This file only contains the screenshots of the navigator project.I uploaded code seperately in this repository only.

In [2]:
```
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import CountVectorizer

pd.set_option('display.max_columns', 100)
df = pd.read_csv('https://query.data.world/s/uikepcpffyo2nhig52xxeevdialfl7')
df.head()
```

Out[2]:

	Unnamed: 0	Title	Year	Rated	Released	Runtime	Genre	Director	Writer	Actors	Plot	Language	Country	Awards	
0	1	The Shawshank Redemption	1994	R	14 Oct 1994	142 min	Crime, Drama	Frank Darabont	Stephen King (short story "Rita Hayworth and S...	Tim Robbins, Morgan Freeman, Bob Gunton, Willi...	Two imprisoned men bond over a number of years...	English	USA	Nominated for 7 Oscars. Another 19 wins & 30 n...	http amaz
1	2	The Godfather	1972	R	24 Mar 1972	175 min	Crime, Drama	Francis Ford Coppola	Mario Puzo (screenplay), Francis Ford Coppola ...	Marlon Brando, Al Pacino, James Caan, Richard ...	The aging patriarch of an organized crime dyna...	English, Italian, Latin	USA	Won 3 Oscars. Another 23 wins & 27 nominations.	http amaz
2	3	The Godfather: Part II	1974	R	20 Dec 1974	202 min	Crime, Drama	Francis Ford Coppola	Francis Ford Coppola (screenplay), Mario Puzo ...	Al Pacino, Robert Duvall, Diane Keaton, Robert ...	The early life and career of Vito Corleone in ...	English, Italian, Spanish, Latin, Sicilian	USA	Won 6 Oscars. Another 10 wins & 20 nominations.	http amaz

(next page)

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

Out[3]: (250, 38)

In [4]:
```
df = df[['Title','Genre','Director','Actors','Plot']]
df.head()
```

Out[4]:

	Title	Genre	Director	Actors	Plot
0	The Shawshank Redemption	Crime, Drama	Frank Darabont	Tim Robbins, Morgan Freeman, Bob Gunton, Willi...	Two imprisoned men bond over a number of years...
1	The Godfather	Crime, Drama	Francis Ford Coppola	Marlon Brando, Al Pacino, James Caan, Richard ...	The aging patriarch of an organized crime dyna...
2	The Godfather: Part II	Crime, Drama	Francis Ford Coppola	Al Pacino, Robert Duvall, Diane Keaton, Robert...	The early life and career of Vito Corleone in ...
3	The Dark Knight	Action, Crime, Drama	Christopher Nolan	Christian Bale, Heath Ledger, Aaron Eckhart, M...	When the menace known as the Joker emerges fro...
4	12 Angry Men	Crime, Drama	Sidney Lumet	Martin Balsam, John Fiedler, Lee J. Cobb, E.G....	A jury holdout attempts to prevent a miscarria...

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

Out[5]: (250, 5)

In [6]:
```
# discarding the commas
df['Actors'] = df['Actors'].map(lambda x: x.split(',')[:3])

# putting the genres in a list of words
df['Genre'] = df['Genre'].map(lambda x: x.lower().split(','))

df['Director'] = df['Director'].map(lambda x: x.split(' '))
```


In [6]:
```
# discarding the commas
df['Actors'] = df['Actors'].map(lambda x: x.split(',')[:3])

# putting the genres in a list of words
df['Genre'] = df['Genre'].map(lambda x: x.lower().split(','))

df['Director'] = df['Director'].map(lambda x: x.split(' '))
```

```
# merging together first and last name for each actor and director
for index, row in df.iterrows():
    row['Actors'] = [x.lower().replace(' ','') for x in row['Actors']]
    row['Director'] = ''.join(row['Director']).lower()
```

In [7]:
```
df.set_index('Title', inplace = True)
df.head()
```

Out[7]:

Title	Genre	Director	Actors	Plot
The Shawshank Redemption	[crime, drama]	frankdarabont	[timrobbins, morganfreeman, bobgunton]	Two imprisoned men bond over a number of years...
The Godfather	[crime, drama]	francisfordcoppola	[marlonbrando, alpacino, jamescaan]	The aging patriarch of an organized crime dyna...
The Godfather: Part II	[crime, drama]	francisfordcoppola	[alpacino, robertduvall, dianekeaton]	The early life and career of Vito Corleone in ...
The Dark Knight	[action, crime, drama]	christophernolan	[christianbale, heathledger, aaroneckhart]	When the menace known as the Joker emerges fro...
12 Angry Men	[crime, drama]	sidneylumet	[martinbalsam, johnfiedler, leej.cobb]	A jury holdout attempts to prevent a miscarria...

In [8]:
```
df['bag_of_words'] = ''
columns = df.columns
for index, row in df.iterrows():
    words = ''
    for col in columns:
        if col != 'Director':
            words = words + ' '.join(row[col])+ ' '
        else:
            words = words + row[col]+ ' '
    row['bag_of_words'] = words

df.drop(columns = [col for col in df.columns if col!= 'bag_of_words'], inplace = True)
```

In [9]:
```
df.head()
```

Out[9]:

Title	bag_of_words
The Shawshank Redemption	crime drama frankdarabont timrobbins morganfr...
The Godfather	crime drama francisfordcoppola marlonbrando a...
The Godfather: Part II	crime drama francisfordcoppola alpacino rober...
The Dark Knight	action crime drama christophernolan christia...
12 Angry Men	crime drama sidneylumet martinbalsam johnfied...

(next page)

In [10]:
```
# instantiating and generating the count matrix
count = CountVectorizer()
count_matrix = count.fit_transform(df['bag_of_words'])


indices = pd.Series(df.index)
indices[:5]
```

Out[10]:
```
0    The Shawshank Redemption
1                The Godfather
2      The Godfather: Part II
3             The Dark Knight
4                12 Angry Men
Name: Title, dtype: object
```

In [11]:
```
cosine_sim = cosine_similarity(count_matrix, count_matrix)
cosine_sim
```

Out[11]:
```
array([[1.        , 0.33333333, 0.33333333, ..., 0.15430335, 0.15430335,
        0.16666667],
       [0.33333333, 1.        , 0.66666667, ..., 0.15430335, 0.15430335,
        0.16666667],
       [0.33333333, 0.66666667, 1.        , ..., 0.15430335, 0.15430335,
        0.16666667],
       ...,
       [0.15430335, 0.15430335, 0.15430335, ..., 1.        , 0.14285714,
        0.15430335],
       [0.15430335, 0.15430335, 0.15430335, ..., 0.14285714, 1.        ,
        0.15430335],
       [0.16666667, 0.16666667, 0.16666667, ..., 0.15430335, 0.15430335,
        1.        ]])
```

(next page)

In [12]:
```
def recommendations(title, cosine_sim = cosine_sim):

    recommended_movies = []

    idx = indices[indices == title].index[0]

    score_series = pd.Series(cosine_sim[idx]).sort_values(ascending = False)

    top_10_indexes = list(score_series.iloc[1:11].index)

    for i in top_10_indexes:
        recommended_movies.append(list(df.index)[i])

    return recommended_movies
```

In [16]:
```
recommendations('The Godfather: Part II')
```

Out[16]:
```
['The Godfather',
 'Apocalypse Now',
 'Scarface',
 'Heat',
 'The Shawshank Redemption',
 'Cool Hand Luke',
 'Casino',
 'Taxi Driver',
 'To Kill a Mockingbird',
 'Once Upon a Time in America']
```