**In your report you should describe the following**:

1] **Your approach to use KNN as a recommender system?**

Ans: Using Nearest neighbors we predict the n items whose rating match with the user choice of movie.

2] **What is the maximum dataset that your recommender system can use?**

Ans: The built recommender system has the ability to run on large data set but it takes in a lot of time to run. Ie larger the data set more time it takes to run

So I recommend to use the data set that was provided with the zip file to

3] **What is the time complexity of your recommender system?**

Ans: O(n\*n)

4] **What is the performance of your recommender system?**

Ans: Recommended system works faster on a small dataset. On large dataset recommender system tends to slow down

5] **Is there a way to scale-up your recommender system to work with very large datasets?**

Ans: Yes, There is a way to scale things up. If we use Sklearn to implement KNN. Our Execution times reduces a lot.

**References:**

<https://www.udemy.com/course/recommender-systems/>

https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-part-2-k-nearest-neighbors-and-matrix-c04b3c2ef55c

<https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea>

**dataset:**

[**https://grouplens.org/datasets/movielens/latest/**](https://grouplens.org/datasets/movielens/latest/)

**Code Results:**

**Note**: Use Python Version 3.6

numpy

pandas 0.21

scipy

Using a wrong version leads to code crash

All dataset needs to be in the same folder. If in different folder please pass the path

Warning says: you try to take the natural logarithm of a negative number. This will result in a NaN.

Depending on the function used, you will be presented with a range of errors.

**Note: Large data set takes a long time to run. almost 15-20 min**

**Working:**

Our recommended system, Ask user to enter a Movie of his choice. Then it will look for user who also like that movie and based and output his fav movie. Then it will provide KNN recommendation based on the favorite movie.

Our system perform Matrix Factorization and then make the recommendation based on user given input choice.

**Input:**

Enter the number of neighbours you want to select: 10

Enter A Movie Name Of Your Choice: Toy Story (1995)

Enter The Number Of fav movies you wanna select: 3

Enter the number of recommendation you wanna see based on your choice: 5

**Output:**

/usr/local/bin/python3.6 /Users/shubhamrathod/Desktop/Programming\_assignment\_2/MyCode.py

Example 10: So 10 neighbours will be selected

Enter the number of neighbours you want to select: 10

example: Toy Story (1995) M\*A\*S\*H (a.k.a. MASH) (1970) Excalibur (1981) Indiana Jones and the Last Crusade (1989)

If given a wrong input -- > error will be popped

Enter A Movie Name Of Your Choice: Toy Story (1995)

Generating Matrix

movieId 1 2 3 4 5 6 7 8 \

userId

1 4.0 NaN 4.0 NaN NaN 4.0 NaN NaN

2 NaN NaN NaN NaN NaN NaN NaN NaN

3 NaN NaN NaN NaN NaN NaN NaN NaN

4 NaN NaN NaN NaN NaN NaN NaN NaN

5 4.0 NaN NaN NaN NaN NaN NaN NaN

movieId 9 10 ... 193565 193567 193571 193573 193579 \

userId ...

1 NaN NaN ... NaN NaN NaN NaN NaN

2 NaN NaN ... NaN NaN NaN NaN NaN

3 NaN NaN ... NaN NaN NaN NaN NaN

4 NaN NaN ... NaN NaN NaN NaN NaN

5 NaN NaN ... NaN NaN NaN NaN NaN

movieId 193581 193583 193585 193587 193609

userId

1 NaN NaN NaN NaN NaN

2 NaN NaN NaN NaN NaN

3 NaN NaN NaN NaN NaN

4 NaN NaN NaN NaN NaN

5 NaN NaN NaN NaN NaN

[5 rows x 9724 columns]

Example: enter 3 , and top 3 fav movie will be selected

Enter The Number Of fav movies you wanna select: 3

Example: enter 5 , to get top 5 recommendation

Enter the number of recommendation you wanna see based on your choice: 5

User Fav Movie for which he has giving highest rating:

['M\*A\*S\*H (a.k.a. MASH) (1970)', 'Excalibur (1981)', 'Indiana Jones and the Last Crusade (1989)']

Top Recommended movie with KNN and without matrix factor are:

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/scipy/spatial/distance.py:714: RuntimeWarning: invalid value encountered in double\_scalars

dist = 1.0 - uv / np.sqrt(uu \* vv)

['Shawshank Redemption, The (1994)', 'Godfather, The (1972)', "Ferris Bueller's Day Off (1986)", 'Sin City (2005)', 'Dark Knight, The (2008)']

Predicting Movies using Matrix Factorizaton

0

1

Movie Recommendation are:

['Othello (1995)', 'Dangerous Minds (1995)', 'Georgia (1995)', 'Braveheart (1995)', 'Taxi Driver (1976)']

Process finished with exit code 0