In [1]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,classification_report,confus
```

In [2]:
```
df=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\movies.csv")
```

In [3]:
```
df.shape
```

Out[3]:
```
(9742, 3)
```

In [4]:
```
df=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\ratings.csv")
```

In [5]:
```
df.shape
```

Out[5]:
```
(100836, 4)
```

In [8]:
```
uq_you_ids=df['userId'].nunique()

print("unique userId:", uq_you_ids)
```

```
unique userId: 610
```

In [9]:
```
movi_rate_cnt=df.groupby('movieId')['rating'].count()

max_rat_movi_id=movi_rate_cnt.idxmax()

df=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\movies.csv")

max_rat_movi=df[df['movieId']==max_rat_movi_id]['title'].values[0]

print("max no of rating:", max_rat_movi)
```

```
max no of rating: Forrest Gump (1994)
```

In [11]:
```
df1=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\movies.csv")

df=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\tags.csv")

mx_movi_id=df1[(df1['title']=='Matrix,The(1999)')]['movieId'].values[0]

mx_tgs=df[df['movieId']==mx_movi_id]['tag']

print("tgs sbmtd by urs to 'Matrix,The(1999)':")
print(mx_tgs.unique())
```

```
tgs sbmtd by urs to 'Matrix, The (1999)':
['martial arts' 'sci-fi' 'alternate universe' 'philosophy'
 'post apocalyptic']
```

In [18]:
```
df=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\ratings.csv")

df1=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\movies.csv")

trm_movi_id=df1[(df1['title']=='Terminator 2: Judgment Day (1991)')]['m

trm_rtgs=df[df['movieId']==trm_movi_id]

avg_rtg=trm_rtgs['rating'].mean()

print("avg usr rtg for 'Terminator 2: Judgment Day (1991)':", avg_rtg)
```

```
avg usr rtg for 'Terminator 2: Judgment Day (1991)': 3.970982142857143
```

In [24]:
```
df=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\ratings.csv")

df1=pd.read_csv(r"C:\Users\viraj\Desktop\movie_data\links.csv")

avg_rtgs=df.groupby('movieId')['rating'].agg(['count', 'mean']).reset_i

plr_movis= avg_rtgs[avg_rtgs['count']>50]

df3=pd.merge(plr_movis,df1,on='movieId',how='inner')

high_imdb_movi=df3.nlargest(1,'imdbId')

print("high imdb rtgs:",high_imdb_movi['movieId'].values[0])
```

```
high imdb rtgs: 109374
```

In [ ]:

In [ ]:

In [ ]: