In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries instal
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/d
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will lis

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that ge
# You can also write temporary files to /kaggle/temp/, but they won't be saved
```

In [3]:
```
df = pd.read_csv('movie_success_rate.csv')
```

In [4]:
```
df.shape
```

Out[4]:
```
(839, 33)
```

In [5]:
```
df.head()
```

Out[5]:

	Rank	Title	Genre	Description	Director	Actors	Yea
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
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
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
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
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

5 rows × 33 columns

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'Action', 'Adventure', 'Aniimation', 'Biography', 'Comed
y',
       'Crime', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Music',
       'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'War',
       'Western', 'Success'],
      dtype='object')
```

In [7]:
```
df['Genre'].value_counts()
```

Out[7]:
```
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 [8]:
```
df['Director'].value_counts()
```

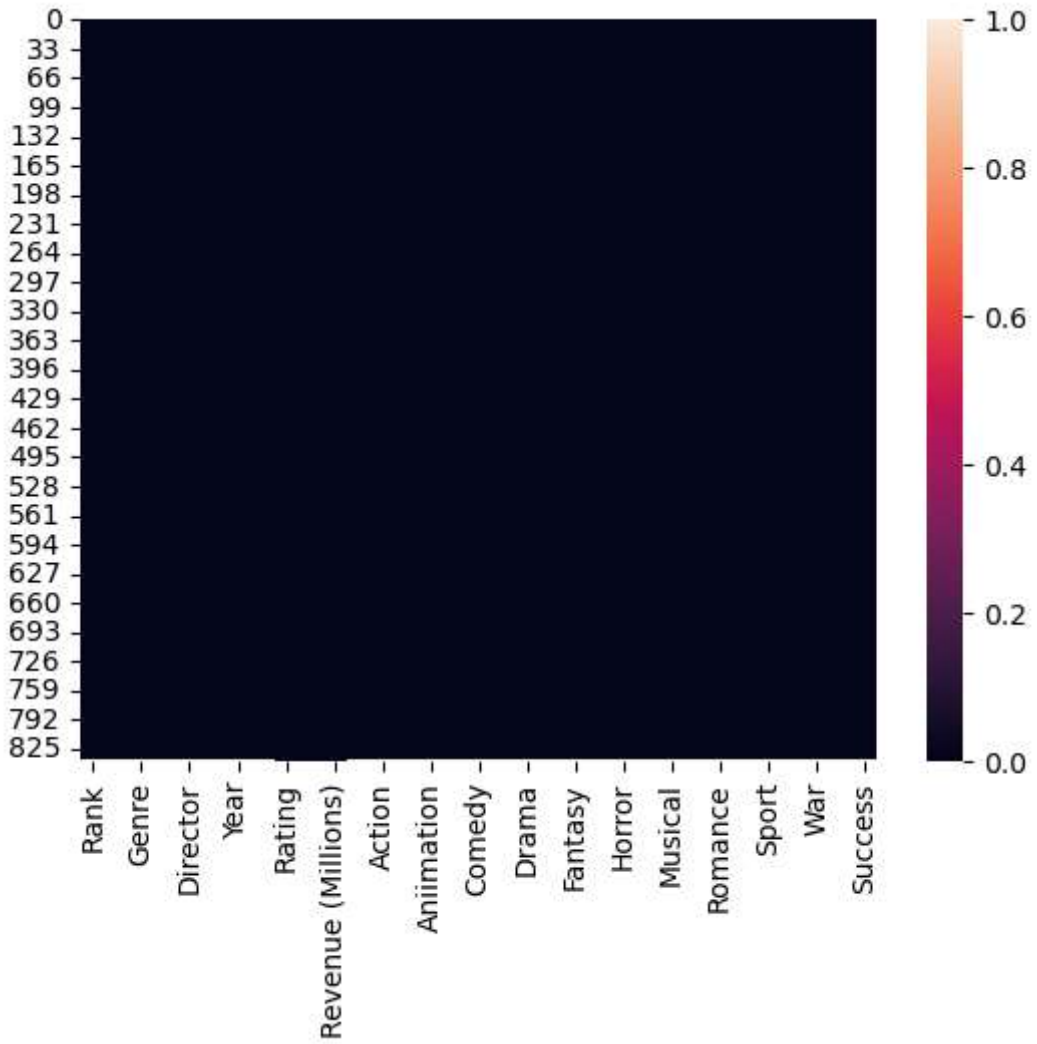
Out[8]:
```
Ridley Scott          8
Paul W.S. Anderson    6
David Yates           6
Michael Bay           6
Antoine Fuqua         5
                     ..
Kyle Balda            1
Chris Renaud          1
Peter Billingsley     1
Lee Toland Krieger    1
Nima Nourizadeh       1
Name: Director, Length: 524,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

In [9]:
```
df['Actors'].value_counts()
```

Out[9]:
```
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 [10]:
```
import seaborn as sns
sns.heatmap(df.isnull())
```

Out[10]: <Axes: >



In [11]:
```
df = df.fillna(df.median())
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_14124\3493596106.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.median is deprecated. In a
future version, it will default to False. In addition, specifying 'numeric_on
ly=None' is deprecated. Select only valid columns or specify the value of num
eric_only to silence this warning.
  df = df.fillna(df.median())
```

LOGICAL REGRESSION

In [12]:
```
df.columns
```

Out[12]:
```
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'Action', 'Adventure', 'Aniimation', 'Biography', 'Comed
y',
       'Crime', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Music',
       'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Sport', 'Thriller', 'War',
       'Western', 'Success'],
      dtype='object')
```

In [13]:
```
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 [14]:
```
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 [15]:
```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression()
log.fit(x_train,y_train)
```

Out[15]:
```
LogisticRegression
LogisticRegression()
```

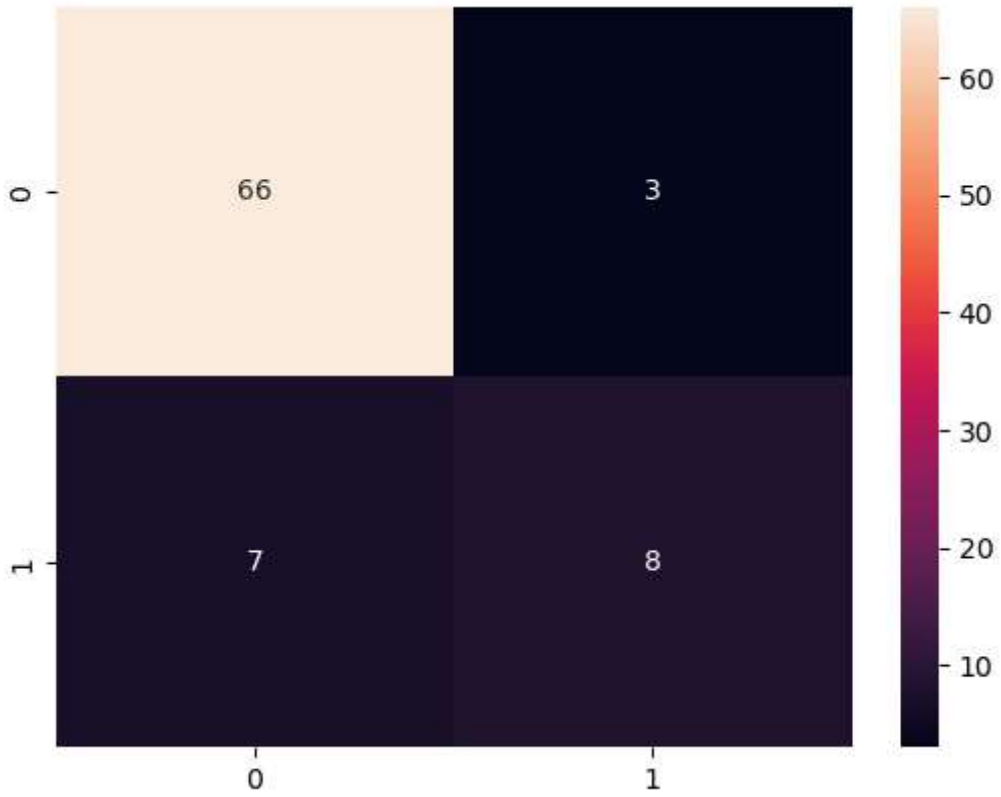
In [16]:
```
log.score(x_test,y_test)
```

Out[16]:
```
0.8809523809523809
```

In [17]:
```
from sklearn.metrics import confusion_matrix
clf = confusion_matrix(y_test,log.predict(x_test))
```

In [18]:
```
sns.heatmap(clf,annot=True)
```

Out[18]: <Axes: >



In [19]:
```
#normalising all columns
x_train_opt = x_train.copy()
x_test_opt = x_test.copy()
```

In [20]:
```
from sklearn.preprocessing import StandardScaler
x_train_opt = StandardScaler().fit_transform(x_train_opt)
x_test_opt = StandardScaler().fit_transform(x_test_opt)
```

In [21]:
```
#fitting again in Logistic Regression
```

In [22]:
```
log.fit(x_train_opt,y_train)
```

Out[22]:
```
LogisticRegression
LogisticRegression()
```

In [23]:
```
log.score(x_test_opt,y_test)
```

Out[23]: 0.8452380952380952

KNN

In [24]:
```
from sklearn.neighbors import KNeighborsClassifier
kn = KNeighborsClassifier(n_neighbors=40)
kn.fit(x_train,y_train)
```

Out[24]:
```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=40)
```

In [25]:
```
kn.score(x_test,y_test)
```

Out[25]:
```
0.8571428571428571
```

DECISION TREE

In [26]:
```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier()
tree.fit(x_train,y_train)
tree.score(x_test,y_test)
```

Out[26]:
```
1.0
```

In [27]:
```
tree.score(x_train,y_train)
```

Out[27]:
```
1.0
```

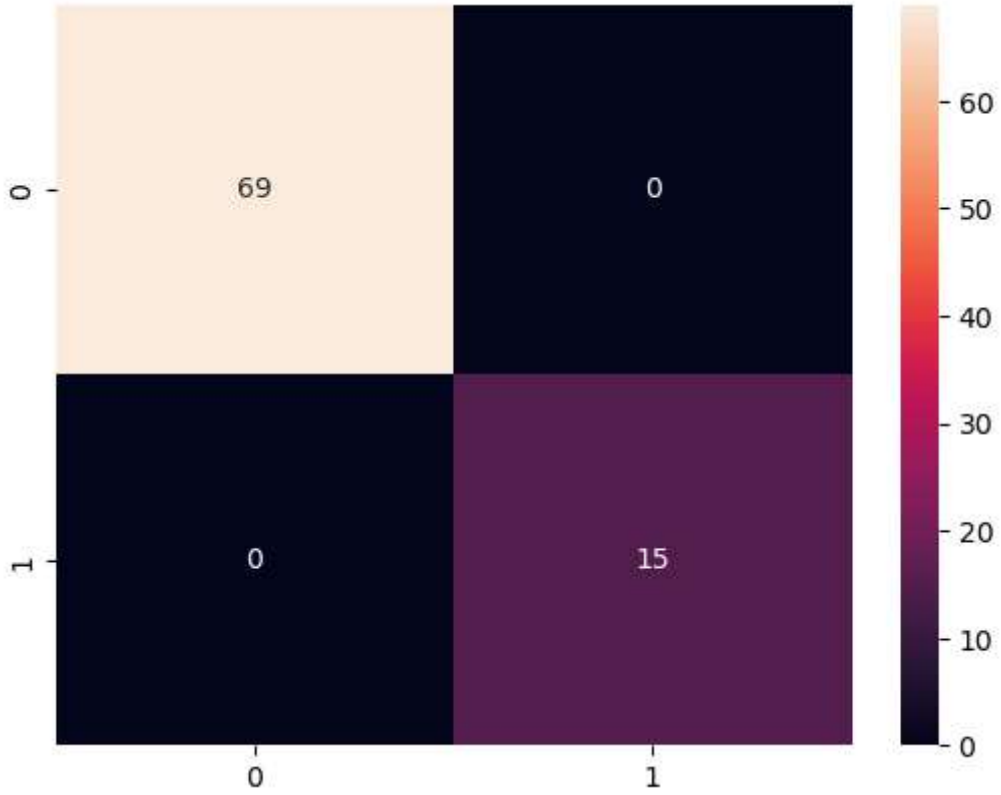
In [28]:
```
from sklearn.metrics import confusion_matrix
clf = confusion_matrix(y_test,tree.predict(x_test))
```

In [29]:
```
clf
```

Out[29]:
```
array([[69,  0],
       [ 0, 15]], dtype=int64)
```

In [30]:
```
sns.heatmap(clf,annot=True)
```

Out[30]: <Axes: >



In [ ]:
```

```