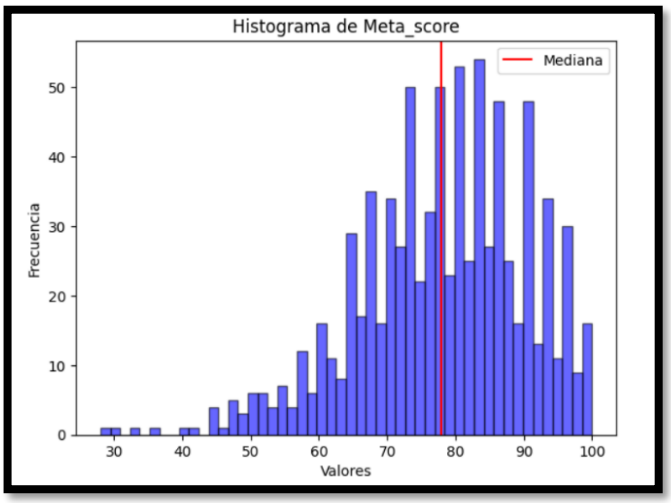
Curtosis: 1.4327269987500322

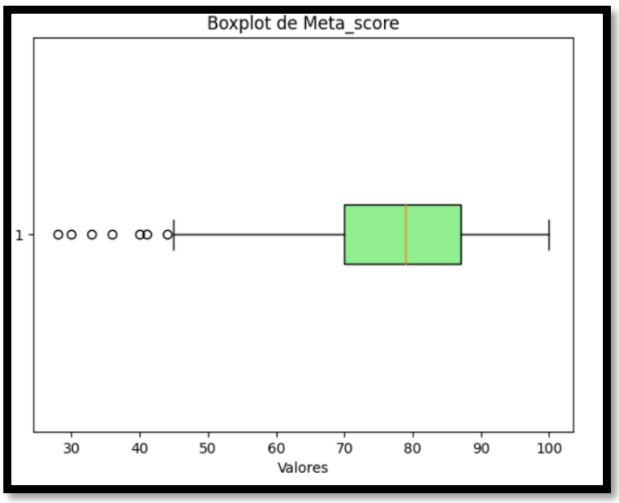
Media: 7.949299999999999

Mediana: 7.9

Desviación estándar: 0.2754912074920095

- **Puntuación en Metacritic:**
    - **Asimetría y Curtosis:** Indican la distribución de las puntuaciones en Metacritic.
    - **Media, Mediana, Desviación Estándar:** Muestran la tendencia central y dispersión de las puntuaciones.





Asimetría: -0.6052248305009935

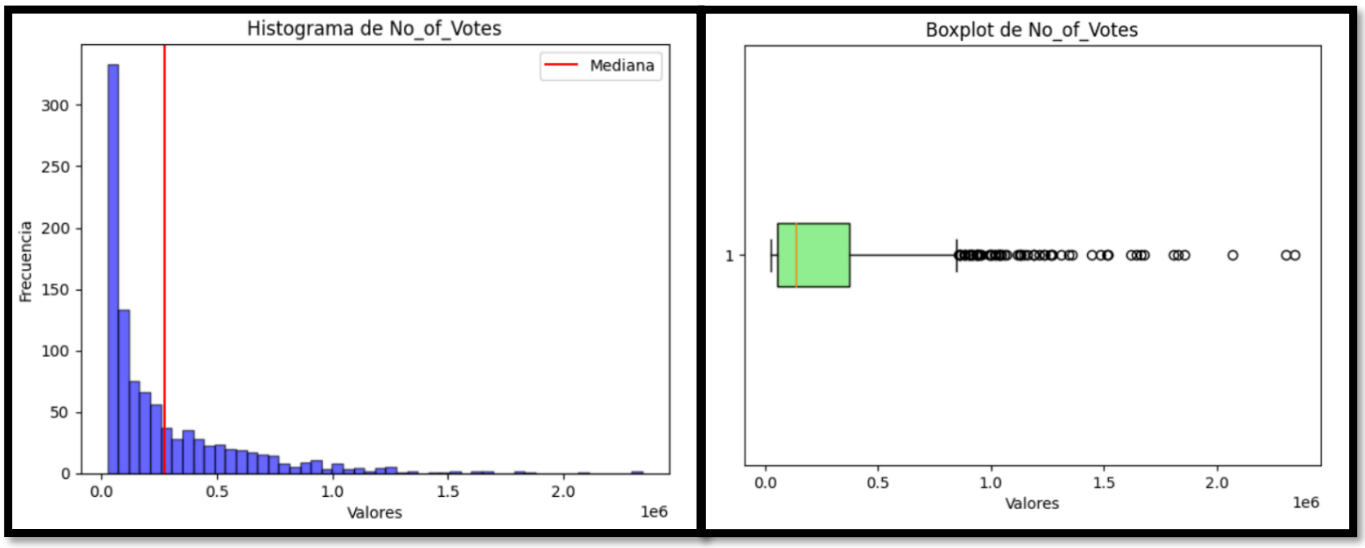
Curtosis: 0.4208306168018683

Media: 77.97153024911032

Mediana: 79.0

Desviación estándar: 12.376099328602027

- **Número de Votos:**
    - **Asimetría y Curtosis:** Indican la distribución de los votos recibidos.
    - **Media, Mediana, Desviación Estándar:** Proveen una visión de la tendencia central y variabilidad en el número de votos.



Asimetría: 2.30001058546554

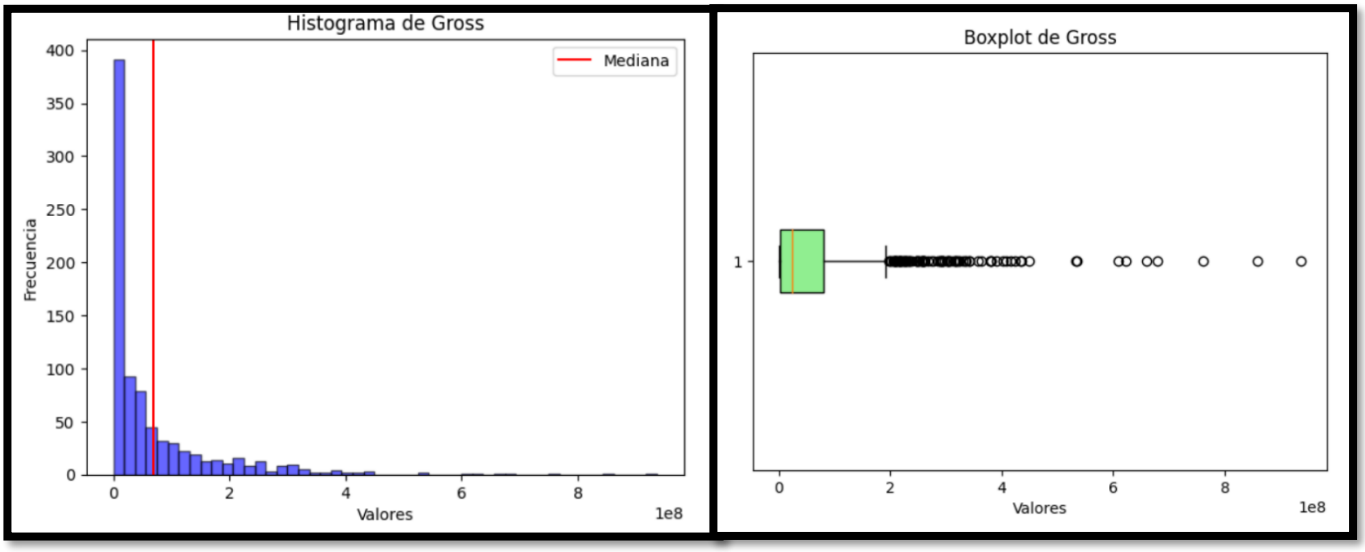
Curtosis: 6.89509932739565

Media: 273692.911

Mediana: 138548.5

Desviación estándar: 327372.703934124

- **Recaudación Bruta:**
  - **Asimetría y Curtosis:** Indican la distribución de los ingresos en taquilla.
  - **Media, Mediana, Desviación Estándar:** Muestran la tendencia central y variabilidad en los ingresos.



Asimetría: 3.1301343288134538

Curtosis: 13.914472951089826

Media: 68034750.87364621

Mediana: 23530892.0

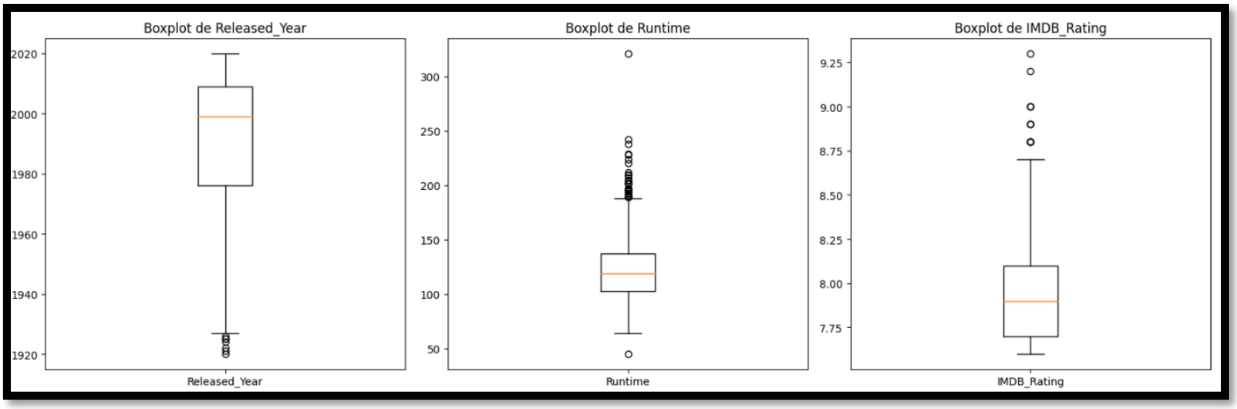
Desviación estándar: 109750043.19218515

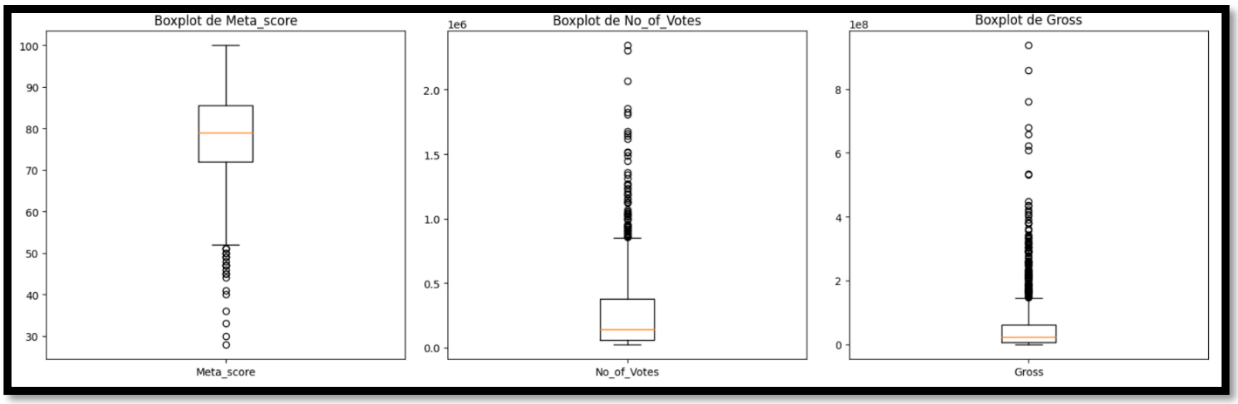
Este análisis proporciona una comprensión integral de las distribuciones y características de las variables numéricas en el conjunto de datos, permitiendo identificar patrones y tendencias que pueden ser útiles para análisis más avanzados.

3.3. Manejo de Outliers

El manejo de outliers es crucial para asegurar que los resultados del análisis y modelado no se vean distorsionados por valores atípicos. Los outliers pueden influir desproporcionadamente en estadísticas descriptivas como la media y pueden afectar negativamente el rendimiento de los modelos predictivos.

Primero, se identificaron las columnas numéricas del conjunto de datos y se visualizaron mediante boxplots para detectar posibles outliers.

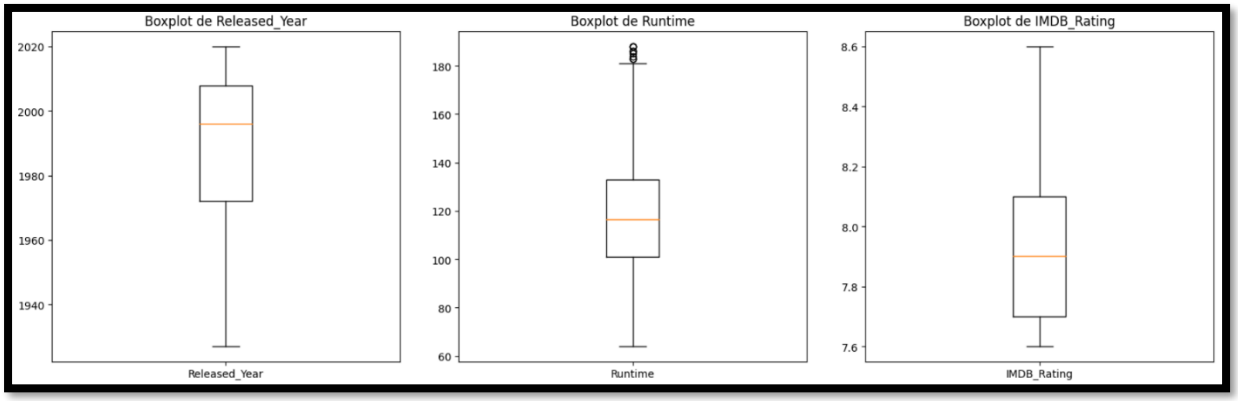


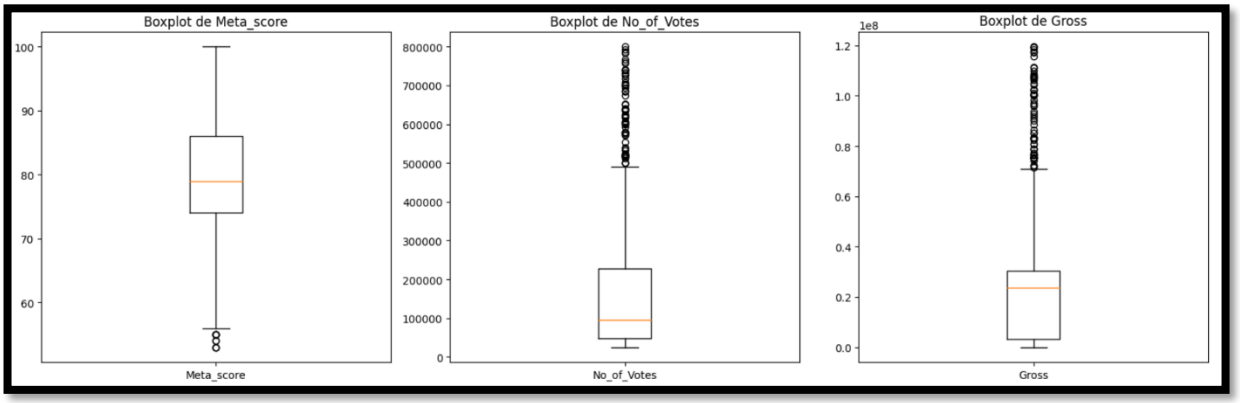


3.4. Método IQR

El Método del Rango Intercuartílico (IQR) se utiliza para identificar y manejar valores atípicos de manera sistemática y robusta. El IQR calcula la dispersión en el centro del conjunto de datos (entre el primer y el tercer cuartil) y define límites para detectar valores extremos que podrían ser errores o anomalías.

Se implementó una función para calcular el IQR y establecer límites inferiores y superiores para cada columna numérica. Los datos que caían fuera de estos límites fueron eliminados, lo que ayuda a garantizar que los análisis y modelos se basen en datos representativos y no en valores atípicos extremos.





3.5. Transformación de Datos

La transformación de datos tiene el objetivo de simplificar y estructurar el conjunto de datos para que sea más adecuado para el análisis y modelado. Incluye la eliminación de columnas irrelevantes y la creación de nuevas variables que mejoran la capacidad de análisis.

3.5.1. **Eliminación de Columnas Irrelevantes:** Se eliminaron columnas como 'Poster_Link', 'Series_Title' y 'Overview' que no aportaban valor significativo para el análisis. Esto redujo la complejidad del conjunto de datos.

```
processed_data = df_no_outliners.drop(['Poster_Link', 'Series_Title', 'Overview'], axis = 1).copy()
```

3.5.2. **Categorías de Directores:** Se agrupó a los directores en categorías basadas en la media de sus puntuaciones en Metacritic. Esto facilita el análisis al convertir un dato continuo en una variable categórica más manejable.

```
# Agrupar por director y calcular la media de "Meta_score"
directors_meta_score = processed_data.groupby("Director")["Meta_score"].mean().to_frame()

# Crear categorías usando qcut
directors_meta_score['Category'] = pd.qcut(directors_meta_score["Meta_score"], q=4, labels=[4, 3, 2, 1])

# Mapear las categorías de vuelta al DataFrame original
processed_data["Director_Category"] = processed_data["Director"].map(directors_meta_score["Category"])
```

3.5.3. **Transformación de Género y Certificación:** Se extrajo el primer género listado y se crearon variables dummy para el género y las categorías de certificación. Esto convierte las variables categóricas en un formato que los modelos de aprendizaje automático pueden utilizar más fácilmente.

```
# Tomar solo el primer género
processed_data["Genre"] = processed_data.apply(lambda row: row["Genre"].split(",")[0], axis=1)

# Asegurarse de que 'Genre' sea de tipo str
processed_data['Genre'] = processed_data['Genre'].astype(str)

# Crear variables dummy para la columna 'Genre'
var = ['Genre']
dummy_genre = pd.get_dummies(processed_data[var], drop_first=True)

# Convertir explícitamente las columnas dummy a enteros
dummy_genre = dummy_genre.astype(int)
```

```
# Crear una nueva característica agrupando categorías
def categorize_certificate(cert):
    if cert in ['A', 'R', 'TV-MA']:
        return 'Adult'
    elif cert in ['UA', 'U/A', 'PG-13', '16', 'TV-14']:
        return 'Teen'
    elif cert in ['U', 'PG', 'G', 'Passed', 'GP', 'Approved', 'TV-PG']:
        return 'General'
    else:
        return 'Unrated'

processed_data['Certificate_Category'] = processed_data['Certificate'].apply(categorize_certificate)
```

```
var = ['Certificate_Category']
dummy_certificate = pd.get_dummies(processed_data[var])

# Convertir explícitamente las columnas dummy a enteros
dummy_certificate = dummy_certificate.astype(int)
```

```
processed_final = pd.concat([processed_data[["Runtime","IMDB_Rating","Certificate_Category",
 "Released_Year","Director","No_of_Votes","Meta_score","Gross"]], dummy_genre, dummy_certificate], axis=1)
```

3.6. Normalización

La normalización de datos es esencial para asegurar que todas las características numéricas tengan la misma escala, lo que facilita la comparación y mejora el rendimiento de los modelos de aprendizaje automático. Sin normalización, las características con rangos más amplios pueden dominar el proceso de modelado.

Se aplicó la técnica de **MinMaxScaler** para escalar las columnas numéricas al **rango [0, 1].** Esto transforma los datos en una escala uniforme, lo que es especialmente útil para algoritmos que son sensibles a la escala de las características, como la regresión lineal y las redes neuronales.

	Runtime	IMDB_Rating	Released_Year	No_of_Votes	Meta_score	Gross
0	0.774194	1.0	1.000000	0.038986	0.787234	0.196857
1	0.548387	1.0	0.989247	0.681148	0.914894	0.446505
2	0.717742	1.0	1.000000	0.038604	0.553191	0.196857
3	0.532258	1.0	0.806452	0.870223	0.553191	0.063256
4	0.491935	1.0	0.795699	0.808419	0.914894	0.084111

3.6.1. Combinar Variables Normalizadas con Variables Dummy:

Después de normalizar las columnas numéricas, se combinan estas columnas normalizadas con las variables dummy creadas anteriormente para los géneros y las categorías de certificación. Esto se realiza utilizando la función **pd.concat**, que concatena los DataFrames a lo largo del eje de columnas (axis=1). Esta combinación es crucial para incluir todas las características relevantes en el

conjunto de datos final, asegurando que tanto las variables numéricas como las categóricas transformadas estén presentes.

```
df_numeric_scaled = pd.concat([ df_numeric_scaled, dummy_genre, dummy_certificate], axis=1)
```

3.6.2. Rellenar Valores Nulos:

Se utiliza el método fillna(0, inplace=True) para rellenar cualquier valor nulo en el conjunto de datos combinado con ceros. Este paso es importante para garantizar que no haya valores faltantes que puedan causar problemas durante el análisis o modelado posterior. Rellenar los valores nulos con ceros asegura que el DataFrame esté completo y listo para su uso.

```
df_numeric_scaled.fillna(0, inplace=True)
```

3.6.3. Revisar Información del Conjunto de Datos Final:

Finalmente, se utiliza **df_numeric_scaled.info()** para revisar la información del conjunto de datos final. Este método proporciona un resumen del DataFrame, incluyendo el número de entradas, nombres de columnas, tipos de datos y el número de valores no nulos. Esta revisión ayuda a verificar que todas las transformaciones se han aplicado correctamente y que el DataFrame está listo para su uso en análisis y modelado.

```
df_numeric_scaled.info()
```

```
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Runtime                       957 non-null    float64
 1   IMDB_Rating                   957 non-null    float64
 2   Released_Year                 957 non-null    float64
 3   No_of_Votes                   957 non-null    float64
 4   Meta_score                    957 non-null    float64
 5   Gross                         957 non-null    float64
 6   Genre_Adventure               957 non-null    float64
 7   Genre_Animation               957 non-null    float64
 8   Genre_Biography               957 non-null    float64
 9   Genre_Comedy                  957 non-null    float64
 10  Genre_Crime                   957 non-null    float64
 11  Genre_Drama                   957 non-null    float64
 12  Genre_Family                  957 non-null    float64
 13  Genre_Film-Noir               957 non-null    float64
 14  Genre_Horror                  957 non-null    float64
 15  Genre_Mystery                 957 non-null    float64
 16  Genre_Thriller                957 non-null    float64
 17  Genre_Western                 957 non-null    float64
 18  Certificate_Category_Adult    957 non-null    float64
 19  Certificate_Category_General  957 non-null    float64
 20  Certificate_Category_Teen     957 non-null    float64
 21  Certificate_Category_Unrated  957 non-null    float64
dtypes: float64(22)
```

4. Modelado Matemático:

En esta sección, abordamos la etapa de modelado matemático utilizando el **DataFrame df_numeric_scaled,** que contiene nuestras características preparadas y normalizadas. El objetivo es construir modelos predictivos para entender mejor las relaciones en los datos y hacer predicciones basadas en las características disponibles.

Para comenzar con el modelado matemático, primero necesitamos preparar nuestros datos. A continuación, se describen los pasos iniciales:

Definición de Variables: Extraemos las variables que usaremos para el modelado. La variable Y representa la etiqueta que queremos predecir, en este caso, la calificación de IMDB. La variable X contiene todas las características de entrada que se usarán para hacer la predicción, excluyendo la calificación de IMDB.

```
Y = df_numeric_scaled['IMDB_Rating'].values
X = df_numeric_scaled.drop('IMDB_Rating', axis=1).values

# Inicializar los parametros
W = 0
b = 0
```

En esta etapa, normalizamos tanto las características como la variable objetivo para asegurar que todas las variables tengan una escala comparable y mejorar la estabilidad y rendimiento del modelo. La normalización se realiza en dos pasos:

Normalización de las Características: Calculamos la media y la desviación estándar de cada característica en X, y luego usamos estos valores para escalar las características. Esto asegura que cada característica tenga una media de 0 y una desviación estándar de 1.

```
# Normalizar las características
X_mean = np.mean(X, axis=0)
X_std = np.std(X, axis=0)
X = (X - X_mean) / X_std

# Normalizar la variable objetivo
Y_mean = np.mean(Y)
Y_std = np.std(Y)
Y = (Y - Y_mean) / Y_std
```

Una vez que los datos están normalizados, pasamos a la inicialización de los parámetros del modelo. En este paso:

Determinar el Tamaño de los Datos: m representa el número de muestras (filas) en el conjunto de datos X, y n el número de características (columnas). Estos valores se obtienen a partir de la forma de X.

```
# Inicializar los parámetros
m, n = X.shape
W = np.zeros(n)
b = 0
```

La función de costo, también conocida como función de pérdida, mide qué tan bien se ajusta el modelo a los datos. En el contexto de la regresión lineal, la función de costo que usamos es **el error cuadrático medio (Mean Squared Error, MSE),** ajustado para ser el error cuadrático medio dividido por dos.

```
# Definir la función de costo
def compute_cost(X, Y, W, b):
    m = len(Y)
    predictions = np.dot(X, W) + b
    cost = np.sum((predictions - Y) ** 2) / (2 * m)
    return cost
```

La función **gradient_descent** implementa el algoritmo de descenso de gradiente, que se utiliza para minimizar la función de costo y ajustar los parámetros del modelo. Este proceso se realiza a través de una serie de iteraciones, actualizando los parámetros en la dirección que reduce el costo.

```python
def gradient_descent(X, Y, W, b, learning_rate, iterations):
    m = len(Y)
    for i in range(iterations):
        # Calcular las predicciones
        predictions = np.dot(X, W) + b

        # Calcular las derivadas
        dW = (1/m) * np.dot(X.T, (predictions - Y))
        db = (1/m) * np.sum(predictions - Y)

        # Actualizar los parámetros
        W = W - learning_rate * dW
        b = b - learning_rate * db

        # Calcular el costo
        cost = compute_cost(X, Y, W, b)

        # Imprimir el costo cada 100 iteraciones
        if i % 100 == 0:
            print(f"Iteración {i}: Costo {cost}")

    return W, b
```

En esta etapa, configuramos los hiperparámetros necesarios para el algoritmo de descenso de gradiente, entrenamos el modelo y verificamos el costo final.

Definir los Hiperparámetros:

Tasa de Aprendizaje (learning_rate): Es un valor que determina el tamaño del paso que damos en cada iteración para actualizar los parámetros del modelo. Un valor comúnmente utilizado es 0.01.

Número de Iteraciones (iterations): Define cuántas veces el algoritmo de descenso de gradiente actualizará los parámetros. En este caso, se establece en 1000 iteraciones.

```python
# Definir los hiperparámetros
learning_rate = 0.01
iterations = 1000

# Entrenar el modelo
W, b = gradient_descent(X, Y, W, b, learning_rate, iterations)
print(f"W: {W}")
print(f"b: {b}")

# Visualización del modelo
# No es posible visualizar en 2D con múltiples características, pero podemos verificar el costo final
final_cost = compute_cost(X, Y, W, b)
print(f"Costo final: {final_cost}")
```

```
Iteración 0: Costo 0.4916232849634562
Iteración 100: Costo 0.2894616147851982
Iteración 200: Costo 0.2657828091247401
Iteración 300: Costo 0.25705285565288327
Iteración 400: Costo 0.25327173556142796
Iteración 500: Costo 0.25149466497916984
Iteración 600: Costo 0.2506041447009601
Iteración 700: Costo 0.2501291121203962
Iteración 800: Costo 0.24985812129159057
Iteración 900: Costo 0.24969185562854274
W: [ 0.38229947 -0.16500887  0.36587049  0.36216975 -0.26089349 -0.00449507
 -0.00944578 -0.00940437 -0.03873946 -0.02027411 -0.01322609 -0.04688387
 -0.01938339  0.0199883   0.01579372 -0.027732    0.0176442  -0.12418409
 -0.13171445 -0.07760595 -0.02660314]
b: 4.573608414088003e-17
Costo final: 0.24958283572926135
```

En esta etapa, calculamos las predicciones del modelo, revertimos la normalización de la variable objetivo para obtener los valores originales, y evaluamos el desempeño del modelo utilizando el RMSE (Error Cuadrático Medio de la Raíz).

Calcular las Predicciones:

Usamos el modelo entrenado para hacer predicciones sobre el conjunto de datos. Las predicciones se obtienen aplicando el modelo (producto punto entre las características X y los pesos W, más el sesgo b).

```python
# Calcular las predicciones
y_pred = np.dot(X, W) + b

# Deshacer la normalización para la variable objetivo
y_pred = y_pred * Y_std + Y_mean

# Calcular el RMSE
def rmse(y_true, y_pred):
    return np.sqrt(np.mean((y_true - y_pred) ** 2))

# Calcular el RMSE en el conjunto de datos original
error = rmse(df_numeric_scaled['IMDB_Rating'], y_pred)
print(f"RMSE del modelo: {error}")
```

RMSE del modelo: 0.1735338110521041

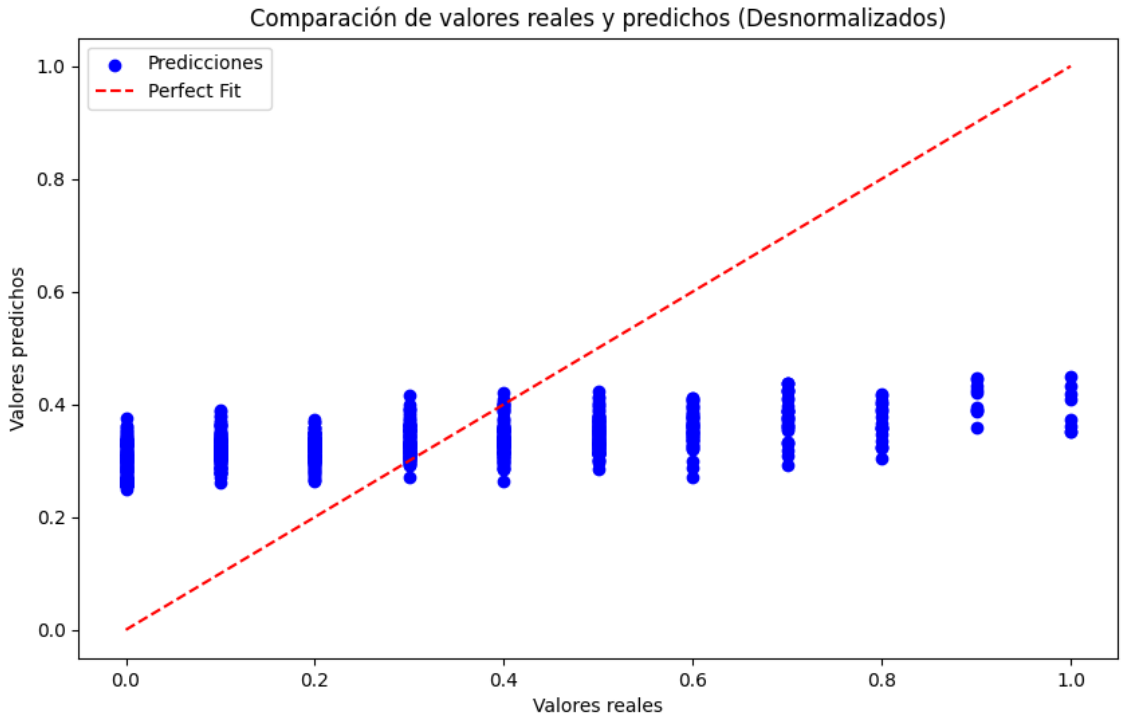
Para evaluar visualmente el rendimiento del modelo, comparamos los valores reales con las predicciones desnormalizadas a través de un gráfico de dispersión. Esto ayuda a entender cómo se ajustan las predicciones del modelo a los valores verdaderos.

Deshacer la Normalización de las Predicciones:

Las predicciones y_pred están en una escala normalizada, por lo que necesitamos revertir esta normalización para que las predicciones estén en la misma escala que los valores originales de IMDB_Rating. Usamos la media y la desviación estándar originales para esta desnormalización

```
# Deshacer la normalización de las predicciones
y_pred_desnormalized = y_pred * Y_std + Y_mean

# Gráfico de valores reales vs valores predichos desnormalizados
plt.figure(figsize=(10, 6))
plt.scatter(df_numeric_scaled['IMDB_Rating'], y_pred_desnormalized, color='blue', label='Predicciones')
plt.plot([df_numeric_scaled['IMDB_Rating'].min(), df_numeric_scaled['IMDB_Rating'].max()],
         [df_numeric_scaled['IMDB_Rating'].min(), df_numeric_scaled['IMDB_Rating'].max()],
         color='red', linestyle='--', label='Perfect Fit')
plt.xlabel('Valores reales')
plt.ylabel('Valores predichos')
plt.title('Comparación de valores reales y predichos (Desnormalizados)')
plt.legend()
plt.show()
```



4. Resultados y Conclusiones:

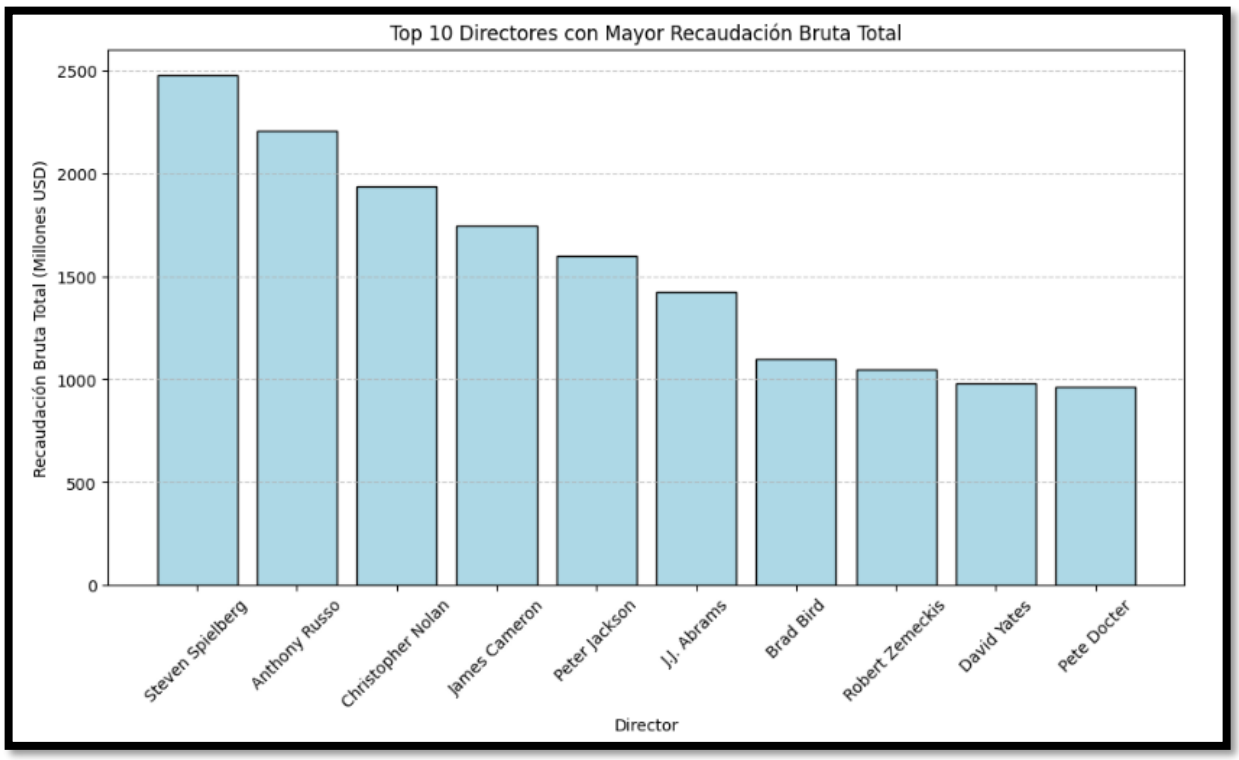
Análisis de la Recaudación Bruta de una Película vs. Directores

Este análisis explora cómo la recaudación bruta de una película se relaciona con el director. El objetivo es identificar qué directores tienden a generar mayores ingresos en taquilla.

Pregunta 1: ¿Qué directores tienden a generar mayores ingresos en taquilla?

Top 10 Directores con Mayor Recaudación Bruta Total:

	Director	Recaudacion bruta	Recaudacion bruta (Millones)
470	Steven Spielberg	2.478133e+09	2,478.13
36	Anthony Russo	2.205039e+09	2,205.04
83	Christopher Nolan	1.937454e+09	1,937.45
202	James Cameron	1.748237e+09	1,748.24
383	Peter Jackson	1.597312e+09	1,597.31
195	J.J. Abrams	1.423171e+09	1,423.17
58	Brad Bird	1.099628e+09	1,099.63
426	Robert Zemeckis	1.049446e+09	1,049.45
107	David Yates	9.789537e+08	978.95
380	Pete Docter	9.629130e+08	962.91



Los directores que tienden a generar mayores ingresos en taquilla son aquellos que se destacan en la industria cinematográfica con una larga trayectoria y éxitos consistentes. Los gráficos muestran claramente que los géneros preferidos por estos directores también contribuyen significativamente a su éxito financiero.

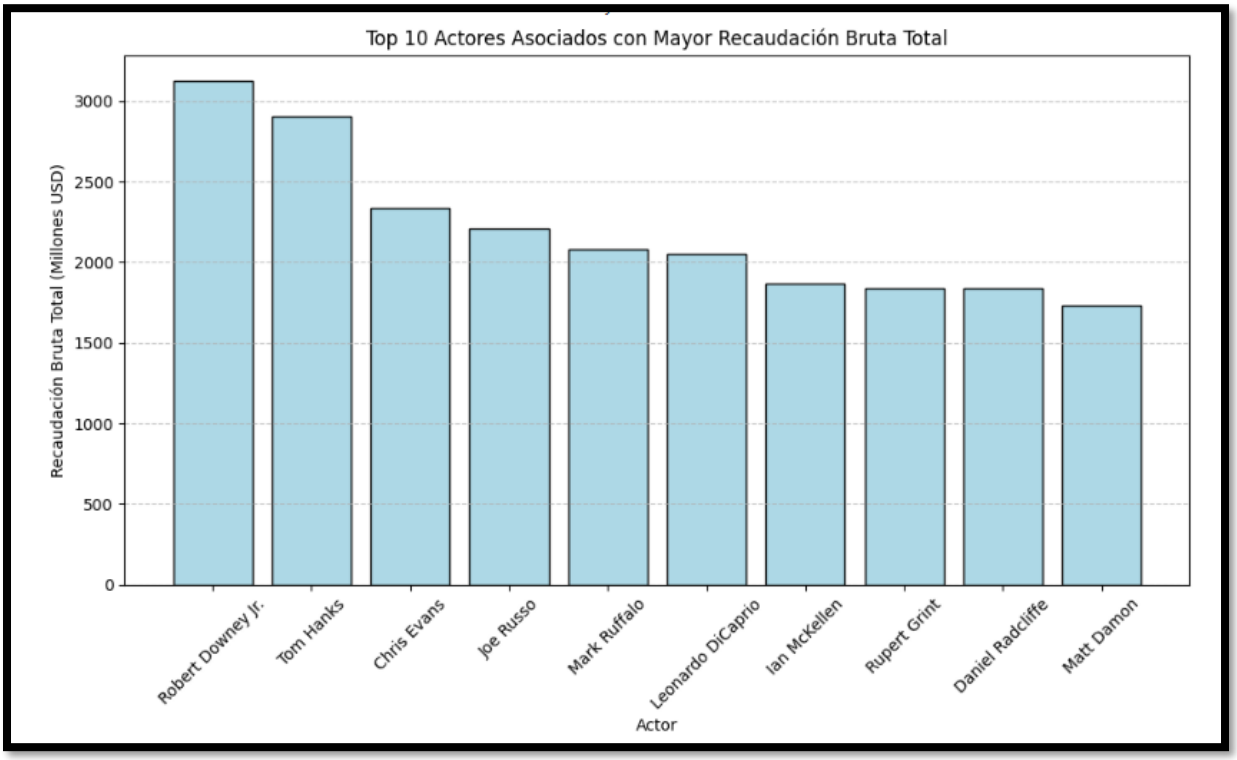
Análisis de la Recaudación Bruta de una Película vs. Diferentes Actores

En este análisis, se examina cómo la recaudación bruta de una película está asociada con los diferentes actores que participan en ella. Se busca determinar qué actores están asociados con películas que generan mayores ingresos.

Pregunta 2: ¿Qué actores están asociados con películas que generan mayores ingresos?

Top 10 Actores Asociados con Mayor Recaudación Bruta Total:

	Actor	Recaudacion bruta	Recaudacion bruta (Millones)
2137	Robert Downey Jr.	3.129073e+09	3,129.07
2498	Tom Hanks	2.903565e+09	2,903.56
427	Chris Evans	2.339664e+09	2,339.66
1205	Joe Russo	2.205039e+09	2,205.04
1655	Mark Ruffalo	2.081926e+09	2,081.93
1511	Leonardo DiCaprio	2.049297e+09	2,049.30
991	Ian McKellen	1.869869e+09	1,869.87
2202	Rupert Grint	1.835901e+09	1,835.90
515	Daniel Radcliffe	1.835901e+09	1,835.90
1696	Matt Damon	1.728542e+09	1,728.54



Los actores que están asociados con mayores ingresos en taquilla son aquellos que tienen una gran popularidad y una amplia gama de géneros en su filmografía. Los gráficos circulares muestran que los géneros en los que estos actores trabajan con mayor frecuencia también contribuyen significativamente a los ingresos.

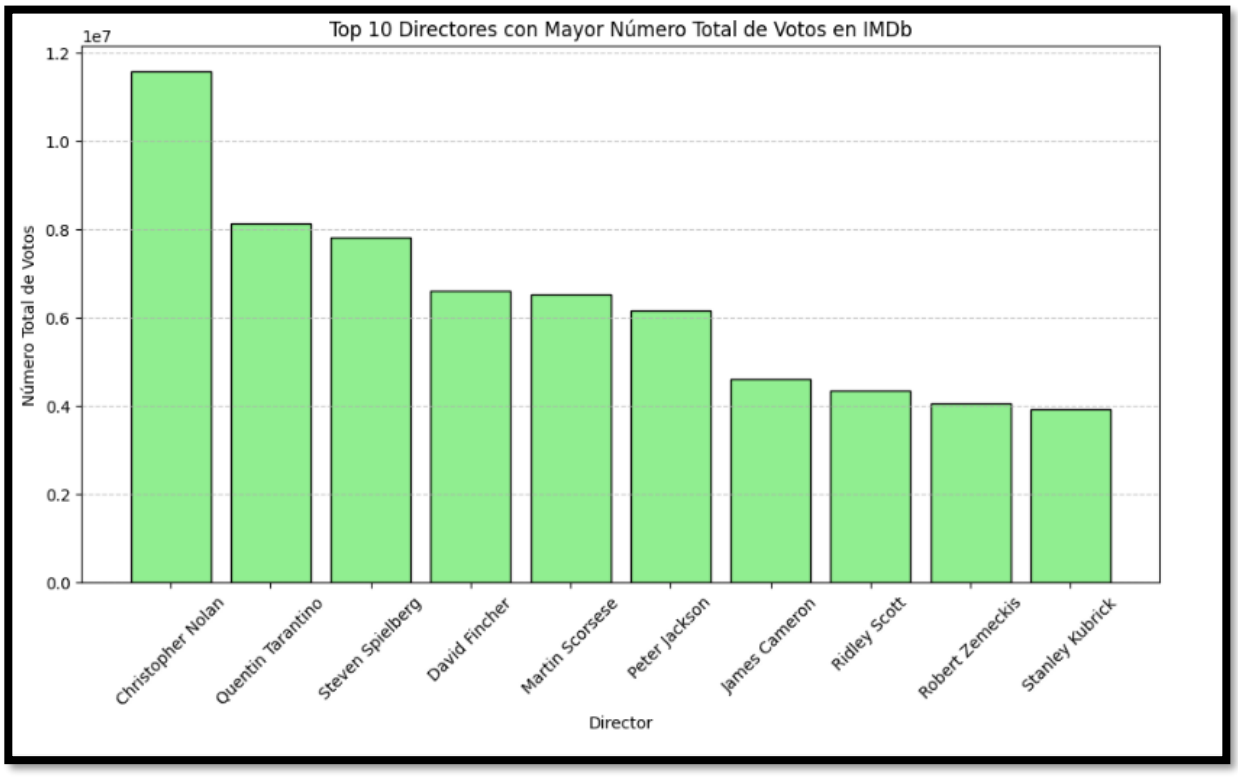
Análisis del Número de Votos de una Película vs. Directores

Este análisis investiga la relación entre el número de votos que recibe una película en IMDB y el director de la película. El objetivo es identificar qué directores tienden a recibir más votos.

Pregunta 3: ¿Qué directores tienden a recibir más votos en IMDB?

Top 10 Directores con Mayor Número Total de Votos:

	Director	Numero de votos
83	Christopher Nolan	11,578,345
391	Quentin Tarantino	8,123,208
470	Steven Spielberg	7,817,166
100	David Fincher	6,607,859
313	Martin Scorsese	6,513,530
383	Peter Jackson	6,148,579
202	James Cameron	4,613,107
411	Ridley Scott	4,339,890
426	Robert Zemeckis	4,055,464
463	Stanley Kubrick	3,919,254



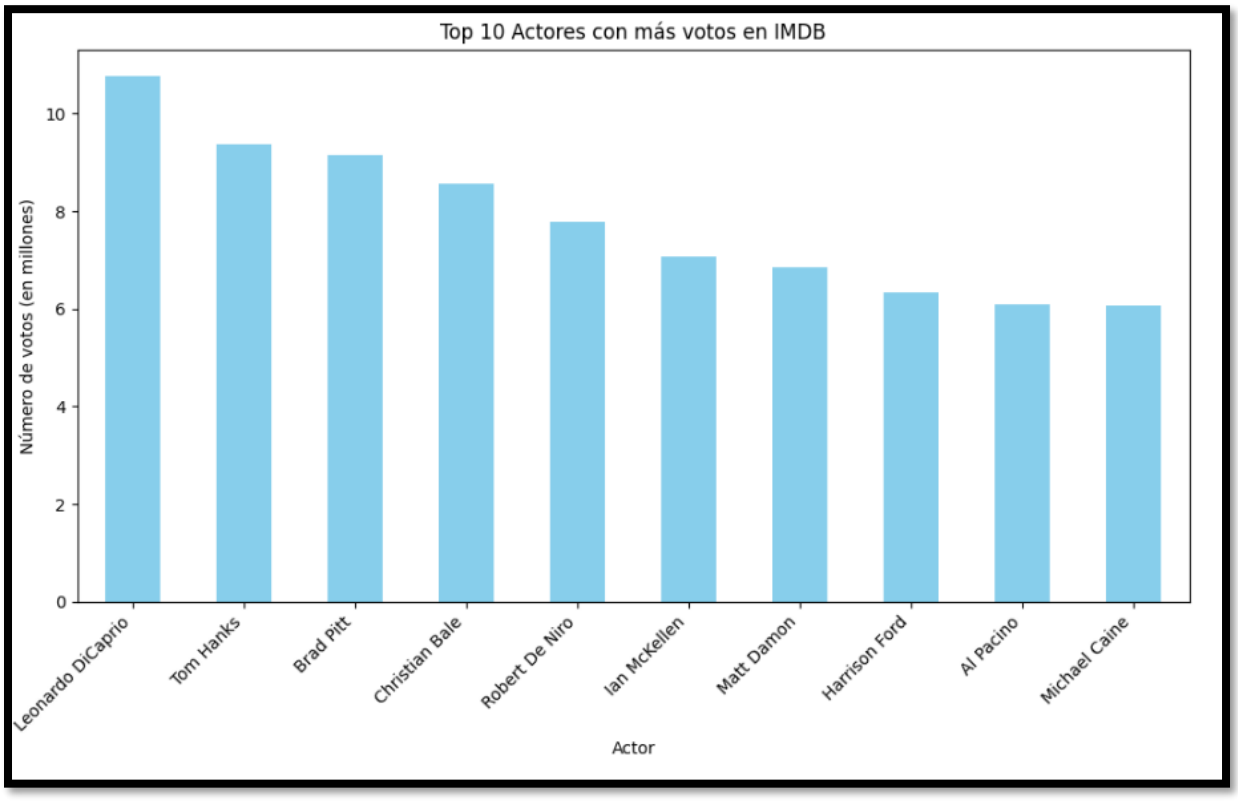
Los directores que tienden a recibir más votos en IMDB son aquellos con una gran base de seguidores y películas populares. Los gráficos muestran que la mediana de votos varía según el género, indicando que ciertos géneros son más propensos a recibir altos números de votos.

Análisis del Número de Votos de una Película vs. Diferentes Actores

En este análisis, se explora cómo el número de votos de una película está relacionado con los actores que participan en ella. Se busca identificar qué actores están asociados con películas que reciben más votos.

Pregunta 4: ¿Qué actores están asociados con películas que reciben más votos?

```
Actor
Leonardo DiCaprio    10.782195
Tom Hanks             9.374159
Brad Pitt             9.144728
Christian Bale        8.565922
Robert De Niro        7.787234
Ian McKellen          7.063378
Matt Damon            6.840383
Harrison Ford         6.341070
Al Pacino             6.098413
Michael Caine         6.064485
Name: Numero de votos, dtype: float64
```



Los actores asociados con películas que reciben más votos son aquellos con una carrera establecida y una fuerte presencia en géneros populares. El análisis muestra que ciertos géneros, como el drama y la acción, son más propensos a recibir altos números de votos, lo que refleja las preferencias del público.

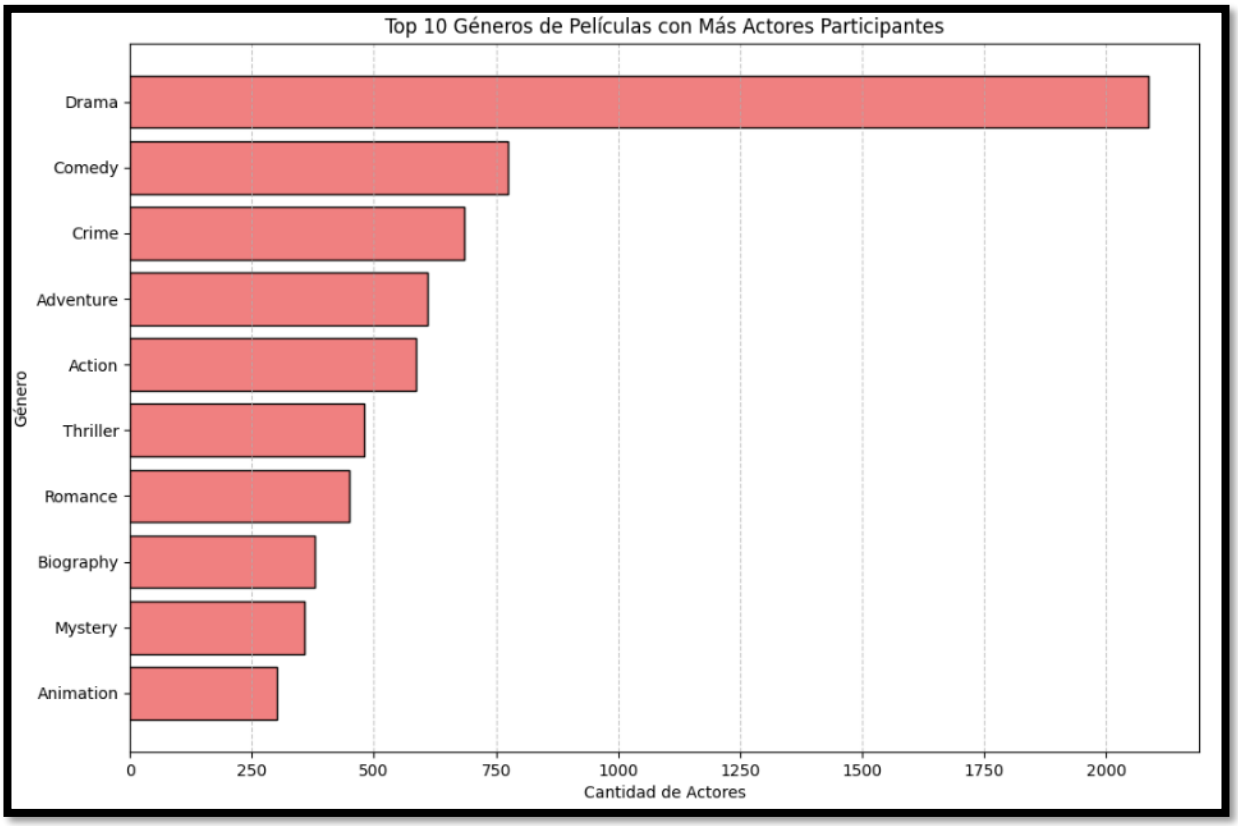
Preferencia de Géneros de las Películas por Parte de los Actores

Este análisis examina las preferencias de los actores por diferentes géneros cinematográficos. Se analiza qué géneros de películas son los que los actores prefieren.

Pregunta 5: ¿Qué géneros de películas son los que los actores prefieren?

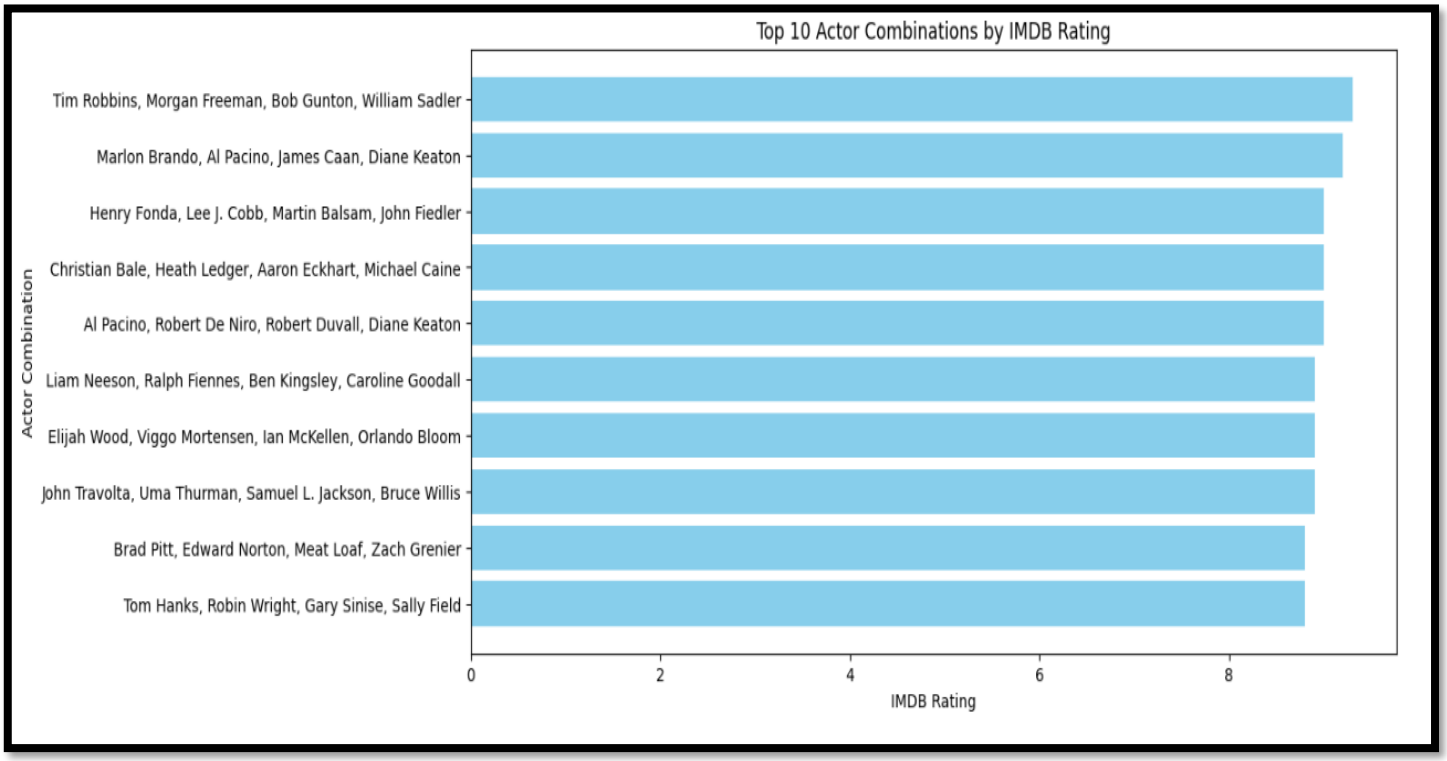
Top 10 Géneros con Más Actores Participantes:

	Genero	Cantidad de Actores
6	Drama	2086
4	Comedy	775
5	Crime	686
1	Adventure	609
0	Action	586
18	Thriller	480
15	Romance	449
3	Biography	379
14	Mystery	357
2	Animation	301



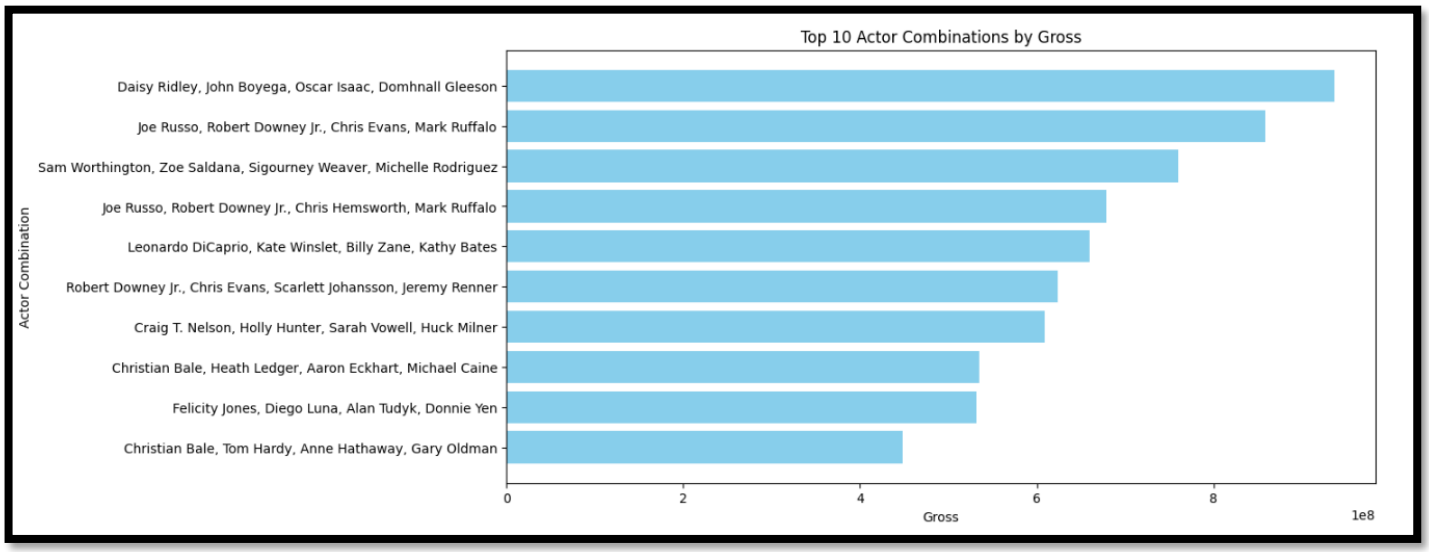
Los directores y actores que están asociados con películas de altos ratings son aquellos con una trayectoria destacada y una alta calidad en sus producciones. El análisis revela que los directores y actores que consistentemente reciben altos ratings tienden a trabajar en géneros que son apreciados tanto por la crítica como por el público.

Pregunta 6: ¿Qué combinación de actores (Star1, Star2, Star3 y Star4) está obteniendo buenas calificaciones en IMDB la mayor parte del tiempo?



Las combinaciones de actores que tienden a recibir las mejores calificaciones en IMDB incluyen a actores muy reconocidos y talentosos, quienes probablemente atraen tanto a críticos como a audiencias con sus interpretaciones.

Pregunta 7: ¿Qué combinación de actores (Star1, Star2, Star3 y Star4) está obteniendo buenos ingresos en taquilla?



Las combinaciones de actores que tienden a generar los mayores ingresos en taquilla incluyen a actores que son muy populares y tienen una gran base de fanáticos. Estas combinaciones suelen estar en películas de alto presupuesto y con gran promoción, lo que contribuye a sus altos ingresos.

4.1. CONCLUSION

En conclusión, en este proyecto se analizó el dataset de las mejores 1000 películas y series de IMDB para identificar patrones y relaciones entre variables clave. Primeramente, se realizó una limpieza y exploración inicial de los datos, seguida de un análisis descriptivo y visualizaciones para entender mejor las distribuciones y relaciones entre las variables. Por ello, los resultados revelaron que ciertos directores y actores están asociados con mayores ingresos en taquilla y mejores calificaciones en IMDB. Específicamente, se identificaron las combinaciones de actores que tienden a recibir las mejores calificaciones en IMDB y generar los mayores ingresos en taquilla. Así que, estos hallazgos proporcionan información valiosa para la toma de decisiones en la industria del cine y la televisión, destacando la importancia de la elección de elenco y dirección en el éxito crítico y comercial de una producción.