Bharat Intern

Name: Sham johari

Task 2

Success of an upcoming movie :

For this project, you need to predict the success of an upcoming movie so that whether or not a company should go for buying it based on ROI. To do this, you need to come up with a model and use the historical data of each.

In [27]:

```
# important libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
```

In [28]:

```
# import the dataset
df = pd.read_csv('movie_success_rate.csv')
```

In [29]:

```
df.head()
```

Out[29]:

	Rank	Title	Genre	Description	Director	Actors	
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...	2
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...	2
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...	2
3	4.0	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2
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...	2

5 rows × 33 columns

In [30]:

```
df.shape
```

Out[30]:

```
(839, 33)
```

In [31]:

```
df.columns
```

Out[31]:

```
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Ye
ar',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'Action', 'Adventure', 'Aniimation', 'Biography', 'Co
medy',
       'Crime', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Musi
c',
       'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'W
ar',
       'Western', 'Success'],
      dtype='object')
```

In [32]:

```
df['Genre'].value_counts()
```

Out[32]:

```
Action,Adventure,Sci-Fi    50
Comedy,Drama,Romance       30
Drama                      29
Drama,Romance              27
Comedy                     26
                           ..
Adventure,Drama,History     1
Action,Crime,Fantasy        1
Comedy,Mystery              1
Adventure,Comedy,Horror     1
Comedy,Family,Fantasy       1
Name: Genre, Length: 189, dtype: int64
```

In [33]:

```
df['Director'].value_counts()
```

Out[33]:

```
Ridley Scott          8
Paul W.S. Anderson    6
David Yates           6
Michael Bay           6
Antoine Fuqua         5
                     ..
Kyle Balda            1
Chris Renaud          1
Peter Billingsley     1
Lee Toland Krieger    1
Nima Nourizadeh       1
Name: Director, Length: 524, dtype: int64
```

In [34]:

```
df['Actors'].value_counts()
```

Out[34]:

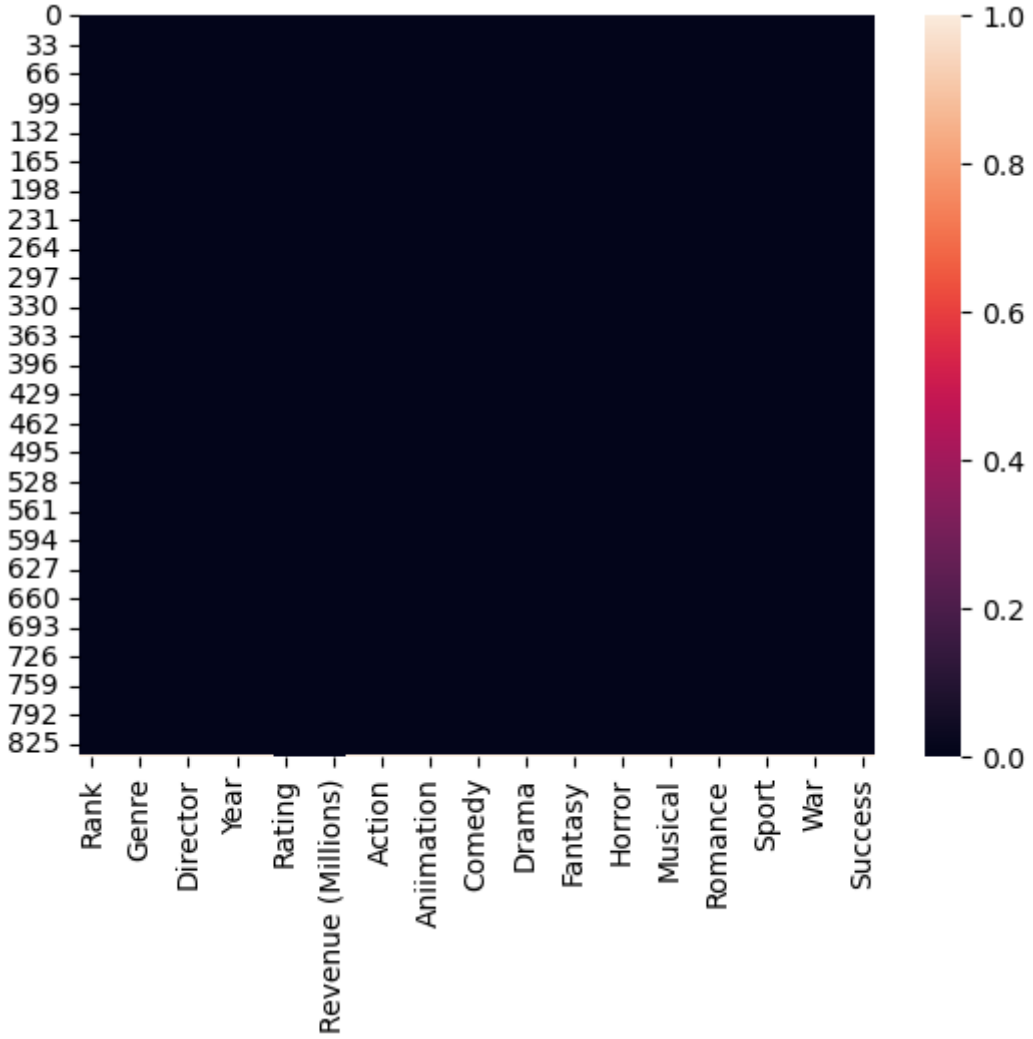
```
Jennifer Lawrence, Josh Hutcherson, Liam Hemsworth, Woody Harrelson    2
Daniel Radcliffe, Emma Watson, Rupert Grint, Michael Gambon            2
Shia LaBeouf, Megan Fox, Josh Duhamel, Tyrese Gibson                   2
Gerard Butler, Aaron Eckhart, Morgan Freeman,Angela Bassett            2
Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana                   1
                                                                      ..
Chris Evans, Jamie Bell, Tilda Swinton, Ed Harris                      1
Chloë Grace Moretz, Matthew Zuk, Gabriela Lopez,Bailey Anne Borders    1
Olivia DeJonge, Ed Oxenbould, Deanna Dunagan, Peter McRobbie           1
Vin Diesel, Paul Walker, Dwayne Johnson, Jordana Brewster              1
Kevin Spacey, Jennifer Garner, Robbie Amell,Cheryl Hines               1
Name: Actors, Length: 834, dtype: int64
```

In [35]:

```
sns.heatmap(df.isnull())
```

Out[35]:

```
<AxesSubplot:>
```



In [36]:

```
df = df.fillna(df.median())
```

```
C:\Users\91883\AppData\Local\Temp\ipykernel_22648\3493596106.py:1: Future
Warning: Dropping of nuisance columns in DataFrame reductions (with 'nume
ric_only=None') is deprecated; in a future version this will raise TypeEr
ror.  Select only valid columns before calling the reduction.
  df = df.fillna(df.median())
```

LOGISTIC REGRESSION

In [37]:

```
df.columns
```

Out[37]:

```
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Ye
ar',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'Action', 'Adventure', 'Aniimation', 'Biography', 'Co
medy',
       'Crime', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Musi
c',
       'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'W
ar',
       'Western', 'Success'],
      dtype='object')
```

In [38]:

```
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 [39]:

```
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.1,stratify=y)
```

In [40]:

```
x_train
```

Out[40]:

	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore	Action	Adventure	Aniimatic
86	2016.0	90.0	5.8	9247.0	0.15	62.0	0.0	0.0	0
802	2016.0	80.0	6.1	2417.0	0.15	69.0	0.0	1.0	0
401	2009.0	108.0	6.7	241709.0	163.95	48.0	0.0	0.0	0
73	2015.0	137.0	7.2	301249.0	350.03	67.0	1.0	0.0	0
720	2013.0	128.0	7.5	69659.0	95.00	62.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	
340	2014.0	86.0	7.6	84016.0	3.33	76.0	0.0	0.0	0
445	2013.0	91.0	7.8	622089.0	274.08	96.0	0.0	0.0	0
640	2009.0	129.0	8.2	144524.0	20.17	80.0	0.0	0.0	0
740	2009.0	85.0	2.7	59512.0	9.35	45.0	1.0	1.0	0
652	2014.0	114.0	7.3	151970.0	31.24	68.0	0.0	0.0	0

755 rows × 26 columns

In [41]:

```
log = LogisticRegression()
log.fit(x_train,y_train)
```

Out[41]:

```
LogisticRegression()
```

In [42]:

```
log.score(x_test,y_test)
```

Out[42]:

```
0.9047619047619048
```

In [43]:

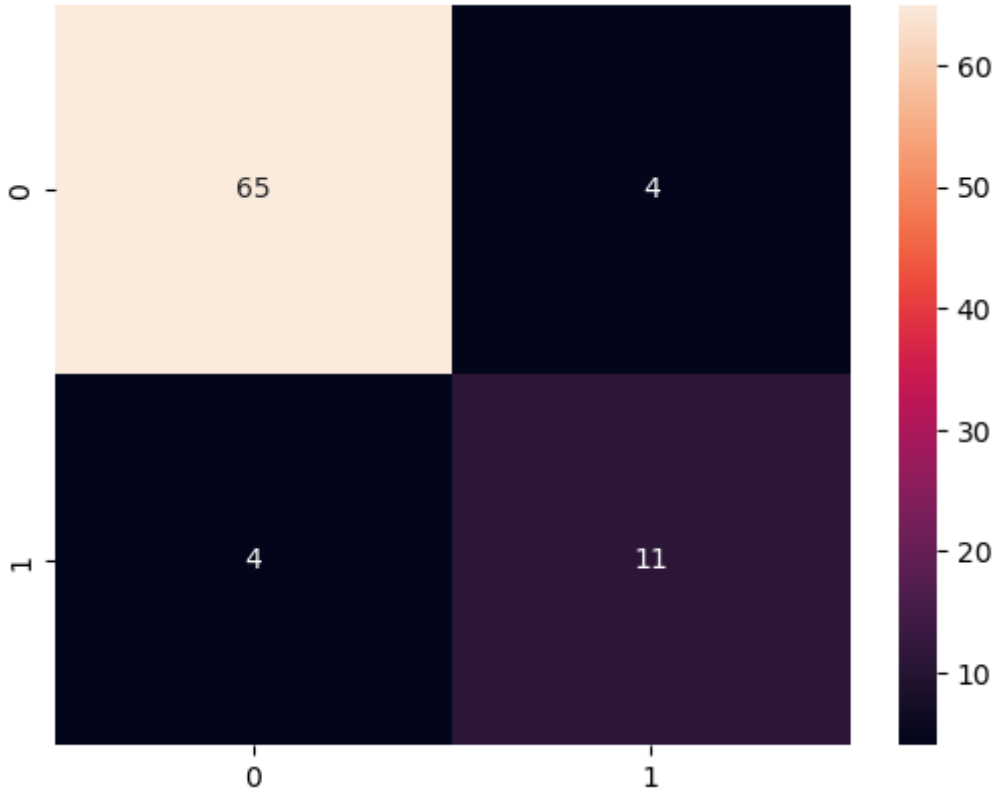
```
clf = confusion_matrix(y_test,log.predict(x_test))
```

In [44]:

```
sns.heatmap(clf,annot=True)
```

Out[44]:

<AxesSubplot:>



Thank you...

In [ ]:

```

```