

Pranjal Patil 2019140048

Jeenal Mehta 2019140039

Prachi Mishra 2019140040

Code:

def predict():

start_time1 = time.time()

def createCharacteristicMatrix(filename):

data_frame = pd.read_csv(filename, sep=":", usecols = [0, 1, 2], names = ['userID', 'movieID', 'rating'], engine = 'python')

data_mat = np.array(data_frame.pivot(index = 'movieID', columns = 'userID', values = 'rating'))

data_mat_rev = np.nan_to_num(data_mat)

return data_mat_rev

print("\n\n Reading the data into the characteristic matrix... \n\n")

X = createCharacteristicMatrix('db/bolratings.dat')

n_movies, n_users = X.shape[0], X.shape[1]

end_time1 = time.time() - start_time1

print ("\n\n Time taken to generate the characteristic matrix consisting of", X.shape[0], "movies (in rows) and", X.shape[1], "users (in columns) =", round(end_time1, 2), "seconds\n\n")

print ("\n\n Defining the utility function ... \n\n")

define the utility function

def maxrate(A):

if len(A) > 0:

X_j = X[list(A)]

maxrate = float(np.sum(np.amax(X_j, axis = 0)))/float(n_users)

else:

```

        maxrate = 0

    return maxrate

K = [1,2,4]

print ("\n\n Running the 'Lazy Greedy' Submodular Maximization Algorithm ... \n\n")

start_time4 = time.time()

lazy_greedy_objective_value_list = []

lazy_recommended_movies = []

time_list_lazy = []

for k in K:

    A_lazy_greedy = set([])

    inter_start_lazy = time.time()

    for i in range(k):

        if i == 0:

            marginal_values_list = [maxrate(A_lazy_greedy.union(set([e]))) -
maxrate(A_lazy_greedy) for e in range(n_movies)]

            e_opt = np.argmax(marginal_values_list)

            A_lazy_greedy = A_lazy_greedy.union(set([e_opt]))

            marginal_values_list_sorted = sorted(marginal_values_list)[::-1][1:]

            movie_index_sorted = list(np.argsort(marginal_values_list))[::-1][1:]

        else:

            while (maxrate(A_lazy_greedy.union(set([movie_index_sorted[0]]))) -
maxrate(A_lazy_greedy)) < marginal_values_list_sorted[1]:

                marginal_values_list[movie_index_sorted[0]] =
maxrate(A_lazy_greedy.union(set([movie_index_sorted[0]]))) - maxrate(A_lazy_greedy)

            marginal_values_list_sorted = sorted(marginal_values_list)[::-1]

```

```

        movie_index_sorted = list(np.argsort(marginal_values_list))[:-1]

        A_lazy_greedy = A_lazy_greedy.union(set([movie_index_sorted[0]]))
        marginal_values_list_sorted = sorted(marginal_values_list)[::-1][1:]
        movie_index_sorted = list(np.argsort(marginal_values_list))[:-1][1:]

    inter_end_lazy = time.time() - inter_start_lazy
    lazy_recommended_movies=list(A_lazy_greedy)
    lazy_greedy_objective_value_list.append(maxrate(A_lazy_greedy))

    time_list_lazy.append(round(inter_end_lazy, 5))
    end_time4 = time.time() - start_time4

    print ("\n\n Time taken to implement the 'Lazy Greedy' Submodular Maximization
Algorithm =", round(end_time4, 5), "seconds\n\n")

    end_time = time.time() - start_time1
    print ("\n\n Time taken by program to run =", round(end_time, 5), "seconds\n\n")
    print(lazy_recommended_movies)
    movies_list=[]
    with open("db/bolmovies.dat", "r") as scan:
        for line in scan:
            iin=line.index(":")
            if(int(line[0:iin]) in lazy_recommended_movies):
                #lin=line.index("(")
                movies_list.append(line[iin+2:-1])
    filename=[]
    for x in movies_list:
        t=[]
        t.append('assets/images/movies/' + x + '.jpg')
        t.append(x)
        filename.append(t)

```

```
print(filename)
```

RESULT SCREENSHOTS

```
[[3. 0. 0. 3. 0. 0. 0. 0. 5. 0.]  
[0. 0. 0. 0. 2. 0. 0. 0. 4. 0.]  
[0. 0. 1. 0. 0. 4. 0. 2. 0. 2.]  
[0. 0. 0. 5. 3. 0. 0. 0. 0. 0.]  
[5. 0. 0. 0. 1. 0. 0. 4. 0. 0.]  
[0. 1. 0. 0. 0. 5. 0. 3. 3. 0.]  
[4. 0. 2. 0. 0. 0. 3. 0. 0. 1.]  
[0. 2. 0. 4. 0. 0. 2. 0. 0. 0.]  
[0. 0. 5. 0. 0. 1. 0. 0. 0. 0.]  
[0. 4. 0. 0. 0. 0. 5. 0. 0. 5.]]
```

Time taken to generate the characteristic matrix consisting of 10 movies (in rows) and 10 users (in columns) = 0.03 seconds

Defining the utility function ...

Running the 'Lazy Greedy' Submodular Maximization Algorithm ...

Time taken to implement the 'Lazy Greedy' Submodular Maximization Algorithm = 0.00306 seconds

Time taken by program to run = 0.03948 seconds

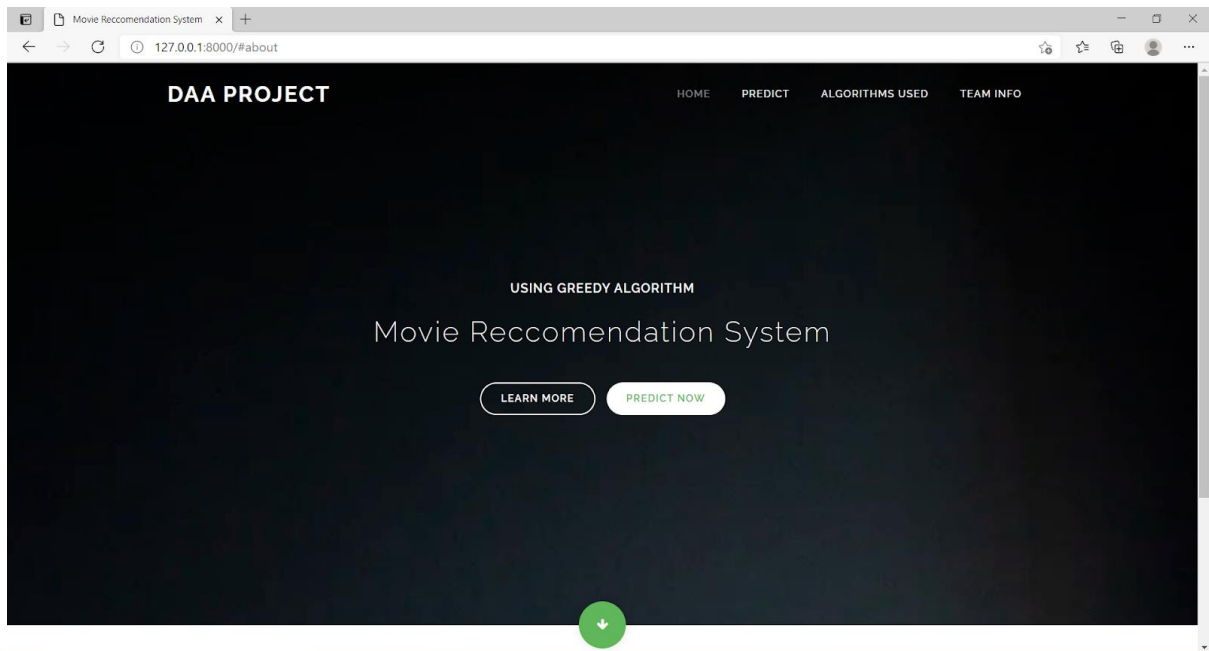
Time taken to implement the 'Lazy Greedy' Submodular Maximization Algorithm = 0.00306 seconds

Time taken by program to run = 0.03948 seconds

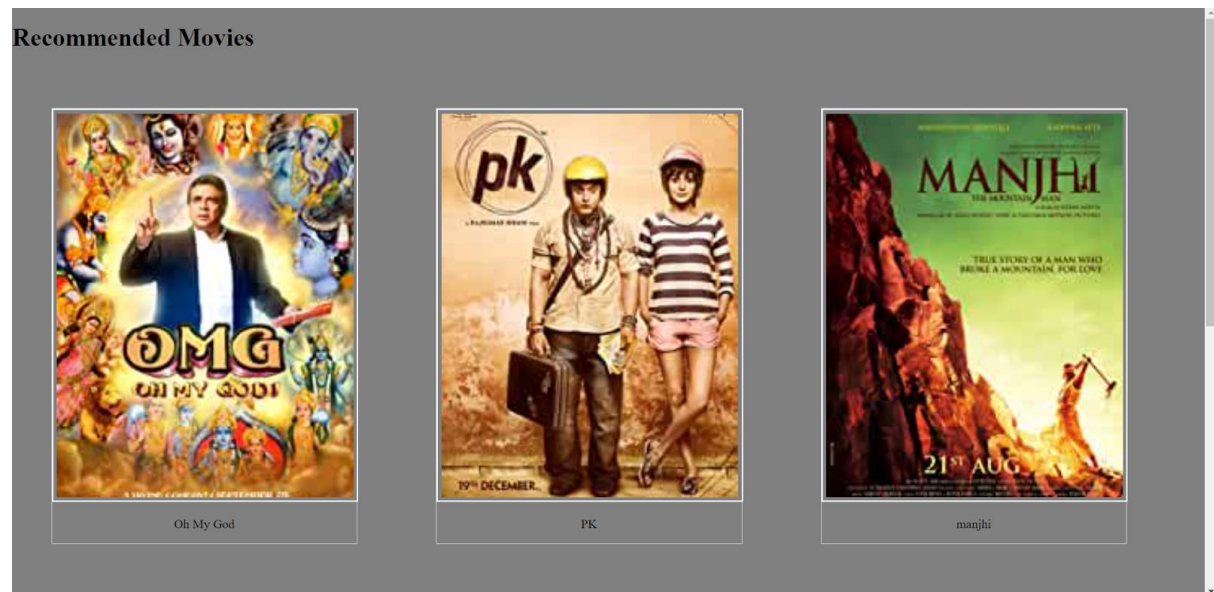
```
[9, 3, 4, 5]  
[['assets/images/movies/Oh My God.jpg', 'Oh My God'], ['assets/images/movies/PK.jpg', 'PK'], ['assets/images/movies/manjhi.jpg', 'manjhi'], ['assets/images/movies/Naaam Shabana.jpg', 'Naam Shabana']]  
127.0.0.1 - - [09/Apr/2021 10:57:27] "GET /predict/ HTTP/1.1" 200 -  
█
```

Frontend—

[Home page](#)



Result page



Part 2:

On a larger dataset

Reading the data into the characteristic matrix...

```
[[5. 0. 0. ... 0. 3. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Time taken to generate the characteristic matrix consisting of 3172 movies (in rows) and 500 users (in columns) = 0.49 seconds

Defining the utility function ...

Running the 'Lazy Greedy' Submodular Maximization Algorithm ...

Time taken to implement the 'Lazy Greedy' Submodular Maximization Algorithm = 34.65754 seconds

Time taken by program to run = 35.14826 seconds

```
[2433, 515, 10, 2836, 922, 2205, 417, 929, 2724, 934, 936, 937, 682, 2865, 2227, 3123, 1591, 2487, 2233, 3004, 2877, 2367, 3008, 2240, 1858, 3137, 1860, 463, 719, 727, 988,
 2140, 225, 2401, 3171, 999, 2663, 2281, 1898, 2156, 2670, 1267, 2420, 2166, 1274, 126, 2047]
[['assets/images/movies/goldenEye.jpg', 'GoldenEye'], ['assets/images/movies/NeverEnding Story III, The.jpg', 'NeverEnding Story III, The'], ['assets/images/movies/Disclousure.jpg', 'Disclosure'], ['assets/images/movies/Barcelona.jpg', 'Barcelona'], ['assets/images/movies/guilty as sin.jpg', 'Guilty as sin'], ['assets/images/movies/Remains of the Day, The.jpg', 'Remains of the Day, The'], ['assets/images/movies/Tigrero A Film That Was Never Made.jpg', 'Tigrero A Film That Was Never Made'], ['assets/images/movies/Multiplicity.jpg', 'Multiplicity'], ['assets/images/movies/War Stories.jpg', 'War Stories'], ['assets/images/movies/Sunset Blvd..jpg', 'Sunset Blvd.'], ['assets/images/movies/Foreign correspondent.jpg', 'Foreign correspondent'], ['assets/images/movies/Father of the Bride.jpg', 'Father of the Bride'], ['assets/images/movies/Ninotchka.jpg', 'Ninotchka'], ['assets/images/movies/Love in the Afternoon.jpg', 'Love in the Afternoon'], ['assets/images/movies/Grace of My Heart.jpg', 'Grace of My Heart'], ['assets/images/movies/2 Days in the Valley.jpg', '2 Days in the Valley'], ['assets/images/movies/Manchurian Candidate, The.jpg', 'Manchurian Candidate, The'], ['assets/images/movies/Akira.jpg', 'Akira'], ['assets/images/movies/Spawn.jpg', 'Spawn'], ['assets/images/movies/Mr. Nice Guy.jpg', 'Mr. Nice Guy'], ['assets/images/movies/Character.jpg', 'Character'], ['assets/images/movies/Land Girls, The.jpg', 'Land Girls, The'], ['assets/images/movies/Gnome-Mobile, The.jpg', 'Gnome-Mobile, The'], ['assets/images/movies/Dark Crystal, The.jpg', 'Dark Crystal, The'], ['assets/images/movies/Best Man, The.jpg', 'Best Man, The'], ['assets/images/movies/Return to Paradise.jpg', 'Return to Paradise'], ['assets/images/movies/Mr. & Mrs. Smith.jpg', 'Mr. & Mrs. Smith'], ['assets/images/movies/Lodger, The.jpg', 'Lodger, The'], ['assets/images/movies/Digging to China.jpg', 'Digging to China'], ['assets/images/movies/My Bodyguard.jpg', 'My Bodyguard'], ['assets/images/movies/Monument Ave..jpg', 'Monument Ave.'], ['assets/images/movies/King Kong.jpg', 'King Kong'], ['assets/images/movies/Pale Rider.jpg', 'Pale Rider'], ['assets/images/movies/Karate Kid, The.jpg', 'Karate Kid, The'], ['assets/images/movies/Civil Action, A.jpg', 'Civil Action, A'], ['assets/images/movies/Blood, Guts, Bullets and Octane.jpg', 'Blood, Guts, Bullets and Octane'], ['assets/images/movies/It Came from Beneath the Sea.jpg', 'It Came from Beneath the Sea'], ['assets/images/movies/Run Silent, Run Deep.jpg', 'Run Silent, Run Deep'], ['assets/images/movies/Runaway Bride.jpg', 'Runaway Bride'], ['assets/images/movies/Outside Providence.jpg', 'Outside Providence'], ['assets/images/movies/Sugar Town.jpg', 'Sugar Town'], ['assets/images/movies/Tommy.jpg', 'Tommy'], ['assets/images/movies/Bachelor, The.jpg', 'Bachelor, The'], ['assets/images/movies/Last Night.jpg', 'Last Night'], ['assets/images/movies/Sea Wolves, The.jpg', 'Sea Wolves, The'], ['assets/images/movies/Room at the Top.jpg', 'Room at the Top']]
```

Frontend—

Result page

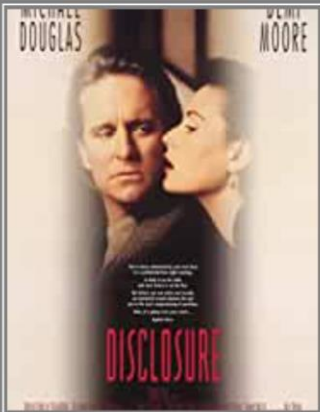
Recommended Movies



GoldenEye



NeverEnding Story III, The



Disclosure



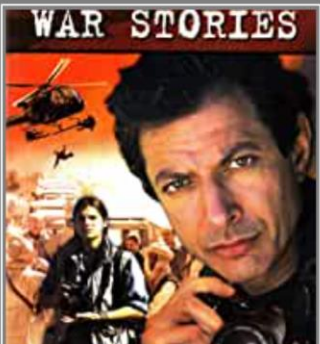
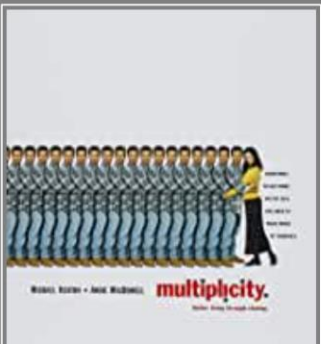
Barcelona



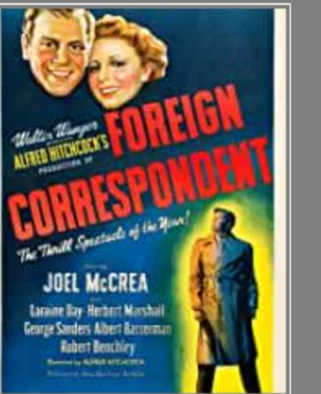
Guilty as Sin



Remains of the Day, The



Sunset Blvd.



Foreign Correspondent



Father of the Bride



