In [1]:
```python
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import matplotlib.pyplot as plt # plotting
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
```

In [2]:
```python
path = "movie_success_rate.csv"

df = pd.read_csv(path)

df.head(5)
```

Out[2]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	...	Music	Musical	Mystery	Romance	Sci-Fi	Sport	Thriller	War	Western	Success
0	1.0	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014.0	121.0	8.1	757074.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
1	2.0	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012.0	124.0	7.0	485820.0	...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
2	3.0	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016.0	117.0	7.3	157606.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
3	4.0	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016.0	108.0	7.2	60545.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	5.0	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016.0	123.0	6.2	393727.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 33 columns

In [3]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 839 entries, 0 to 838
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Rank               838 non-null    float64
 1   Title              838 non-null    object 
 2   Genre              838 non-null    object 
 3   Description        838 non-null    object 
 4   Director           838 non-null    object 
 5   Actors             838 non-null    object 
 6   Year               838 non-null    float64
 7   Runtime (Minutes)  838 non-null    float64
 8   Rating             839 non-null    float64
 9   Votes              839 non-null    float64
 10  Revenue (Millions) 839 non-null    float64
 11  Metascore          838 non-null    float64
 12  Action             838 non-null    float64
 13  Adventure          838 non-null    float64
 14  Aniimation         838 non-null    float64
 15  Biography          838 non-null    float64
 16  Comedy             838 non-null    float64
 17  Crime              838 non-null    float64
 18  Drama              838 non-null    float64
 19  Family             838 non-null    float64
 20  Fantasy            838 non-null    float64
 21  History            838 non-null    float64
 22  Horror             838 non-null    float64
 23  Music              838 non-null    float64
 24  Musical            838 non-null    float64
 25  Mystery            838 non-null    float64
 26  Romance            838 non-null    float64
 27  Sci-Fi             838 non-null    float64
 28  Sport              838 non-null    float64
 29  Thriller           838 non-null    float64
 30  War                838 non-null    float64
 31  Western            838 non-null    float64
 32  Success            838 non-null    float64
dtypes: float64(28), object(5)
memory usage: 216.4+ KB
```

In [4]:
```python
## Check the nulls
df.isna().sum()
```

Out[4]:
```
Rank                  1
Title                 1
Genre                 1
Description           1
Director              1
Actors                1
Year                  1
Runtime (Minutes)     1
Rating                0
Votes                 0
Revenue (Millions)    0
Metascore             1
Action                1
Adventure             1
Aniimation            1
Biography             1
Comedy                1
Crime                 1
Drama                 1
Family                1
Fantasy               1
History               1
Horror                1
Music                 1
Musical               1
Mystery               1
Romance               1
Sci-Fi                1
Sport                 1
Thriller              1
War                   1
Western               1
Success               1
dtype: int64
```

In [5]:
```python
df.dropna(how='any',inplace=True)
print(df)
```

```
       Rank                    Title                     Genre  \
0       1.0  Guardians of the Galaxy   Action,Adventure,Sci-Fi   
1       2.0               Prometheus  Adventure,Mystery,Sci-Fi   
2       3.0                    Split           Horror,Thriller   
3       4.0                     Sing   Animation,Comedy,Family   
4       5.0            Suicide Squad  Action,Adventure,Fantasy   
..      ...                      ...                       ...   
833   994.0 Resident Evil: Afterlife   Action,Adventure,Horror   
834   995.0                Project X                    Comedy   
835   997.0           Hostel: Part II                   Horror   
836   998.0   Step Up 2: The Streets       Drama,Music,Romance   
837  1000.0               Nine Lives     Comedy,Family,Fantasy   

                                           Description             Director  \
0    A group of intergalactic criminals are forced ...           James Gunn   
1    Following clues to the origin of mankind, a te...         Ridley Scott   
2    Three girls are kidnapped by a man with a diag...   M. Night Shyamalan   
3    In a city of humanoid animals, a hustling thea...  Christophe Lourdelet   
4    A secret government agency recruits some of th...           David Ayer   
..                                                 ...                  ...   
833  While still out to destroy the evil Umbrella C...    Paul W.S. Anderson   
834  3 high school seniors throw a birthday party t...       Nima Nourizadeh   
835  Three American college students studying abroa...             Eli Roth   
836  Romantic sparks occur between two dance studen...           Jon M. Chu   
837  A stuffy businessman finds himself trapped ins...      Barry Sonnenfeld   

                                                Actors    Year  \
0    Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...  2014.0   
1    Noomi Rapace, Logan Marshall-Green, Michael Fa...  2012.0   
2    James McAvoy, Anya Taylor-Joy, Haley Lu Richar...  2016.0   
3    Matthew McConaughey,Reese Witherspoon, Seth Ma...  2016.0   
4    Will Smith, Jared Leto, Margot Robbie, Viola D...  2016.0   
..                                                 ...     ...   
833  Milla Jovovich, Ali Larter, Wentworth Miller,K...  2010.0   
834  Thomas Mann, Oliver Cooper, Jonathan Daniel Br...  2012.0   
835  Lauren German, Heather Matarazzo, Bijou Philli...  2007.0   
836  Robert Hoffman, Briana Evigan, Cassie Ventura,...  2008.0   
837  Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...  2016.0   

     Runtime (Minutes)  Rating     Votes  ...  Music  Musical  Mystery  \
0                121.0     8.1  757074.0  ...    0.0      0.0      0.0   
1                124.0     7.0  485820.0  ...    0.0      0.0      1.0   
2                117.0     7.3  157606.0  ...    0.0      0.0      0.0   
3                108.0     7.2   60545.0  ...    0.0      0.0      0.0   
4                123.0     6.2  393727.0  ...    0.0      0.0      0.0   
..                 ...     ...       ...  ...    ...      ...      ...   
833               97.0     5.9  140900.0  ...    0.0      0.0      0.0   
834               88.0     6.7  164088.0  ...    0.0      0.0      0.0   
835               94.0     5.5   73152.0  ...    0.0      0.0      0.0   
836               98.0     6.2   70699.0  ...    1.0      0.0      0.0   
837               87.0     5.3   12435.0  ...    0.0      0.0      0.0   

     Romance  Sci-Fi  Sport  Thriller  War  Western  Success  
0        0.0     1.0    0.0       0.0  0.0      0.0      1.0  
1        0.0     1.0    0.0       0.0  0.0      0.0      1.0  
2        0.0     0.0    0.0       1.0  0.0      0.0      0.0  
3        0.0     0.0    0.0       0.0  0.0      0.0      0.0  
4        0.0     0.0    0.0       0.0  0.0      0.0      0.0  
..       ...     ...    ...       ...  ...      ...      ...  
833      0.0     0.0    0.0       0.0  0.0      0.0      0.0  
834      0.0     0.0    0.0       0.0  0.0      0.0      0.0  
835      0.0     0.0    0.0       0.0  0.0      0.0      0.0  
836      1.0     0.0    0.0       0.0  0.0      0.0      0.0  
837      0.0     0.0    0.0       0.0  0.0      0.0      0.0  

[838 rows x 33 columns]
```

In [6]:
```python
df.shape
```

Out[6]:
```
(838, 33)
```

In [7]:
```python
df.columns
```

Out[7]:
```
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'Action', 'Adventure', 'Aniimation', 'Biography', 'Comedy',
       'Crime', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Music',
       'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'War',
       'Western', 'Success'],
      dtype='object')
```

In [8]:
```python
df.dtypes
```

Out[8]:
```
Rank                  float64
Title                  object
Genre                  object
Description            object
Director               object
Actors                 object
Year                  float64
Runtime (Minutes)     float64
Rating                float64
Votes                 float64
Revenue (Millions)    float64
Metascore             float64
Action                float64
Adventure             float64
Aniimation            float64
Biography             float64
Comedy                float64
Crime                 float64
Drama                 float64
Family                float64
Fantasy               float64
History               float64
Horror                float64
Music                 float64
Musical               float64
Mystery               float64
Romance               float64
Sci-Fi                float64
Sport                 float64
Thriller              float64
War                   float64
Western               float64
Success               float64
dtype: object
```

In [9]:
```python
df.isnull()
```

Out[9]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	...	Music	Musical	Mystery	Romance	Sci-Fi	Sport	Thriller	War	Western	Success
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
833	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
834	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
835	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
836	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
837	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

838 rows × 33 columns

In [10]:
```python
df.isnull().any()
```

Out[10]:
```
Rank                  False
Title                 False
Genre                 False
Description           False
Director              False
Actors                False
Year                  False
Runtime (Minutes)     False
Rating                False
Votes                 False
Revenue (Millions)    False
Metascore             False
Action                False
Adventure             False
Aniimation            False
Biography             False
Comedy                False
Crime                 False
Drama                 False
Family                False
Fantasy               False
History               False
Horror                False
Music                 False
Musical               False
Mystery               False
Romance               False
Sci-Fi                False
Sport                 False
Thriller              False
War                   False
Western               False
Success               False
dtype: bool
```

In [11]:
```python
df.duplicated()
```

Out[11]:
```
0      False
1      False
2      False
3      False
4      False
       ...  
833    False
834    False
835    False
836    False
837    False
Length: 838, dtype: bool
```

In [12]:
```python
df.duplicated().sum()
```

Out[12]:
```
0
```

In [13]:
```python
df.describe()
```

Out[13]:

	Rank	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore	Action	Adventure	Aniimation	...	Music	Musical	Mystery	Romance	Sci-Fi	Sport	Thriller
count	838.000000	838.00000	838.000000	838.000000	8.380000e+02	838.000000	838.000000	838.000000	838.000000	838.000000	...	838.000000	838.000000	838.000000	838.000000	838.000000	838.000000	838.000000
mean	485.247017	2012.50716	114.638425	6.814320	1.932303e+05	84.564558	59.575179	0.330549	0.291169	0.053699	...	0.023866	0.005967	0.102625	0.143198	0.127685	0.017900	0.176611
std	286.572065	3.17236	18.470922	0.877754	1.930990e+05	104.520227	16.952416	0.470692	0.454573	0.225558	...	0.152724	0.077059	0.303650	0.350484	0.333938	0.132666	0.381567
min	1.000000	2006.00000	66.000000	1.900000	1.780000e+02	0.000000	11.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	238.250000	2010.00000	101.000000	6.300000	6.127650e+04	13.967500	47.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	475.500000	2013.00000	112.000000	6.900000	1.368795e+05	48.150000	60.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	729.750000	2015.00000	124.000000	7.500000	2.710830e+05	116.800000	72.000000	1.000000	1.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1000.000000	2016.00000	187.000000	9.000000	1.791916e+06	936.630000	100.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 28 columns

In [14]:
```python
df.skew()
```

```
C:\Users\Sutharsahana\AppData\Local\Temp\ipykernel_16228\1665899112.py:1: FutureWarning: The default value of numeric_only in DataFrame.skew is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.skew()
```

Out[14]:
```
Rank                   0.061969
Year                  -0.586622
Runtime (Minutes)      0.793446
Rating                -0.677057
Votes                  2.466783
Revenue (Millions)     2.560612
Metascore             -0.124203
Action                 0.721731
Adventure              0.920997
Aniimation             3.966772
Biography              3.103034
Comedy                 0.883154
Crime                  1.959976
Drama                  0.000000
Family                 3.817229
Fantasy                2.500880
History                5.537190
Horror                 2.602357
Music                  6.250139
Musical               12.852905
Mystery                2.623581
Romance                2.040924
Sci-Fi                 2.235180
Sport                  7.285247
Thriller               1.699113
War                    9.005682
Western               14.396056
Success                1.688377
dtype: float64
```

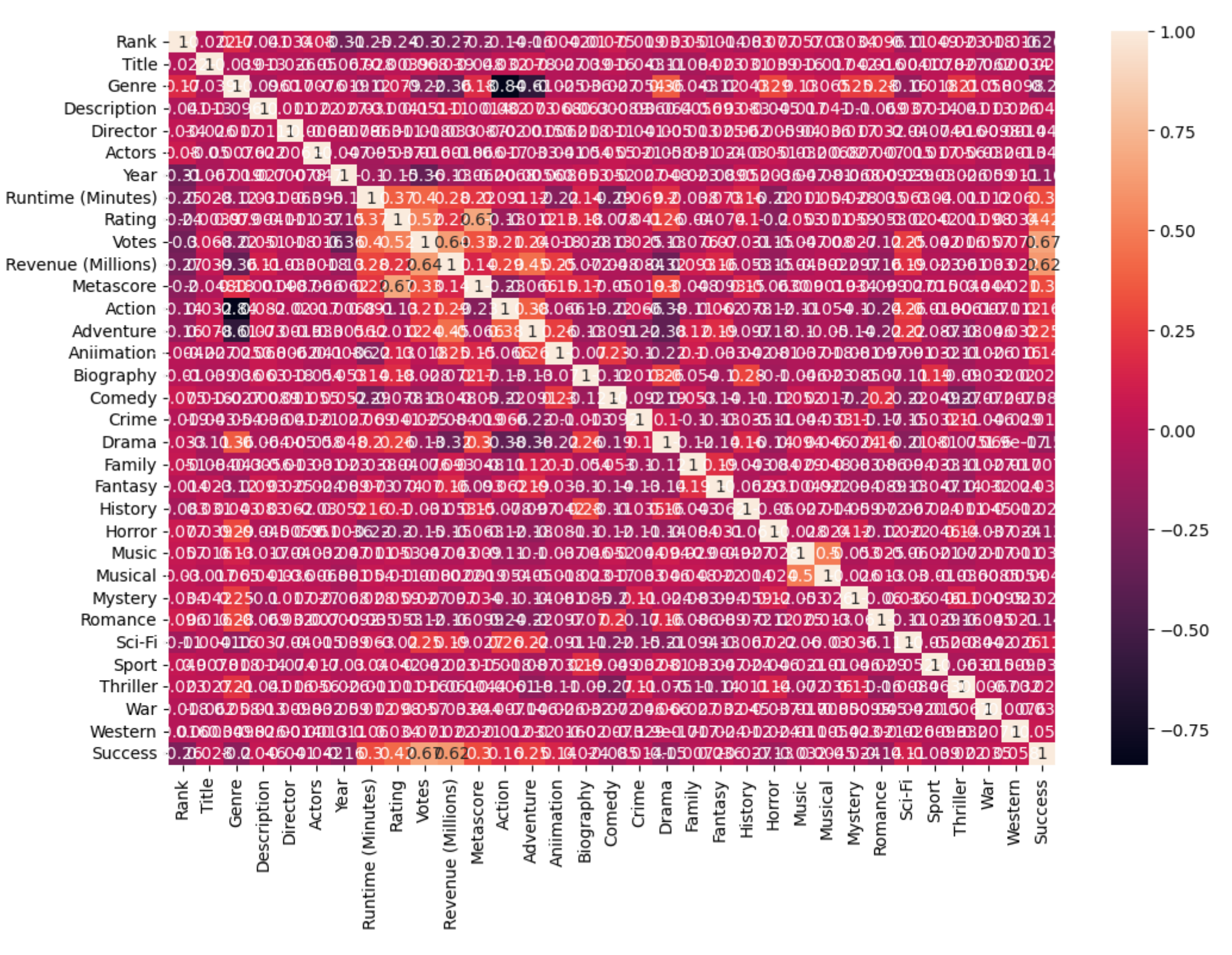
In [15]:
```python
from sklearn.preprocessing import LabelEncoder
Numerics=LabelEncoder()

#data['label']=Numerics.fit_transform(data['label'])
df['Title']=Numerics.fit_transform(df['Title'])
df['Genre']=Numerics.fit_transform(df['Genre'])
df['Director']=Numerics.fit_transform(df['Director'])
df['Actors']=Numerics.fit_transform(df['Actors'])
df['Description']=Numerics.fit_transform(df['Description'])
```

In [ ]:
```python
import seaborn as sns  #seaborn
sns.pairplot(df,hue="Revenue (Millions)",diag_kind="hist")
```

In [17]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(12,8))
sns.heatmap(df.corr(),annot=True)
```

Out[17]:
```
<Axes: >
```



In [18]:
```python
df['Genre'].value_counts()
```

Out[18]:
```
11     50
111    30
139    29
165    27
101    26
       ..
67      1
26      1
126     1
61      1
114     1
Name: Genre, Length: 189, dtype: int64
```

In [19]:
```python
df['Director'].value_counts()
```

Out[19]:
```
417    8
391    6
126    6
341    6
36     5
      ..
297    1
80     1
397    1
304    1
372    1
Name: Director, Length: 524, dtype: int64
```

In [20]:
```python
df['Actors'].value_counts()
```

Out[20]:
```
381    2
197    2
727    2
292    2
156    1
      ..
140    1
136    1
614    1
795    1
458    1
Name: Actors, Length: 834, dtype: int64
```

In [21]:
```python
df = df.fillna(df.median())
```

In [22]:
```python
df.columns
```

Out[22]:
```
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'Action', 'Adventure', 'Aniimation', 'Biography', 'Comedy',
       'Crime', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Music',
       'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'War',
       'Western', 'Success'],
      dtype='object')
```

In [23]:
```python
x = df[['Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'Action', 'Adventure', 'Aniimation', 'Biography', 'Comedy',
       'Crime', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Music',
       'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'War',
       'Western']]
y = df['Success']
```

In [24]:
```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.1,stratify=y)
```

In [25]:
```python
from sklearn.linear_model import LogisticRegression
log = LogisticRegression()
log.fit(x_train,y_train)
```

Out[25]:
```
▾ LogisticRegression
LogisticRegression()
```

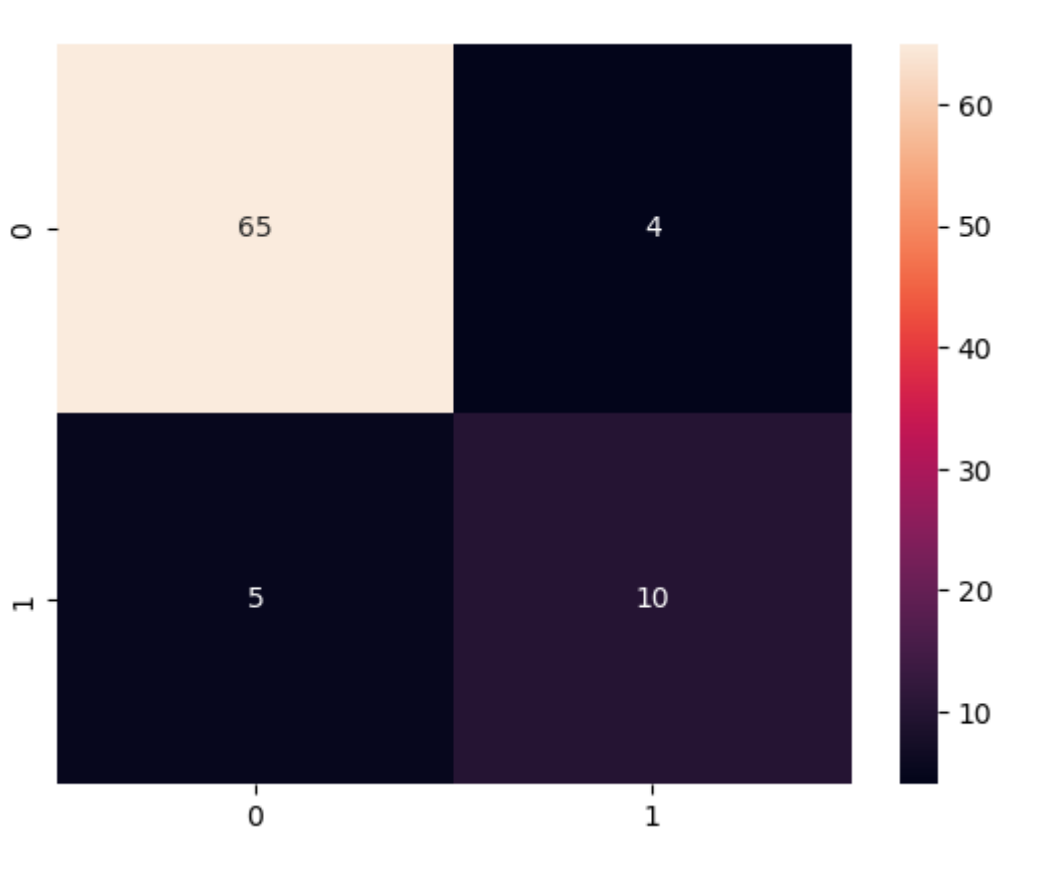
In [26]:
```python
log.score(x_test,y_test)
```

Out[26]:
```
0.8928571428571429
```

In [27]:
```python
from sklearn.metrics import confusion_matrix
clf = confusion_matrix(y_test,log.predict(x_test))
```

In [28]:
```python
sns.heatmap(clf,annot=True)
```

Out[28]:
```
<Axes: >
```



In [29]:
```python
#normalising all columns
x_train_opt = x_train.copy()
x_test_opt = x_test.copy()
```

In [30]:
```python
from sklearn.preprocessing import StandardScaler
x_train_opt = StandardScaler().fit_transform(x_train_opt)
x_test_opt = StandardScaler().fit_transform(x_test_opt)
```

In [31]:
```python
log.fit(x_train_opt,y_train)
```

Out[31]:
```
▾ LogisticRegression
LogisticRegression()
```

In [32]:
```python
log.score(x_test_opt,y_test)
```

Out[32]:
```
0.8809523809523809
```

--DECISION TREE CLASSIFIER--

In [33]:
```python
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier()
tree.fit(x_train,y_train)
tree.score(x_test,y_test)
```

Out[33]:
```
1.0
```

In [34]:
```python
tree.score(x_train,y_train)
```

Out[34]:
```
1.0
```

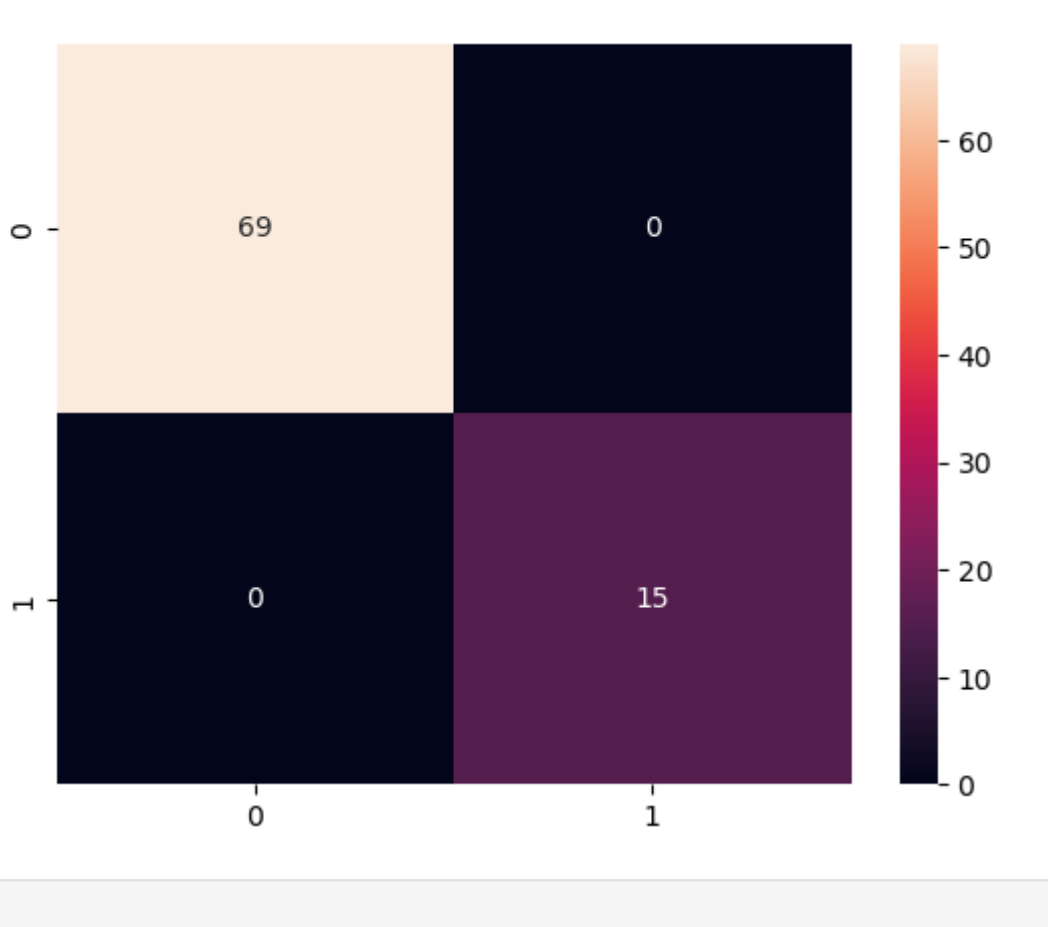
In [35]:
```python
from sklearn.metrics import confusion_matrix
clf = confusion_matrix(y_test,tree.predict(x_test))
```

In [36]:
```python
clf
```

Out[36]:
```
array([[69,  0],
       [ 0, 15]], dtype=int64)
```

In [37]:
```python
sns.heatmap(clf,annot=True)
```

Out[37]:
```
<Axes: >
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: