Movie Recommendation System

Hossein Zinaghaji
300098309

Soroush Salehi
300140057

Sahand Malek
300031550

Contents

1 Introduction

Recommender Systems are a subset of Information Filtering Systems. The purpose of recommender systems is to offer users the most relevant items (data, merchandise, entertainment and more). In recommender systems, data related to user behaviour (in purchasing goods, receiving data, watching entertainment and more) are analyzed and modelled. Then, the generated model is used to predict the most appropriate item for users. The most common use of recommender systems is in implementing business applications (in particular, in the area of consumer products and services).

People in large companies managing customer experience optimization units or business product strategies will undoubtedly use product recommender systems to enhance or improve their customer shopping experience. Today, integrating product and service recommendation systems with product or service delivery strategies has become one of the standards of large multinational organizations and companies to outperform competitors. These companies use product recommendation systems and services to "personalize" user-friendly content and products.

2 Problem Formulation

Recommendation systems can be used in a variety of areas. These systems are used to generate Playlists on movie and music services such as Netflix and YouTube. It is also used in services such as Amazon to offer products to users and in platforms such as social networks to offer content to users. In this context, a recommender system is implemented to recommend users movies they may like.

3 Methodology

A recommender system is an information refinement model that scores or ranks items for the user.

3.1 Recommendation approaches

Recommender systems often use three ranking methods, each of which is described below.

3.1.1 Content-Based approach

In some recommender systems, content-based refinement may be used, which offers suggestions based on the user's past choices (for example, in the movie recommendation web application, if the user has played a few sci-fi films in the past, the recommendation system may suggest recent sci-fi films that he has not yet purchased).

Moreover, content-based refinement offers suggestions using the content of the objects intended for bid. Therefore, certain content may be analyzed, such as text, images, audio. For example, in the use of movie recommendation, to recommend videos to the user "A", the content-based recommender system tries to understand the commonalities between videos that user "A" has previously rated high (specific actors, directors, genres, Subject, etc.). Then, it only recommends movies that have a high degree of similarity to each of the user preferences.

3.1.2 Collaborative approach

In Collaborative item-based recommender systems, recommendations for a particular user are predicted based on items previously rated by similar users. For example, in a movie recommendation application, to recommend videos to the user "A", the collaboration-based recommender system tries to find the user like "A", for example, other users who have similar preferences in movies. Then, only the most popular movies of them are recommended.

3.1.3 Hybrid Approach

It is a combination of previous methods that have tried to take advantage of those methods by combining them and cover their limitations.

3.1.4 Clustering Approach

Clustering is the process by which a set of objects can be divided into separate groups. Each division is called a cluster. The members of each cluster are very similar in their characteristics, and the similarity between the clusters is minimal. In such a case, the purpose of clustering is to assign labels to objects that indicate the membership of each object in the cluster.

3.2 Techniques

Different approaches and methods are used for making our recommendation systems. In this section, we review these approaches and methods.

3.2.1 TF-IDF

The value of TF-IDF stands for two words: TF means Term Frequency, which means the number of repetitions of a word in a text, and IDF means Inverse Document Frequency, which can be translated to the number of repetitions in the text. TF-IDF is a way to measure the importance of a word in a document. TF measures that a term is repeated several times in a document and the IDF is used to measure the significance of a term.

3.2.2 Cosine Similarity

The cosine similarity is a measure of the similarity between two nonzero vectors. Using this formula, we can find the similarity between two documents d1 and d2:

$$Cosine\ Similarity\ (d1, d2)\ =\ Dot\ product(d1, d2)\ /\ ||d1||\ *\ ||d2||$$

3.2.3 Pearson correlation coefficient

In statistical discussions, the Pearson correlation coefficient measures the amount of linear correlation between two random variables. The value of this coefficient varies between 0 and 1, where "1" means complete positive correlation, "0" means no correlation, and "-1" means perfect negative correlation. This coefficient, which is widely used in statistics, was compiled by Carl Pearson based on the original idea of Francis Galton.

3.2.4 SVD

Just as we can break a number into prime numbers, we can do the same for matrices. That is, we break down a matrix into its constituent factors. One of the methods of this analysis is Singular Value Decomposition or SVD.

In short, SVD is a method that decomposes one matrix into three other matrices. For example, matrix A Can be written as follows:

$$A = USV^T$$

Matrix A : $m \times n$

Matrix U : $m \times m$

Matrix S : $m \times n$

Matrix V : $n \times n$

3.2.5 NMF

NMF is used for feature extraction and is generally seen to be useful when there are many attributes, mainly when the attributes are vague or are not reliable predictors. By combining attributes, NMF can display patterns, topics, or themes that have importance.

3.2.6 k-Nearest Neighbors

K-Nearest Neighbors is a non-parametric method used in data mining, machine learning, and pattern recognition. According to the website kdnuggets, the k-nearest algorithm is one of the ten most widely used algorithms in various machine learning and data mining projects, both in industry and at university.

One of the main reasons why classification algorithms are so widely used is that "decision making" is one of the significant challenges in most analytics projects.

The distance between two points p and q is the size of the line segment that connects them (pq vector). In Cartesian coordinates, if p is (p_1, p_2, …, p_n), and q is (q_1, q_2, …, q_n) then:

$$d(p,q) = \sqrt{(p_{1-}q_1)^2 + (p_{2-}q_2)^2 + \cdots + (p_{n-}q_n)^2}$$

In the next step, all the features of a movie together will form a vector, and the number of them makes up our n-dimensional space. Implementation of the k-nearest neighborhood model is possible using the following pseudo-code:

1. Loading data
2. Initial selection of the value of k
3. To create predictive classes, repeat from 1 to the total number of data points:
    1) Calculate the distance of the test data from each row of the training data set. Euclidean distance is used here as the measurement distance, which is the most common method. Other measures that can also be used include Chebyshev, Cassinus, and others

2) Calculate the calculated distances in ascending order based on the amount of distance
3) Select the top k lines of the sorted array
4) Classes with the highest repetition will be received in these lines
5) Return the value of the predicted class

3.2.7 k-NN with means

k-NN means algorithms are directly derived from a basic nearest neighbors approach. In this algorithms, the actual number of neighbors that are aggregated to compute an estimation is necessarily less than or equal to k. It is a basic collaborative filtering algorithm, taking into account the mean ratings of each user.

3.2.8 KMeans

There are many methods and algorithms for converting objects into similar groups. The k-means algorithm is one of the simplest and most popular algorithms used in Data Mining, especially in the field of Unsupervised Learning.

The method's objective is to Gain Points as Cluster Centers. These points are the average of the points belonging to each cluster. Assign each data sample to a cluster whose data has the least distance to the center of that cluster. In the simplest form of this method, the number of clusters needed is chosen randomly. The data are then assigned to one of these clusters with respect to the similarity (similarity), and thus new clusters are obtained. By repeating the same procedure, we can calculate new centers for each iteration by averaging the data and assigning the data again to new clusters. This process will continue until there is no change in the data.

In the K-means algorithm, k members (which k is the number of clusters) are randomly selected from n members as the centers of the clusters. The remaining n - k members are then allocated to the nearest cluster. After assigning all members to the cluster centers, they are recalculated and assigned to the clusters with the new centers, and this will continue until the cluster centers remain constant.

In this method, the optimization of an Object Function is used. Clustering responses in this way may be minimized or maximized by the target function. This means that if the criterion is the "Distance Measure" between objects, the objective function will be based on minimization. The answer to the clustering operation is to find clusters that minimize the distance between the objects in each cluster. Conversely, if the Dissimilarity Function is used to measure the similarity of objects, they select the target function so that the clustering response maximizes its value per cluster.

Suppose observations $(x_1, x_2, \ldots, x_n)$ have to be divided into k clusters. We illustrate these clusters as $S = \{S_1, S_2, \ldots, S_k\}$. Cluster members should be selected from the observations to minimize the within-cluster sum of squares (WCSS) function, which is one-dimensional like variance. Therefore, the objective function in this algorithm is written as follows:

$$\underset{\mathrm{S}}{\arg\min}\sum_{i=1}^{k}\sum_{\mathrm{x}\in S_i}\left\|\mathrm{x}-\mu_i\right\|^2 = \underset{\mathrm{S}}{\arg\min}\sum_{i=1}^{k}|S_i|\operatorname{Var}S_i$$

Here μ_i represents means of the cluster S_i, and $|S_i|$ represents the number of cluster i_{th} members. However, it can be shown that minimizing this value, means maximizing the mean of squares between points in different clusters (BCSS) because, according to the general variance law, with decreasing WCSS, the BCSS value increases because of the variance The whole is fixed.

3.2.9 Hierarchical

Hierarchical clustering algorithms are one of the most popular and widely used clustering methods. These algorithms are divided into two sets of agglomerative and partitioning. In the agglomerative method, also referred to as the bottom-up method, each sample is initially a cluster, and at each stage, the samples stick together to form larger clusters, and eventually, all the samples form a large cluster. But in partitioning, the opposite is the case; that is, all the samples are considered as one large cluster and then divided into smaller clusters at each stage until each sample is a cluster.

In this method, in contrast to k-mean clustering, each observation may be subdivided into more than one cluster because clusters are formed based on different levels of distance. Thus each cluster may be subdivided into another cluster at a distance. However, clustering is a method that classifies it into similar groups by observing the "Features" or "Attributes" of the observations.

To perform the calculations related to this clustering method, we need two distance (similarity) criteria. 1)The amount of distance between the pair of observations, 2)The amount of distance between the clusters. In the first case, distance functions can be used for quantitative or qualitative data. So if the data is small, for example, the Euclidean or Manhattan distance can be used. Also, qualitative data can also be obtained from Simple Matching or Hamming Distance data.

Usually, prior to the start of the cumulative hierarchical clustering steps, a Distance Matrix or Similarity Matrix is used to accelerate the computation. This matrix shows the distance between the pairs of observations. Of course, the type of function to be measured by the distance is influenced by the values in this matrix.

In hierarchical clustering or HAC, with respect to the values of this matrix, the observations or clusters having the least distance (most similarity) merge and form a new cluster. Next, again the gap between the new observations or clusters is updated by the distance matrix, computed, and the integration process continues so that only one cluster remains.

The result of hierarchical clustering is usually shown by Dendrogram. A dendrogram is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from hierarchical clustering. The main use of a dendrogram is to work out the best way to allocate objects to clusters.

3.3 Python Libraries

In this project, Several Python libraries are used. They are listed below.

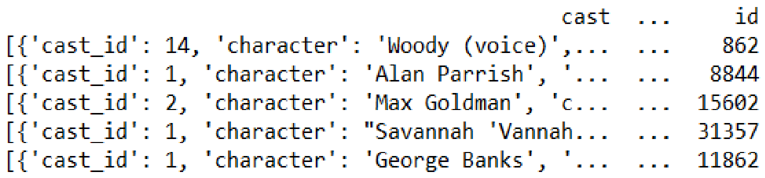
- pandas
- numpy
- scipy
- scikit-learn
- scikit-surprise

4 Data Description and Data Sources

GroupLens, a research group at the University of Minnesota, has generously released the MovieLens dataset. This dataset contains approximately 1 million ratings by two users for one movie. Besides, this dataset includes the genres of films and information about them. The data set mentioned above will be used to build the recommender system described here.

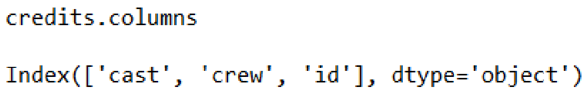
4.1 Data description

- It contains 100004 ratings and 1296 tag applications across 9125 movies. These data were created by 671 users between January 09, 1995, and October 16, 2016. This dataset was generated on October 17, 2016.
- Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. An id represents each user, and no other information is provided.
- The data are contained in the following files.
  - ✓ credits.csv



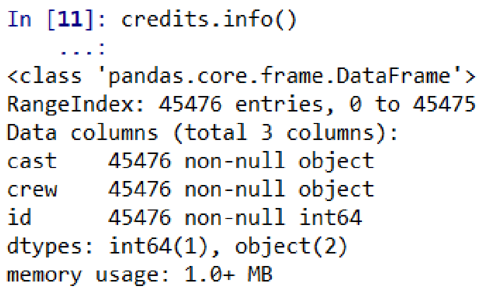
```
                                                cast  ...     id
[{'cast_id': 14, 'character': 'Woody (voice)',...  ...    862
[{'cast_id': 1, 'character': 'Alan Parrish', '...  ...   8844
[{'cast_id': 2, 'character': 'Max Goldman', 'c...  ...  15602
[{'cast_id': 1, 'character': "Savannah 'Vannah...  ...  31357
[{'cast_id': 1, 'character': 'George Banks', '...  ...  11862
```

Figure 4-1 Credit Head



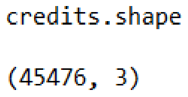
```
credits.columns

Index(['cast', 'crew', 'id'], dtype='object')
```

Figure 4-2 Credit Columns



```
In [11]: credits.info()
    ...:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45476 entries, 0 to 45475
Data columns (total 3 columns):
cast    45476 non-null object
crew    45476 non-null object
id      45476 non-null int64
dtypes: int64(1), object(2)
memory usage: 1.0+ MB
```

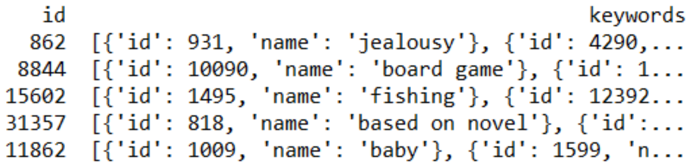
Figure 4-3 Credit Info



```
credits.shape

(45476, 3)
```

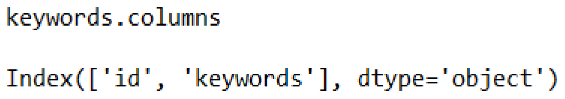
Figure 4-4 Credit Shape

- ✓ keywords.csv



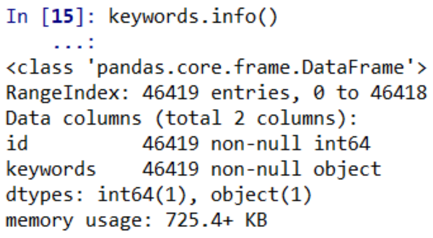
```
          id                                          keywords
  862  [{'id': 931, 'name': 'jealousy'}, {'id': 4290,...
 8844  [{'id': 10090, 'name': 'board game'}, {'id': 1...
15602  [{'id': 1495, 'name': 'fishing'}, {'id': 12392...
31357  [{'id': 818, 'name': 'based on novel'}, {'id':...
11862  [{'id': 1009, 'name': 'baby'}, {'id': 1599, 'n...
```

Figure 4-5 Keywords Head



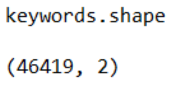
```
keywords.columns

Index(['id', 'keywords'], dtype='object')
```

Figure 4-6 Keyword Columns



```
In [15]: keywords.info()
    ...:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46419 entries, 0 to 46418
Data columns (total 2 columns):
id          46419 non-null int64
keywords    46419 non-null object
dtypes: int64(1), object(1)
memory usage: 725.4+ KB
```

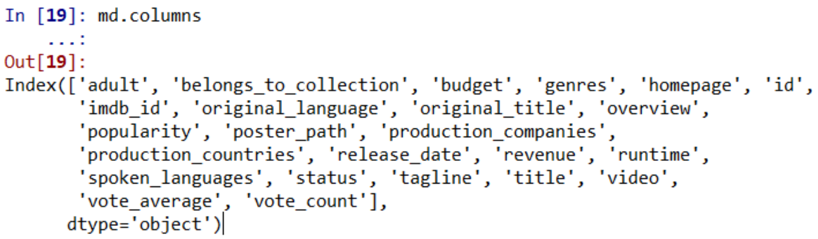
Figure 4-7 Keywords Info



```
keywords.shape

(46419, 2)
```

Figure 4-8 Keyword Shape

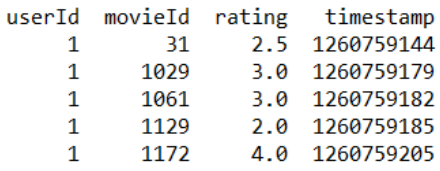
- ✓ movies_metadata.csv



```
In [19]: md.columns
    ...:
Out[19]:
Index(['adult', 'belongs_to_collection', 'budget', 'genres', 'homepage', 'id',
       'imdb_id', 'original_language', 'original_title', 'overview',
       'popularity', 'poster_path', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'video',
       'vote_average', 'vote_count'],
      dtype='object')
```

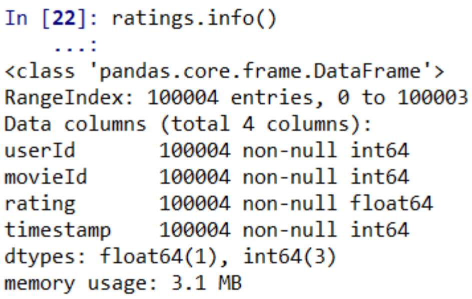
Figure 4-9 Movies Metadata Columns

- ✓ ratings.csv



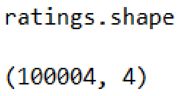
```
   userId  movieId  rating   timestamp
        1       31     2.5  1260759144
        1     1029     3.0  1260759179
        1     1061     3.0  1260759182
        1     1129     2.0  1260759185
        1     1172     4.0  1260759205
```

Figure 4-10 Rating head



```
In [22]: ratings.info()
    ...:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100004 entries, 0 to 100003
Data columns (total 4 columns):
userId       100004 non-null int64
movieId      100004 non-null int64
rating       100004 non-null float64
timestamp    100004 non-null int64
dtypes: float64(1), int64(3)
memory usage: 3.1 MB
```

Figure 4-11 Rating Info



```
ratings.shape

(100004, 4)
```

Figure 4-12 Rating Shape

- ✓ ratings_small.csv

4.2 Data Visualization

It can be seen that the histogram of movies and the ratings ,4 is the most.The ratings with 0.5 is less than the others that's because there are many ratings sites with integer numbers.

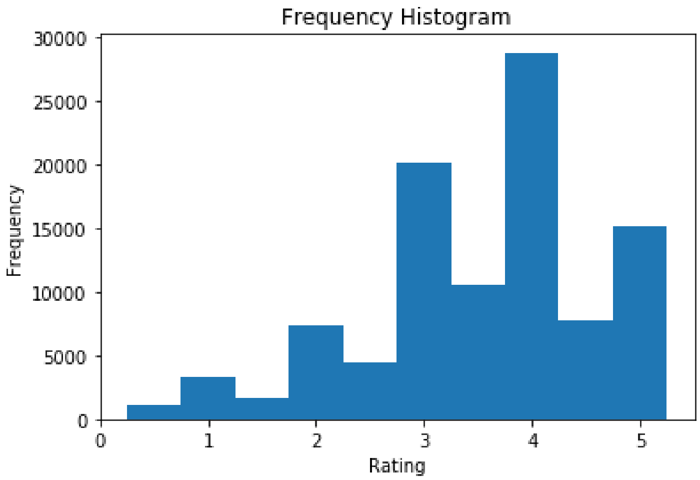


Figure 4-13 User's Rating Histogram

5 Evaluation Methods

In this project, Several evaluation methods will be used, each of which is described below.

5.1 Root-Mean-Square Error (RMSE)

Root-Mean-Square Error (RMSE) is the difference between the value predicted by the model or statistical estimator and the actual value is. RMSD is a useful tool for comparing prediction errors by a single dataset and is not used for comparing multiple datasets.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}}$$

5.2 Mean Absolute Error(MAE)

It is similar to RMSE. However, it uses the average magnitude of the errors in a set of predictions, without considering their direction.

$$MAE = \frac{1}{n}(\sum_{i=1}^{n}|P_i - O_i|)$$

5.3 Silhouette

The focus of the silhouette method relies on the quality of the clustering performed. This criterion determines how the data is distributed in the clusters. The higher the silhouette, the better the clustering quality.

In the silhouette method, the clustering algorithm is implemented for different values of k, and for each implementation, the silhouette criterion is calculated for each cluster member. Then, the obtained silhouettes are averaged. The optimal value of k is the value for which the maximum silhouette is obtained.

5.4 Elbow method

The main idea behind clustering methods such as k-means is to obtain the number of clusters in such a way to minimize the sum of the intra-cluster data intervals (or the sum of the squares of the intra-cluster intervals). The sum of the intra-cluster intervals of the data shows the amount of clustering performed, and the goal is to minimize it as much as possible.

The elbow method considers the sum of the intra-cluster intervals of the data as a function of the number of clusters. This allows the number of clusters to be selected so that adding another cluster does not improve the WSS minimization.

5.5 Qualitative metrics

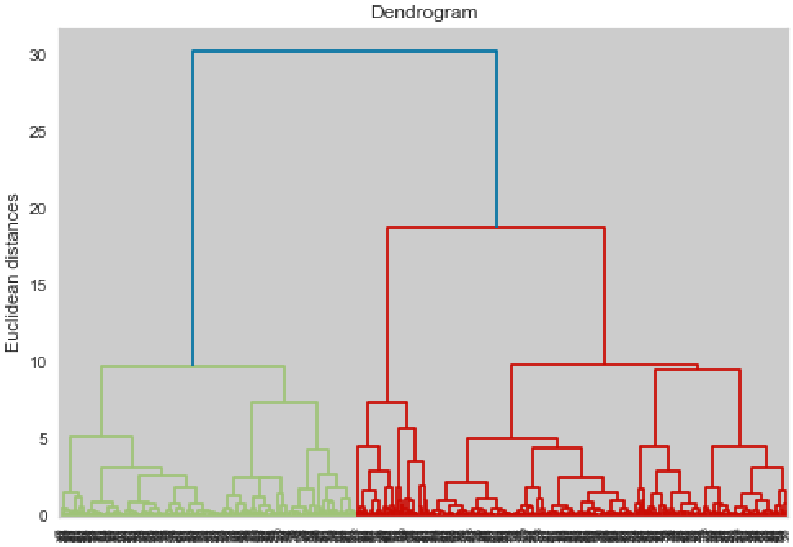
We do not have a quantitative metric to judge our machine's performance in some of the recommender systems. In these cases, the evaluation will have to be done qualitatively. For doing this, each of the members uses the system and tries to evaluate it qualitatively.

6 Results Expectations

Here are the results of our project:

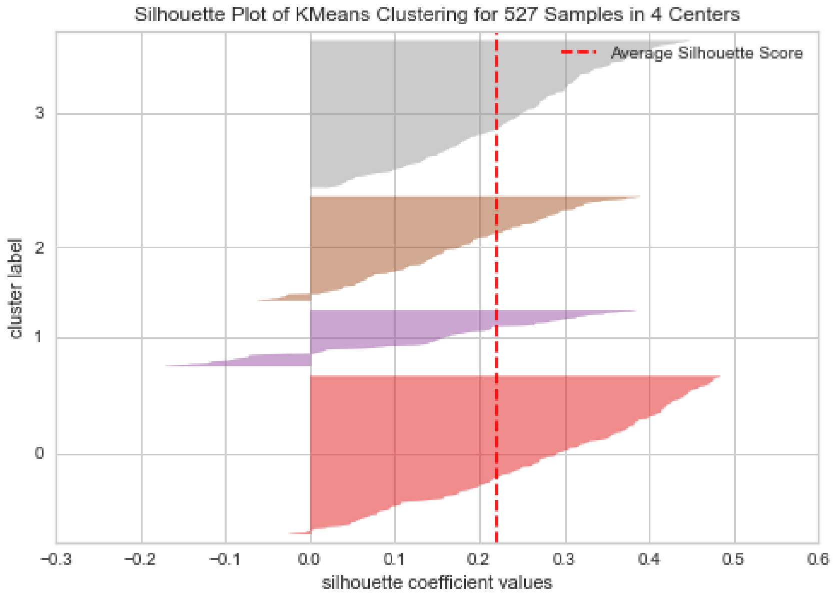
6.1 Number of clusters:

6.1.1 Dendrogram

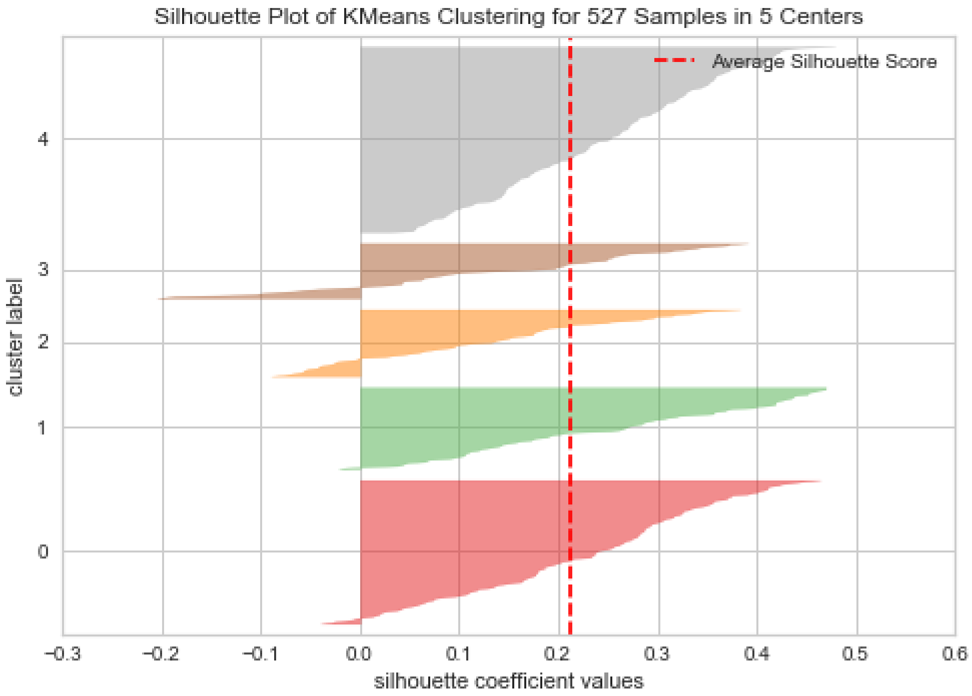


6.2 Silhouette

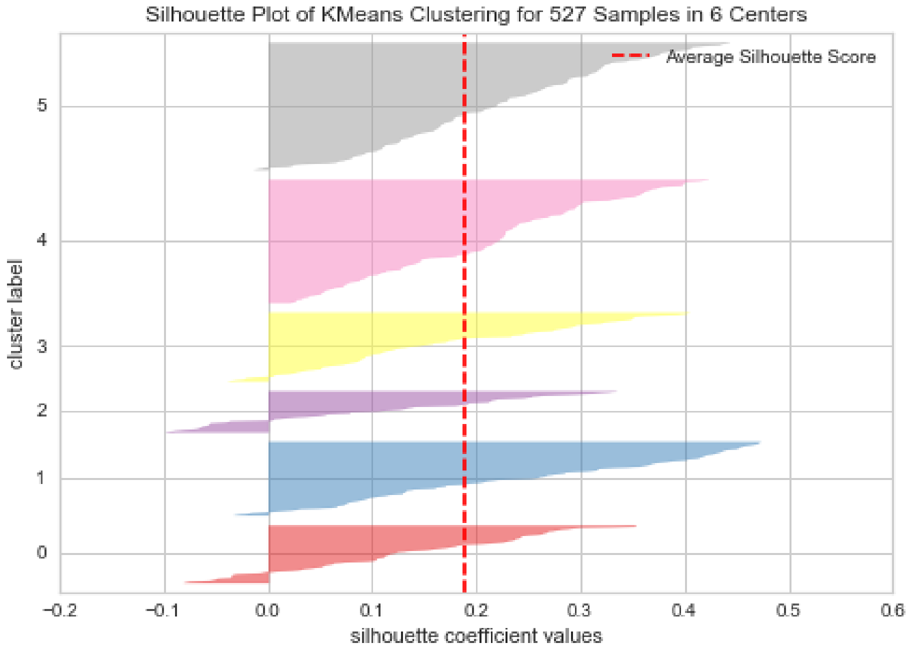
6.2.1.1 Silhouette plot for 4 centers



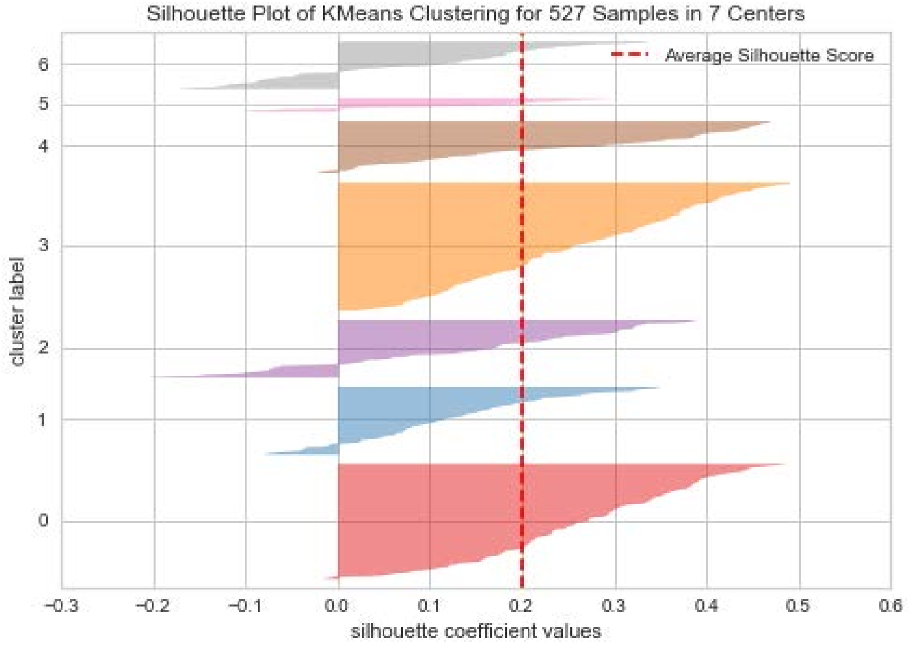
6.2.1.2 Silhouette plot for 5 centers



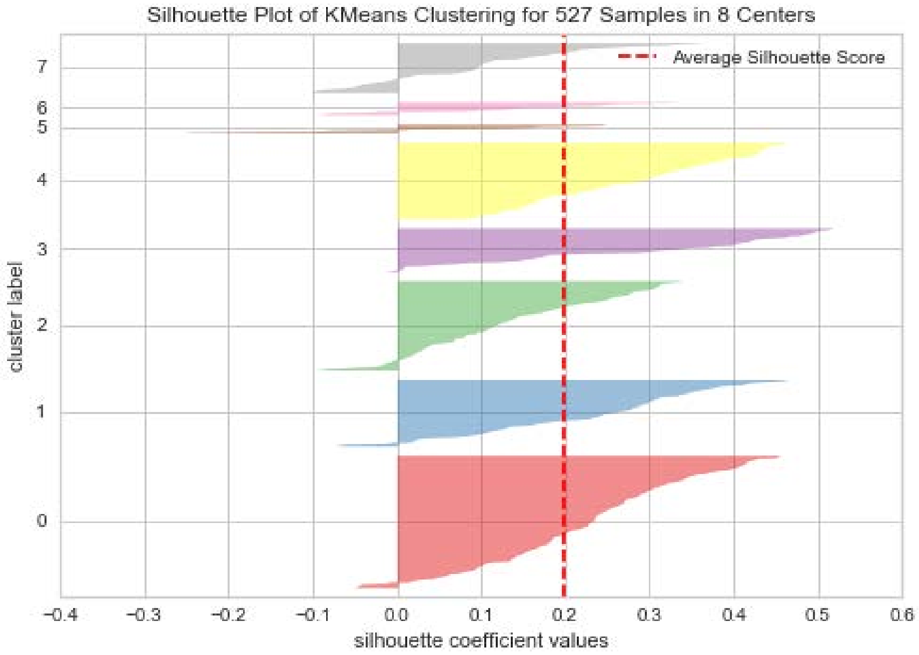
6.2.1.3 *Silhouette plot for 6 centers*



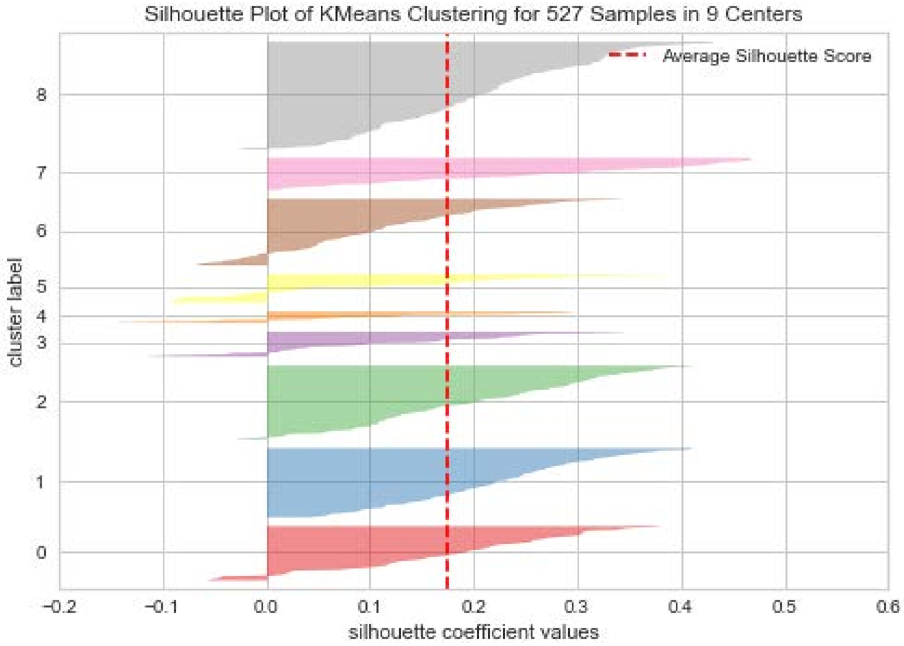
6.2.1.4 *Silhouette plot for 7 centers:*



6.2.1.5 Silhouette plot for 8 centers:



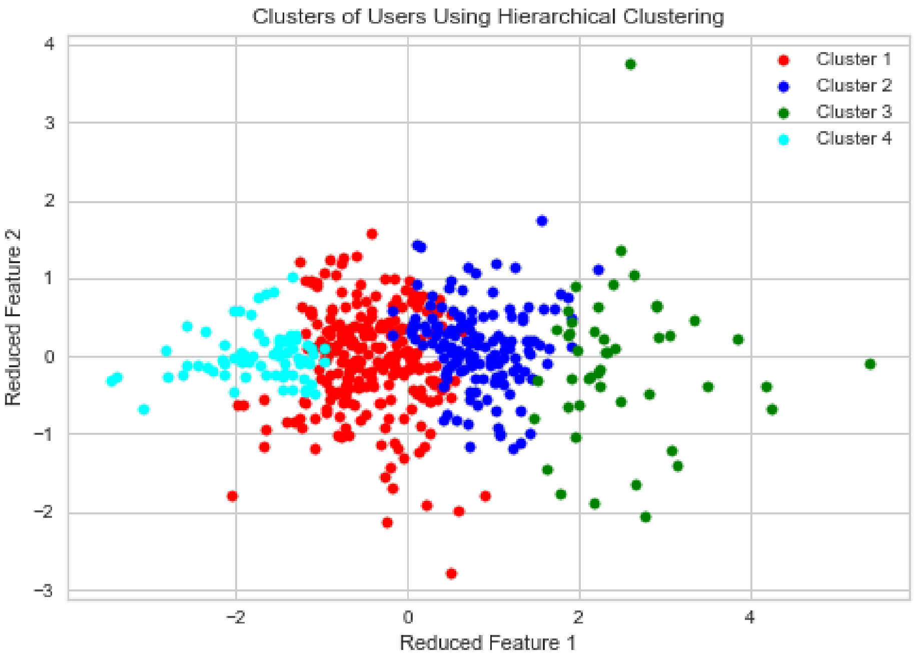
6.2.1.6 Silhouette plot for 9 centers



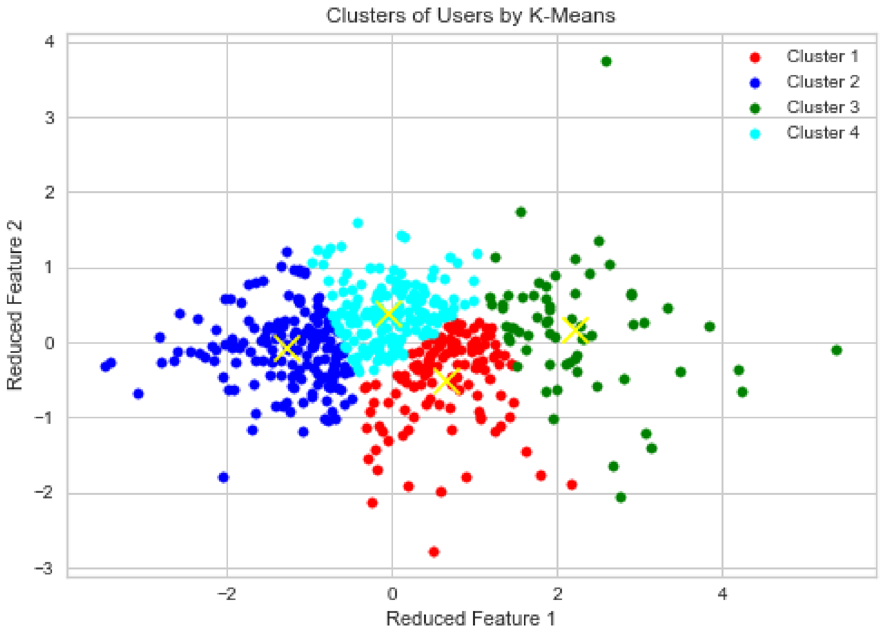
After considering the the above results we diceided that the best number of clusters that satisfy our evaluation is 4. The clustering result is as follows.

6.3 Clustering result

6.3.1 Clusters of users using Hierarchical Clustering



6.3.2 Clusters of users using K-Means



6.4 Recommendation systems

6.4.1 Simple recommendation system

The simple recommender suggests movies to users based on their popularity. The basic idea behind this system is that movies that are generally more popular have a higher chance of being liked by most of the users. The downside of this model is that it doesn't give personalized recommendations. The weighted rating formula to build the chars is as follows:

$$Weighted\ Rating(WR) = \left(\frac{v}{v+m}.R\right) + \left(\frac{m}{v+m}.C\right)$$

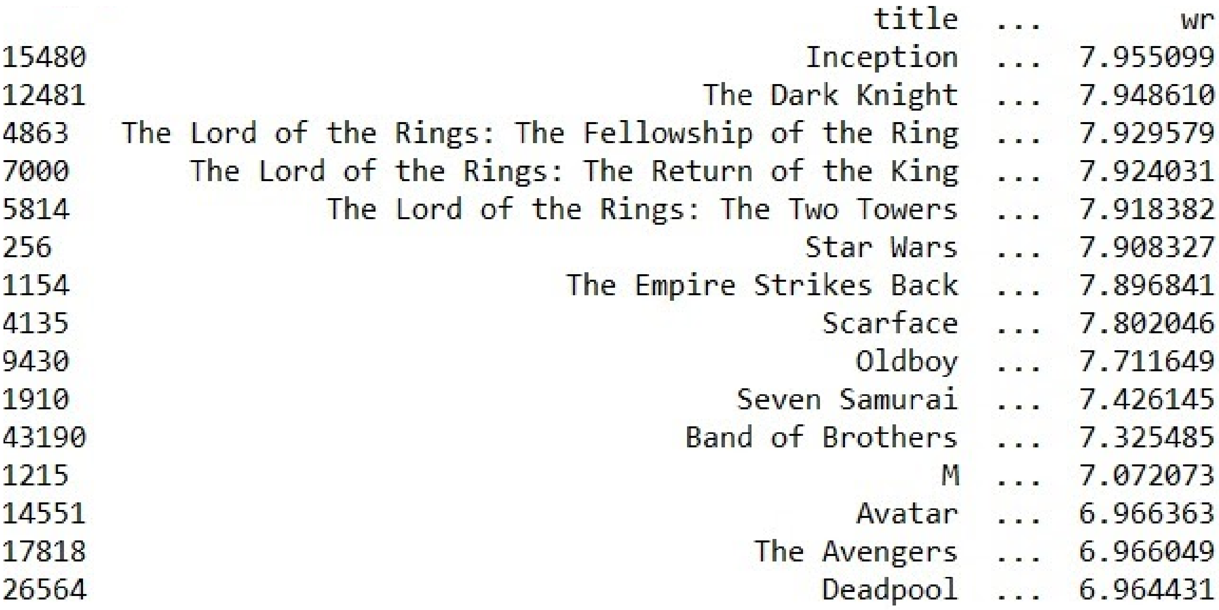
$$where,\ v\ is\ the\ number\ of\ votes\ for\ the\ movie$$

$$m\ is\ the\ minimum\ votes\ required\ to\ be\ listed\ in\ the\ chart$$

$$R\ is\ the\ average\ rating\ of\ the\ movie$$

$$C\ is\ the\ mean\ vote\ across\ the\ whole\ Data$$

Here are the results for the top 15 action movies:



```
                                                   title  ...        wr
15480                                          Inception  ...  7.955099
12481                                    The Dark Knight  ...  7.948610
4863   The Lord of the Rings: The Fellowship of the Ring  ...  7.929579
7000       The Lord of the Rings: The Return of the King  ...  7.924031
5814               The Lord of the Rings: The Two Towers  ...  7.918382
256                                            Star Wars  ...  7.908327
1154                             The Empire Strikes Back  ...  7.896841
4135                                           Scarface  ...  7.802046
9430                                             Oldboy  ...  7.711649
1910                                      Seven Samurai  ...  7.426145
43190                                  Band of Brothers  ...  7.325485
1215                                                  M  ...  7.072073
14551                                            Avatar  ...  6.966363
17818                                      The Avengers  ...  6.966049
26564                                          Deadpool  ...  6.964431
```

Figure 6-1 Simple Recommendation for Action movies

6.4.2 Content-based recommendation systems

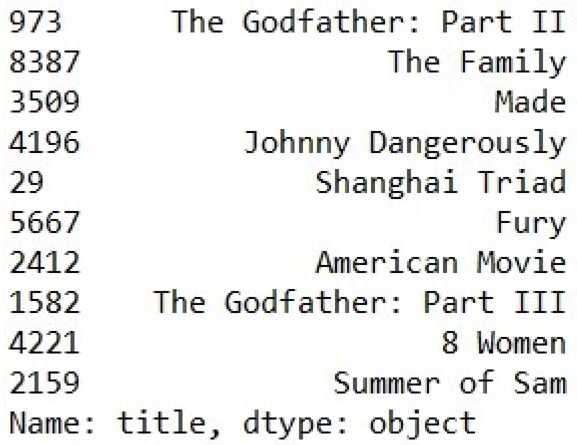
In this recommendation system, we used Cosine Similarity which is a measure of the similarity between two documents. The formula is as follows:

$$Cosine\ Similarity(d_1, d_2) = \frac{Dot\ Product(d_1,\ d_2)}{||d_1|| * ||d_2||}$$

We have used two different content-based recommendation systems that are as below:

6.4.2.1 Content-based RS Using movie descriptions and taglines Results

We used the Cosine similarity score to build the chart. Furthermore, we have used TF-IDF to use movie descriptions and taglines as our features. Here is the 10 top result based on The Godfather:



```
973      The Godfather: Part II
8387                 The Family
3509                       Made
4196        Johnny Dangerously
29               Shanghai Triad
5667                       Fury
2412              American Movie
1582    The Godfather: Part III
4221                    8 Women
2159              Summer of Sam
Name: title, dtype: object
```

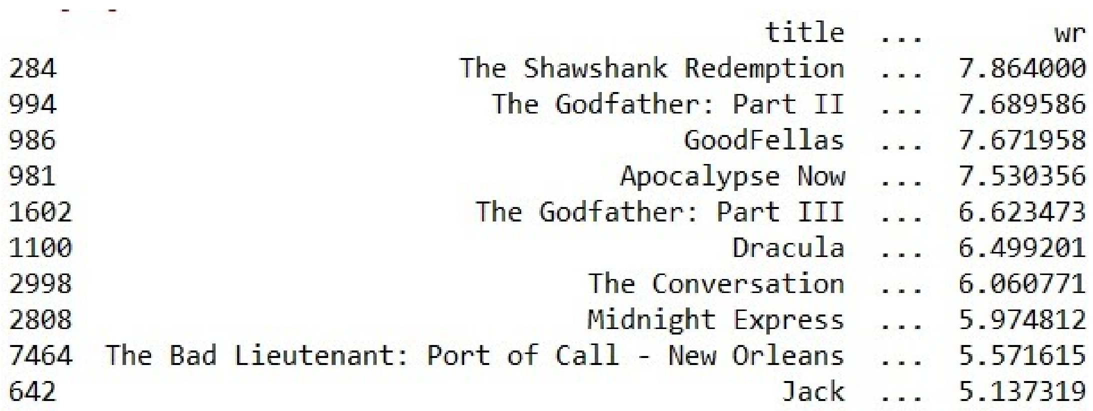
Figure 6-2 Recommendation for The Godfather by Content-based RS Using movie descriptions and taglines Results

We see that for The Godfather, our system can identify it like The Godfather film and afterward recommend other Godfather films as its top recommendations. But unfortunately, that is all this system can do at the moment.

This is not of much use for most people as it doesn't take into consideration essential features such as cast, crew, director, and genre, which determine the rating and the popularity of a movie.

6.4.2.2 Content-based RS: Using movie description, taglines, keywords, cast, director and genres

To improve our system, we added keywords, cast, director and genres. The result is as follows:



```
                                                 title  ...        wr
284                           The Shawshank Redemption  ...  7.864000
994                             The Godfather: Part II  ...  7.689586
986                                         GoodFellas  ...  7.671958
981                                     Apocalypse Now  ...  7.530356
1602                           The Godfather: Part III  ...  6.623473
1100                                           Dracula  ...  6.499201
2998                                  The Conversation  ...  6.060771
2808                                  Midnight Express  ...  5.974812
7464  The Bad Lieutenant: Port of Call - New Orleans  ...  5.571615
642                                               Jack  ...  5.137319
```

Figure 6-3 Recommendation for The Godfather by Content-based RS using more features

We are much more satisfied with the results I get this time around. The recommendations seem to have recognized other Francis Ford Coppola movies (due to the high weightage given to the director) and put them as top recommendations.

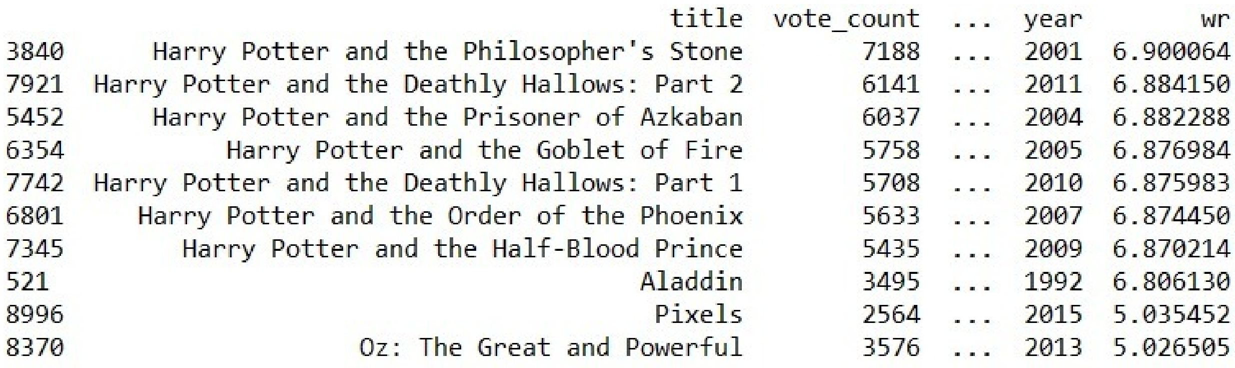
We can experiment on this engine by trying out different weights for our features (directors, actors, genres), limiting the number of keywords that can be used in the soup, weighing genres based on their frequency, only showing movies with the same languages, etc.

6.4.2.3 Content-based RS Evaluation

The Recommendation system has two significant limitations:

- It is only capable of suggesting movies that are close to a specific film. That is, it is not capable of capturing tastes and providing recommendations across genres.
- Also, the engine that we built is not personal. It doesn't capture the individual tastes, and Anyone querying our engine for recommendations based on a movie will receive the same movies, regardless of his/her liking.

We also do a Quantitative evaluation of our systems by using another movie (Harry Potter and the Chamber of Secrets). The table below shows the result.



```
                                          title  vote_count  ...  year        wr
3840     Harry Potter and the Philosopher's Stone        7188  ...  2001  6.900064
7921  Harry Potter and the Deathly Hallows: Part 2        6141  ...  2011  6.884150
5452     Harry Potter and the Prisoner of Azkaban        6037  ...  2004  6.882288
6354          Harry Potter and the Goblet of Fire        5758  ...  2005  6.876984
7742  Harry Potter and the Deathly Hallows: Part 1        5708  ...  2010  6.875983
6801    Harry Potter and the Order of the Phoenix        5633  ...  2007  6.874450
7345       Harry Potter and the Half-Blood Prince        5435  ...  2009  6.870214
521                                        Aladdin        3495  ...  1992  6.806130
8996                                         Pixels        2564  ...  2015  5.035452
8370                     Oz: The Great and Powerful        3576  ...  2013  5.026505
```

Figure 6-4Recommendation for Harry Potter and the Chamber of Secrets by Content-based R

It was predictable that the other seven harry potter movies have more similarity with the movie so our recommendation system recommends other harry potter movies. The other film is Aladdin, which has some magical scenes and characters. Pixels also is a science fiction movie directed by the same director. Oz the great and powerful, which is another movie about withes and their world the same as harry potter.

In conclusion, one can say that our systems could capture similar films with Harry Potter and the Chamber of Secrets, and this is a satisfactory result. However, pixels have poor ratings. In the next section, we will improve our system and consider the ratings.

6.4.3 CF-based recommendation system

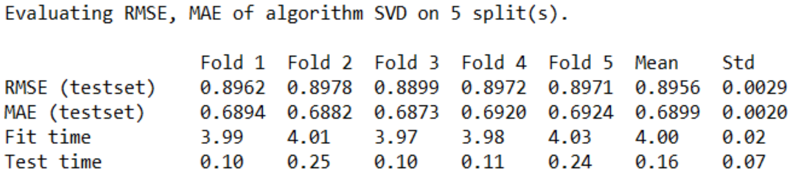
Because of the Limitation of our past methods, We also used a CF-based recommendation system.

For CF-based Recommendation systems, we used the Surprise library in python and tried different algorithms:

- SVD
- NMF
- KNN
- KNN With Means
- Average of SVD, NMF, KNN, and KNN With Means

6.4.3.1 SVD Result

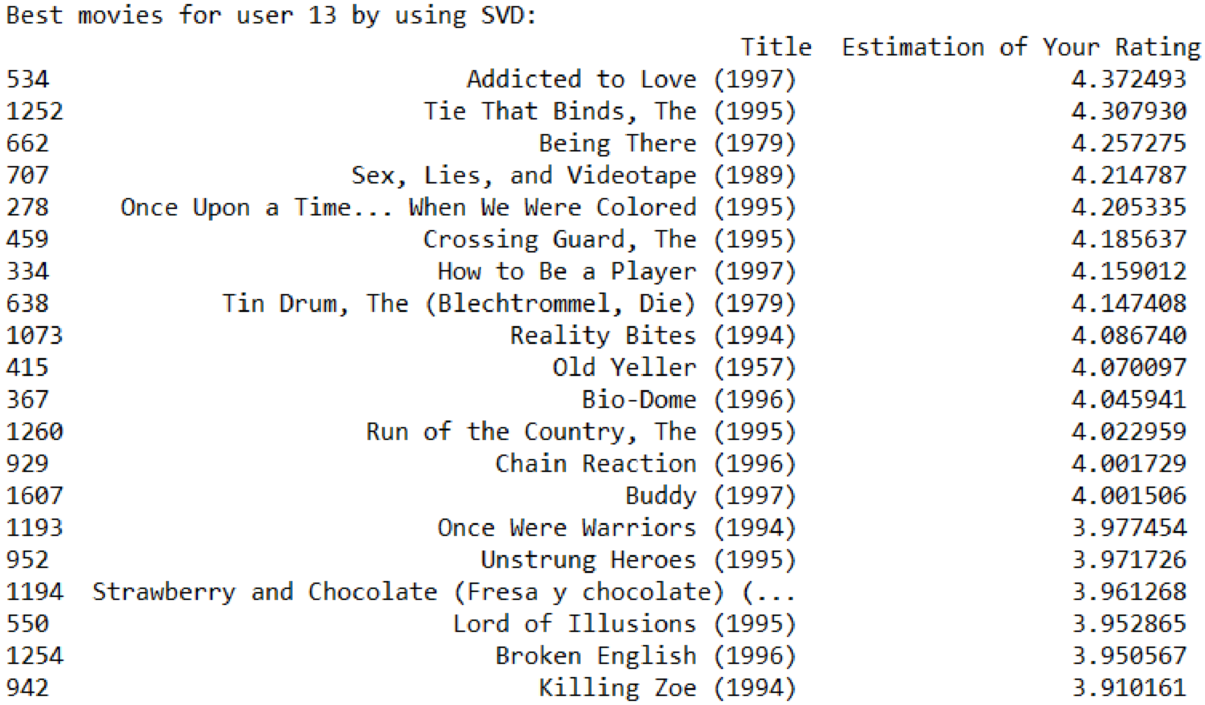
The table below shows RMSE and MAR for SVD.



```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.8962  0.8978  0.8899  0.8972  0.8971  0.8956  0.0029
MAE (testset)     0.6894  0.6882  0.6873  0.6920  0.6924  0.6899  0.0020
Fit time          3.99    4.01    3.97    3.98    4.03    4.00    0.02
Test time         0.10    0.25    0.10    0.11    0.24    0.16    0.07
```

Figure 6-5 SVD Evaluation

SVD prediction for user 13 The Godfather's rating is 3.22, and the table below shows the 20 best movies for user 13.

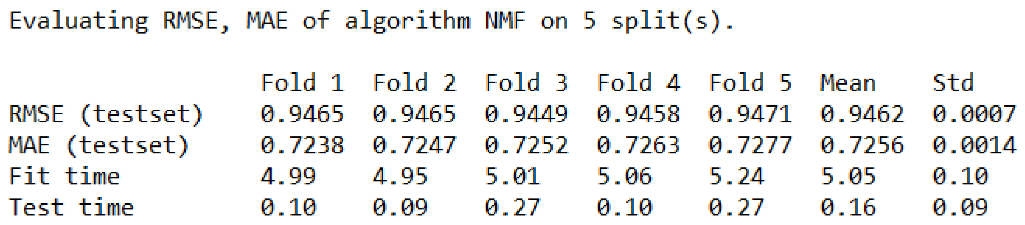


```
Best movies for user 13 by using SVD:
                                                 Title  Estimation of Your Rating
534                              Addicted to Love (1997)                 4.372493
1252                           Tie That Binds, The (1995)                 4.307930
662                                  Being There (1979)                 4.257275
707                     Sex, Lies, and Videotape (1989)                 4.214787
278       Once Upon a Time... When We Were Colored (1995)                 4.205335
459                            Crossing Guard, The (1995)                 4.185637
334                              How to Be a Player (1997)                 4.159012
638               Tin Drum, The (Blechtrommel, Die) (1979)                 4.147408
1073                                Reality Bites (1994)                 4.086740
415                                    Old Yeller (1957)                 4.070097
367                                      Bio-Dome (1996)                 4.045941
1260                        Run of the Country, The (1995)                 4.022959
929                                Chain Reaction (1996)                 4.001729
1607                                        Buddy (1997)                 4.001506
1193                            Once Were Warriors (1994)                 3.977454
952                               Unstrung Heroes (1995)                 3.971726
1194  Strawberry and Chocolate (Fresa y chocolate) (...                 3.961268
550                              Lord of Illusions (1995)                 3.952865
1254                               Broken English (1996)                 3.950567
942                                   Killing Zoe (1994)                 3.910161
```

Figure 6-6 SVD Recommendation for user 13

6.4.3.2 NMF Result

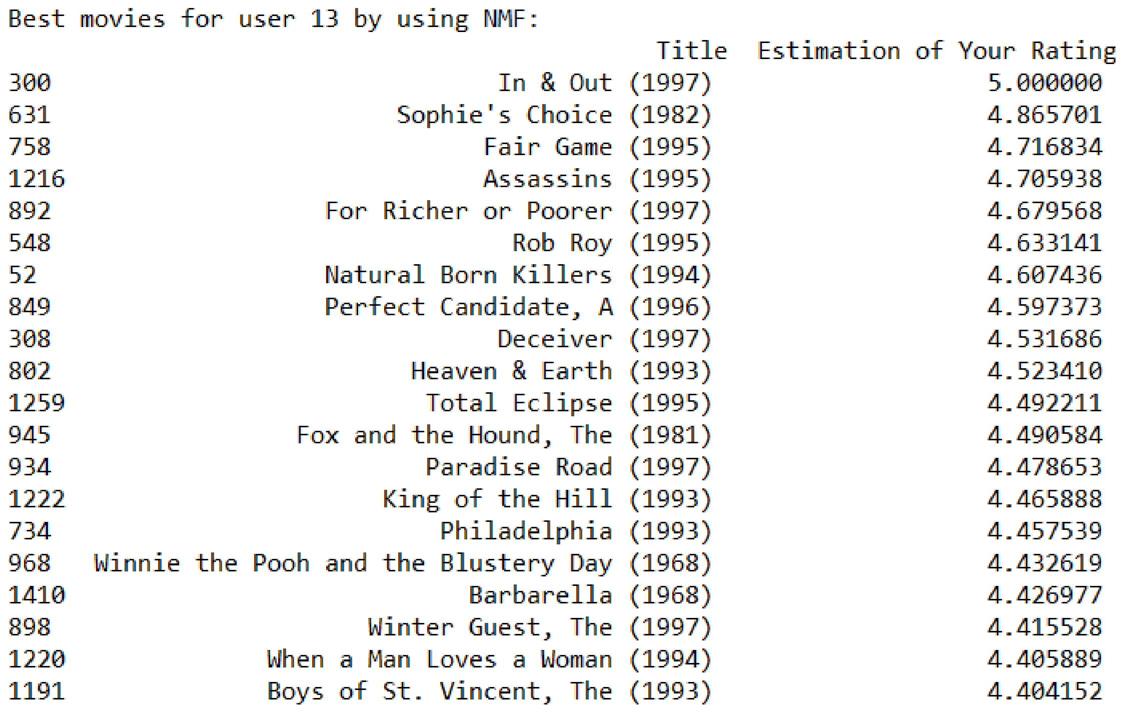
The table below shows RMSE and MAR for NMF.



```
Evaluating RMSE, MAE of algorithm NMF on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9465  0.9465  0.9449  0.9458  0.9471  0.9462  0.0007
MAE (testset)     0.7238  0.7247  0.7252  0.7263  0.7277  0.7256  0.0014
Fit time          4.99    4.95    5.01    5.06    5.24    5.05    0.10
Test time         0.10    0.09    0.27    0.10    0.27    0.16    0.09
```

Figure 6-7 NMF Evaluation

NMF prediction for user 13 The Godfather's rating is 2.91, and the table below shows the 20 best movies for user 13.

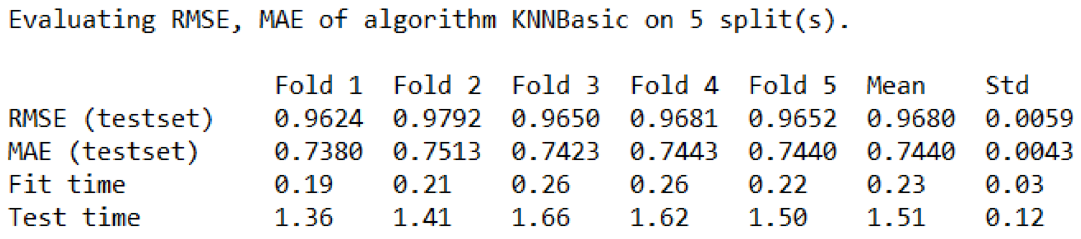


```
Best movies for user 13 by using NMF:
                                             Title  Estimation of Your Rating
300                               In & Out (1997)                   5.000000
631                        Sophie's Choice (1982)                   4.865701
758                              Fair Game (1995)                   4.716834
1216                             Assassins (1995)                   4.705938
892                     For Richer or Poorer (1997)                 4.679568
548                                Rob Roy (1995)                   4.633141
52                    Natural Born Killers (1994)                   4.607436
849                   Perfect Candidate, A (1996)                   4.597373
308                               Deceiver (1997)                   4.531686
802                         Heaven & Earth (1993)                   4.523410
1259                         Total Eclipse (1995)                   4.492211
945                   Fox and the Hound, The (1981)                 4.490584
934                          Paradise Road (1997)                   4.478653
1222                      King of the Hill (1993)                   4.465888
734                           Philadelphia (1993)                   4.457539
968     Winnie the Pooh and the Blustery Day (1968)                 4.432619
1410                            Barbarella (1968)                   4.426977
898                       Winter Guest, The (1997)                  4.415528
1220                When a Man Loves a Woman (1994)                 4.405889
1191                Boys of St. Vincent, The (1993)                 4.404152
```

Figure 6-8 NMF Recommendation for user 13

6.4.3.3 KNN Result

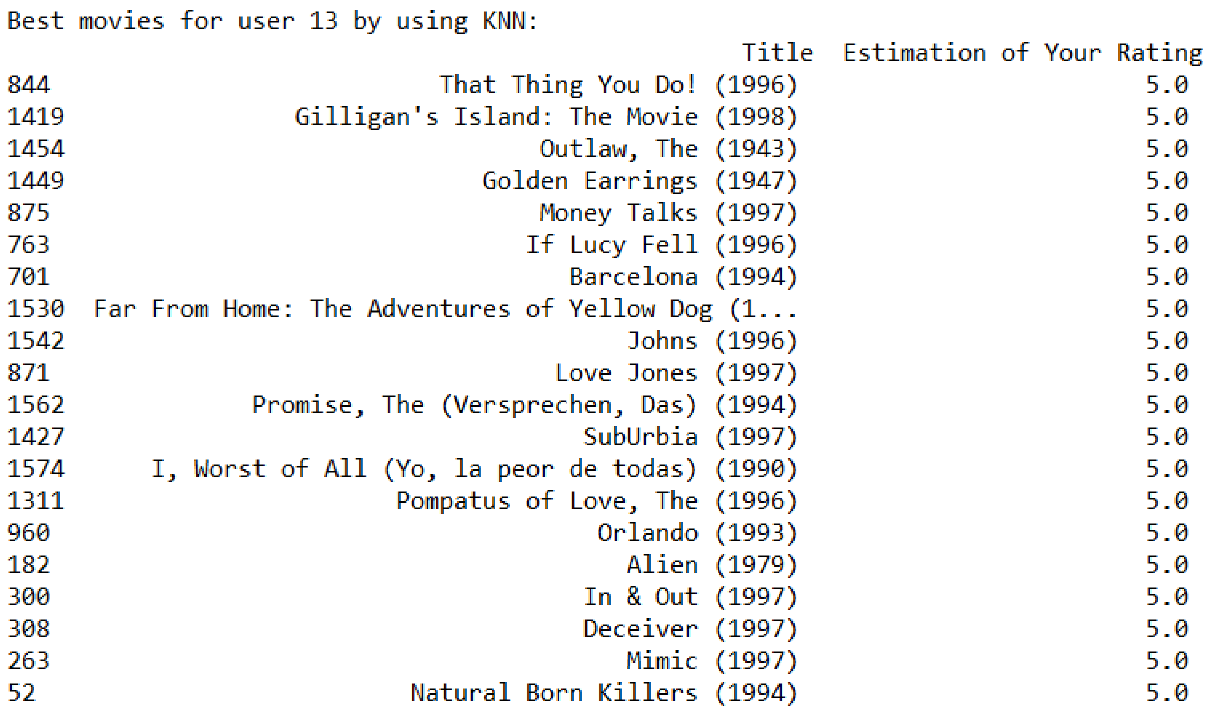
The table below shows RMSE and MAR for KNN.



```
Evaluating RMSE, MAE of algorithm KNNBasic on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9624  0.9792  0.9650  0.9681  0.9652  0.9680  0.0059
MAE (testset)     0.7380  0.7513  0.7423  0.7443  0.7440  0.7440  0.0043
Fit time          0.19    0.21    0.26    0.26    0.22    0.23    0.03
Test time         1.36    1.41    1.66    1.62    1.50    1.51    0.12
```

Figure 6-9 KNN Evaluation

KNN prediction for user 13 The Godfather's rating is 3.33, and the table below shows the 20 best movies for user 13.

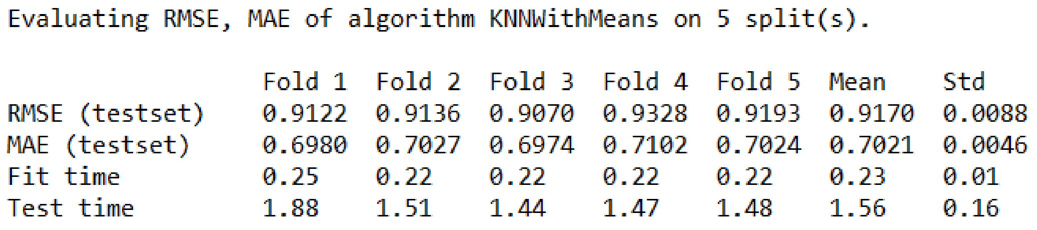


```
Best movies for user 13 by using KNN:
                                                    Title  Estimation of Your Rating
844                                That Thing You Do! (1996)                       5.0
1419                   Gilligan's Island: The Movie (1998)                       5.0
1454                                   Outlaw, The (1943)                       5.0
1449                               Golden Earrings (1947)                       5.0
875                                    Money Talks (1997)                       5.0
763                                   If Lucy Fell (1996)                       5.0
701                                      Barcelona (1994)                       5.0
1530  Far From Home: The Adventures of Yellow Dog (1...                       5.0
1542                                          Johns (1996)                       5.0
871                                      Love Jones (1997)                       5.0
1562              Promise, The (Versprechen, Das) (1994)                       5.0
1427                                       SubUrbia (1997)                       5.0
1574      I, Worst of All (Yo, la peor de todas) (1990)                       5.0
1311                           Pompatus of Love, The (1996)                       5.0
960                                         Orlando (1993)                       5.0
182                                           Alien (1979)                       5.0
300                                        In & Out (1997)                       5.0
308                                        Deceiver (1997)                       5.0
263                                           Mimic (1997)                       5.0
52                             Natural Born Killers (1994)                       5.0
```

Figure 6-10 KNN Recommendation for user 13

6.4.3.4 KNN with Means Result

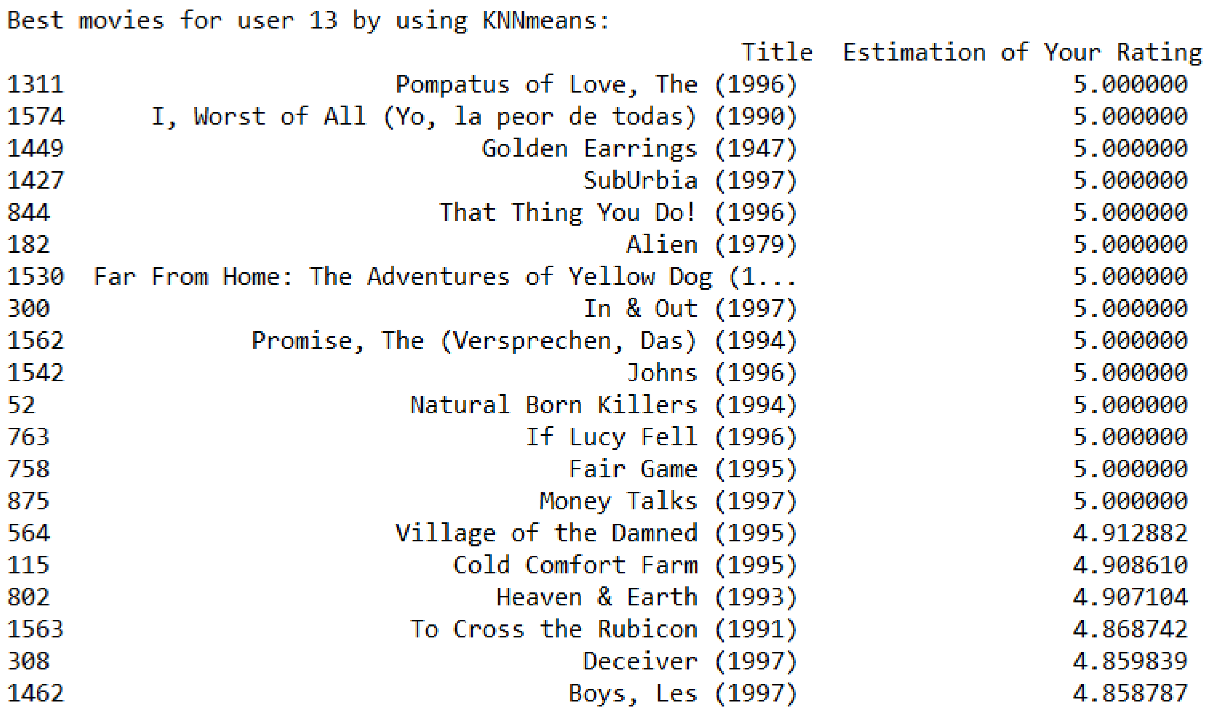
The table below shows RMSE, and MAR for KNN with Means.



```
Evaluating RMSE, MAE of algorithm KNNWithMeans on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std     
RMSE (testset)    0.9122  0.9136  0.9070  0.9328  0.9193  0.9170  0.0088  
MAE (testset)     0.6980  0.7027  0.6974  0.7102  0.7024  0.7021  0.0046  
Fit time          0.25    0.22    0.22    0.22    0.22    0.23    0.01    
Test time         1.88    1.51    1.44    1.47    1.48    1.56    0.16    
```

Figure 6-11 KNN Evaluation

KNN with Means prediction for user 13 The Godfather’s rating is 3.65, and the table below shows the 20 best movies for user 13.

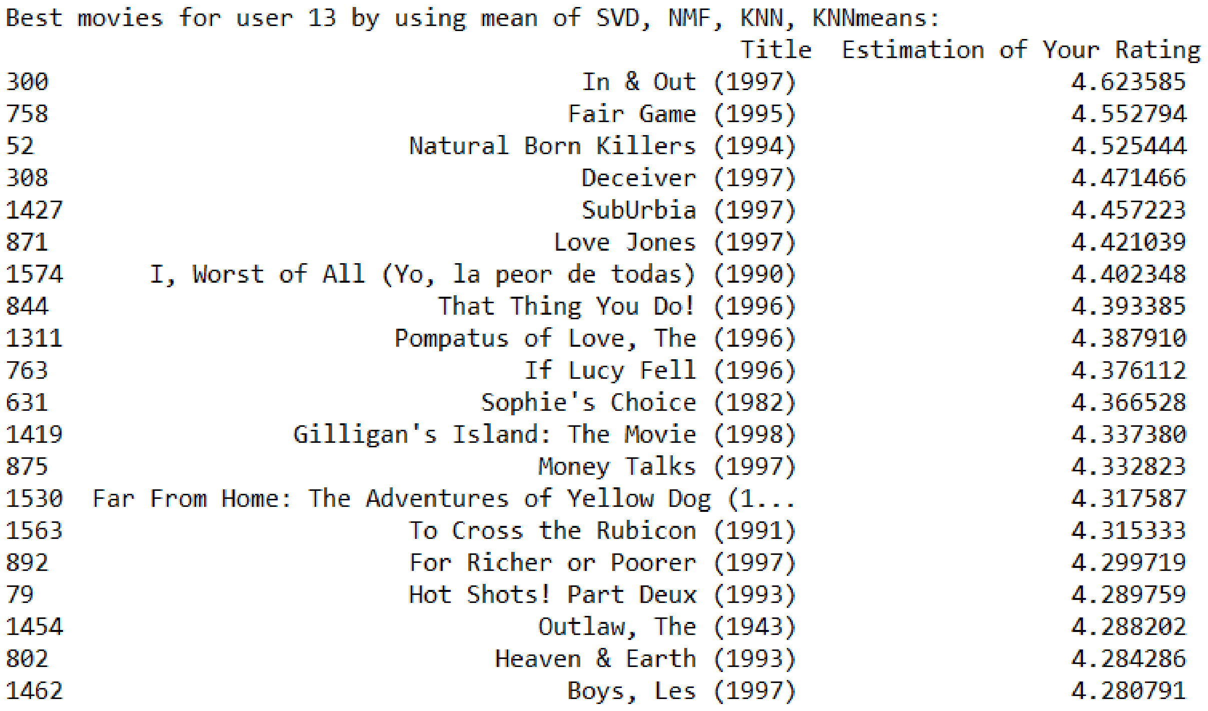


```
Best movies for user 13 by using KNNmeans:
                                                    Title  Estimation of Your Rating
1311                           Pompatus of Love, The (1996)                  5.000000
1574        I, Worst of All (Yo, la peor de todas) (1990)                  5.000000
1449                                 Golden Earrings (1947)                  5.000000
1427                                        SubUrbia (1997)                  5.000000
844                               That Thing You Do! (1996)                  5.000000
182                                            Alien (1979)                  5.000000
1530  Far From Home: The Adventures of Yellow Dog (1...                  5.000000
300                                         In & Out (1997)                  5.000000
1562            Promise, The (Versprechen, Das) (1994)                  5.000000
1542                                           Johns (1996)                  5.000000
52                               Natural Born Killers (1994)                  5.000000
763                                     If Lucy Fell (1996)                  5.000000
758                                        Fair Game (1995)                  5.000000
875                                      Money Talks (1997)                  5.000000
564                           Village of the Damned (1995)                  4.912882
115                                 Cold Comfort Farm (1995)                  4.908610
802                                    Heaven & Earth (1993)                  4.907104
1563                            To Cross the Rubicon (1991)                  4.868742
308                                         Deceiver (1997)                  4.859839
1462                                       Boys, Les (1997)                  4.858787
```

Figure 6-12 KNN with means Recommendation for user 13

6.4.3.5 *Average of the four methods Result*

We also used a mean of Average of SVD, NMF, KNN, and KNN With Means. The table below is recommendations based on the Average method.



```
Best movies for user 13 by using mean of SVD, NMF, KNN, KNNmeans:
                                                   Title  Estimation of Your Rating
300                                      In & Out (1997)                   4.623585
758                                     Fair Game (1995)                   4.552794
52                           Natural Born Killers (1994)                   4.525444
308                                      Deceiver (1997)                   4.471466
1427                                     SubUrbia (1997)                   4.457223
871                                    Love Jones (1997)                   4.421039
1574       I, Worst of All (Yo, la peor de todas) (1990)                   4.402348
844                            That Thing You Do! (1996)                   4.393385
1311                        Pompatus of Love, The (1996)                   4.387910
763                                  If Lucy Fell (1996)                   4.376112
631                               Sophie's Choice (1982)                   4.366528
1419                 Gilligan's Island: The Movie (1998)                   4.337380
875                                   Money Talks (1997)                   4.332823
1530   Far From Home: The Adventures of Yellow Dog (1...                   4.317587
1563                         To Cross the Rubicon (1991)                   4.315333
892                          For Richer or Poorer (1997)                   4.299719
79                           Hot Shots! Part Deux (1993)                   4.289759
1454                                  Outlaw, The (1943)                   4.288202
802                               Heaven & Earth (1993)                   4.284286
1462                                    Boys, Les (1997)                   4.280791
```

Figure 6-13 Average Recommendation for user 13

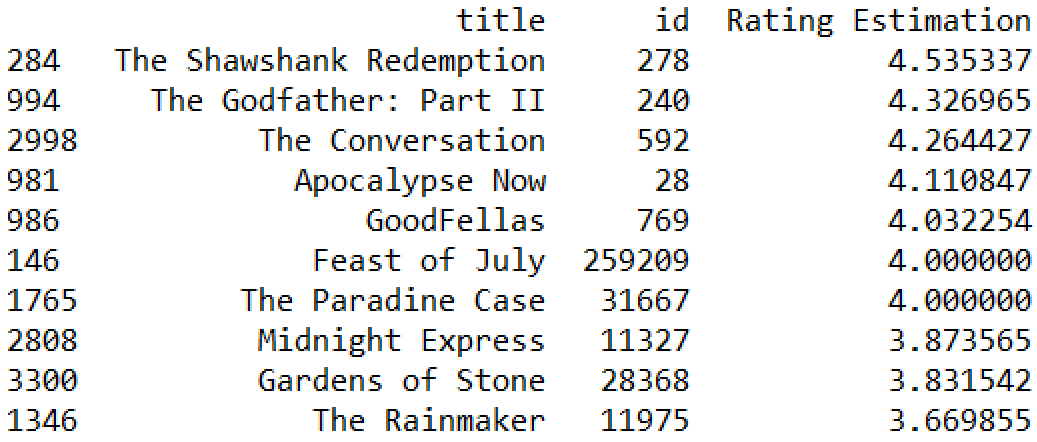
6.4.3.6 *Evaluation of CF-based Recommendation System*

As one can see, the best algorithm for the CF-based Recommendation system is SVD with the lowest errors. However, the CF-based Recommendation System has some significant limitations and disadvantages. It is hard for the system to include side features for a movie. For movie recommendations, the side features include, for example, Casts, directors, and movie descriptions. Including available side, features improve the quality of the model. It also Cannot handle fresh items.

6.5 Hybrid Recommendation System

It is a combination of previous methods that have tried to take advantage of those methods by combining them and cover their limitations. For doing this, User gives a movie name, and our system finds similar movies with the user request, and sort them based on the CF-based System.

Here are the result for user 13 requesting The Godfather.



```
                         title      id  Rating Estimation
284   The Shawshank Redemption     278           4.535337
994     The Godfather: Part II     240           4.326965
2998          The Conversation     592           4.264427
981             Apocalypse Now      28           4.110847
986                 GoodFellas     769           4.032254
146              Feast of July  259209           4.000000
1765         The Paradine Case   31667           4.000000
2808          Midnight Express   11327           3.873565
3300          Gardens of Stone   28368           3.831542
1346             The Rainmaker   11975           3.669855
```

Figure 6-14 Hybrid Recommendation for user 13 and The Godfather

As you can see, it includes The Shawshank Redemption, The Godfather: part II, and also some other movies of Francis Ford Coppola.