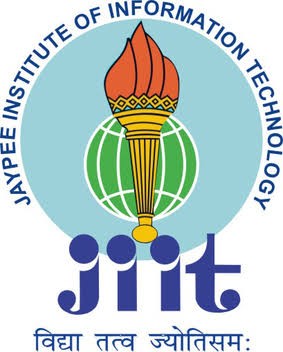
**MINOR PROJECT REPORT**

**RECOMMENDATION ENGINE**

**(Under COPACS Research Group)**

****

**Recommender Systems**

Recommender systems are widespread tools that are employed by a wide range of organizations and companies for recommending items such as movies, books and even employees for projects. But with the advent of big data it has become difficult to process the large amount of data for recommendations.

Recommender systems are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item. Recommender systems have become extremely common in recent years, and are applied in a variety of applications. The most popular ones are probably movies, music, news, books, research articles, search queries, social tags, and products in general.

**Problem Statement**

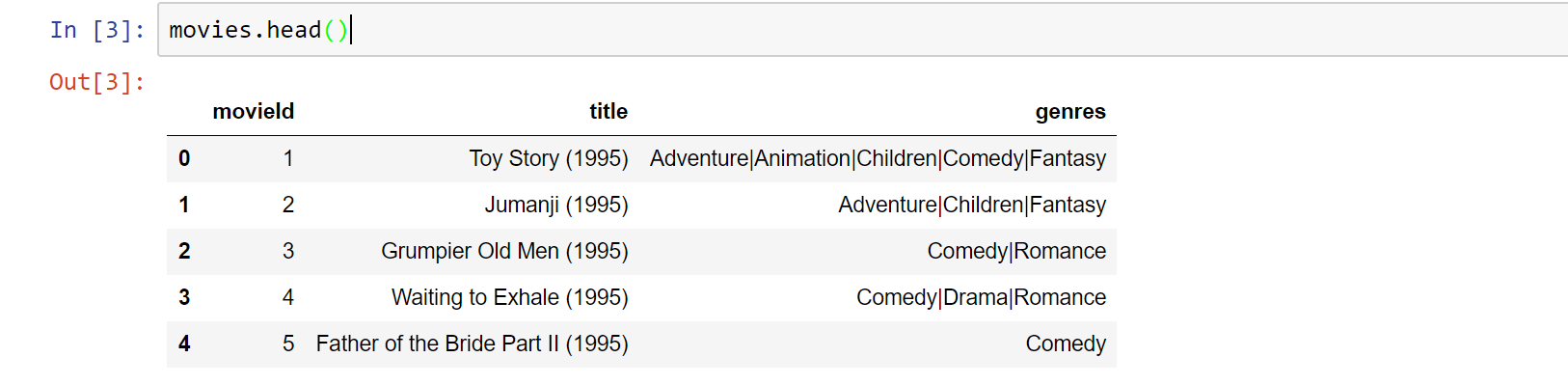
To design an online books recommendation engine, and we want to suggest for your clients’ books recommendations based on movies they watched or they searched for. Our goal is to calculate how similar these books and movies are and recommend the users something which they really like.

**Technologies Used**

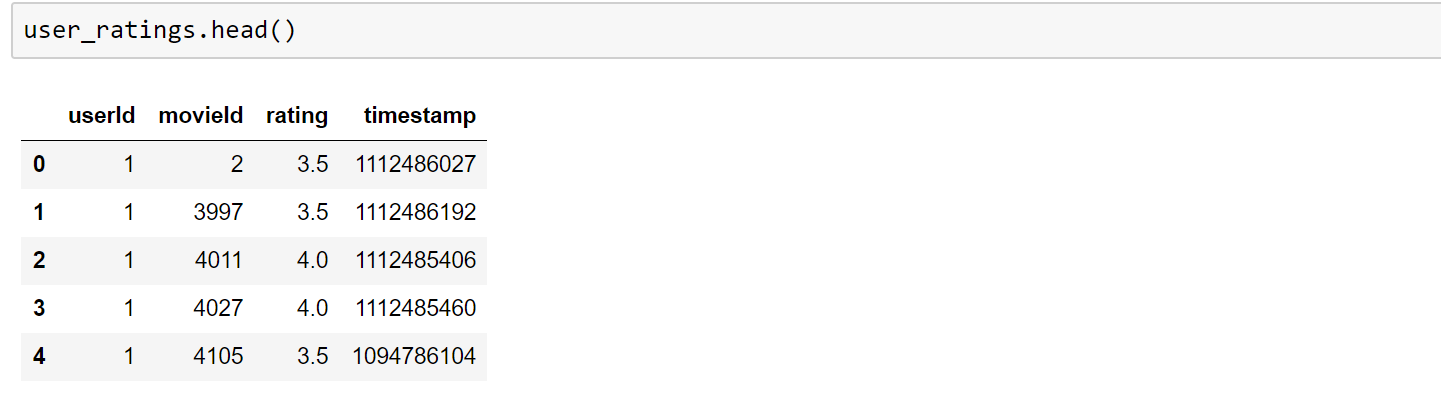
* Jupyter Notebook
* Supervised Machine Learning (KNN)
* Collaborative Filtering
* Natural language processing

**Dataset**

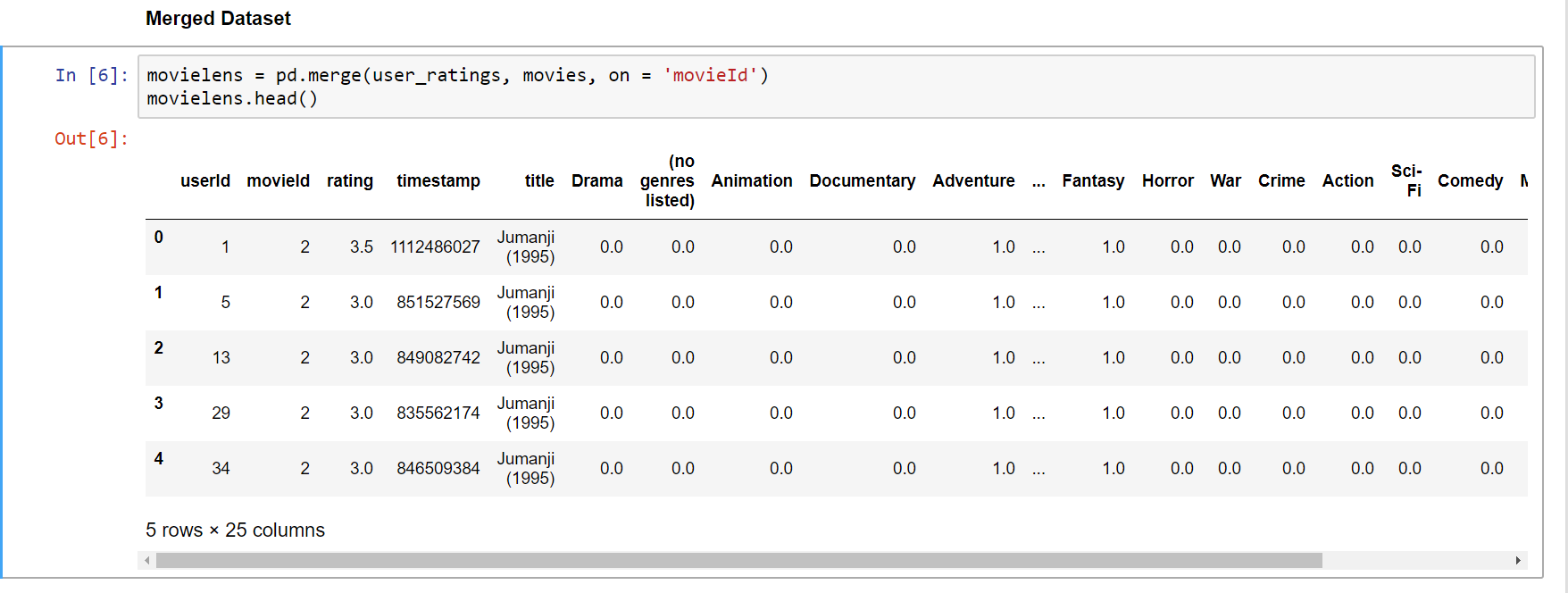
* **Movielens** – grouplens.org



Size – 27278 rows with 3 attributes namely movieId, title and genres. There are multiple genres for each movie. These genres are later on separated to form a one hot encoded matrix.

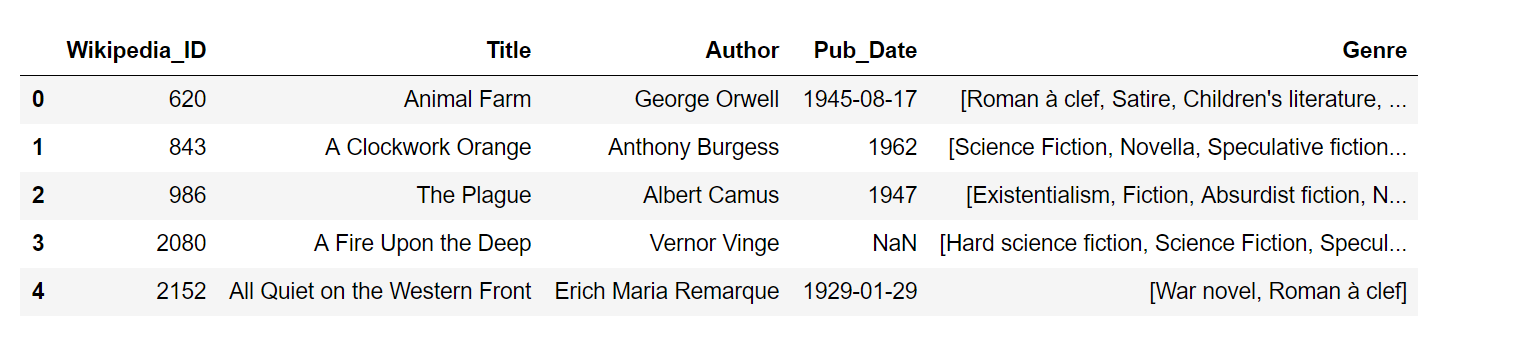


Size – 20 million rows with 4 attributes namely userId, movieId, rating, timestamp. Out of them only userId, movieId, and rating have been utilized.



Size – 20 million rows with 25 attributes which is formed by merging the movies and users dataset. The columns are the unique tags of movies.

* **Booksummaries** – cs.cmu.edu



Size – 16559 entries with 5 attributes namely Wikipedia\_ID, Title, Author, Pub\_Date, and Genre. Only the Title and Genre columns are further used for our analysis.

**Algorithm and Approach**

**Information collection phase**

This phase collects relevant information of users to generate a user profile or model for the prediction tasks including user’s attribute, behaviors or content of the resources the user accesses. A recommendation agent cannot function accurately until the user profile/model has been well constructed. The system needs to know as much as possible from the user in order to provide reasonable recommendation right from the onset. Recommender systems rely on different types of input such as the most convenient high quality explicit feedback, which includes explicit input by users regarding their interest in item or implicit feedback by inferring user preferences indirectly through observing user behavior. Hybrid feedback can also be obtained through the combination of both explicit and implicit feedback. In E-learning platform, a user profile is a collection of personal information associated with a specific user. This information includes cognitive skills, intellectual abilities, learning styles, interest, preferences and interaction with the system. The user profile is normally used to retrieve the needed information to build up a model of the user. Thus, a user profile describes a simple user model. The success of any recommendation system depends largely on its ability to represent user’s current interests. Accurate models are indispensable for obtaining relevant and accurate recommendations from any prediction techniques.

**Explicit feedback**

The system normally prompts the user through the system interface to provide ratings for items in order to construct and improve his model. The accuracy of recommendation depends on the quantity of ratings provided by the user. The only shortcoming of this method is, it requires effort from the users and also, users are not always ready to supply enough information. Despite the fact that explicit feedback requires more effort from user, it is still seen as providing more reliable data, since it does not involve extracting preferences from actions, and it also provides transparency into the ––recommendation process that results in a slightly higher perceived recommendation quality and more confidence in the recommendations.

**Implicit feedback**

The system automatically infers the user’s preferences by monitoring the different actions of users such as the history of purchases, navigation history, and time spent on some web pages, links followed by the user, content of e-mail and button clicks among others. Implicit feedback reduces the burden on users by inferring their user’s preferences from their behavior with the system. The method though does not require effort from the user, but it is less accurate. Also, it has also been argued that implicit preference data might in actuality be more objective, as there is no bias arising from users responding in a socially desirable way and there are no self-image issues or any need for maintaining an image for other.

**Hybrid feedback**

The strengths of both implicit and explicit feedback can be combined in a hybrid system in order to minimize their weaknesses and get a best performing system. This can be achieved by using an implicit data as a check on explicit rating or allowing the user to give explicit feedback only when he chooses to express explicit interest.

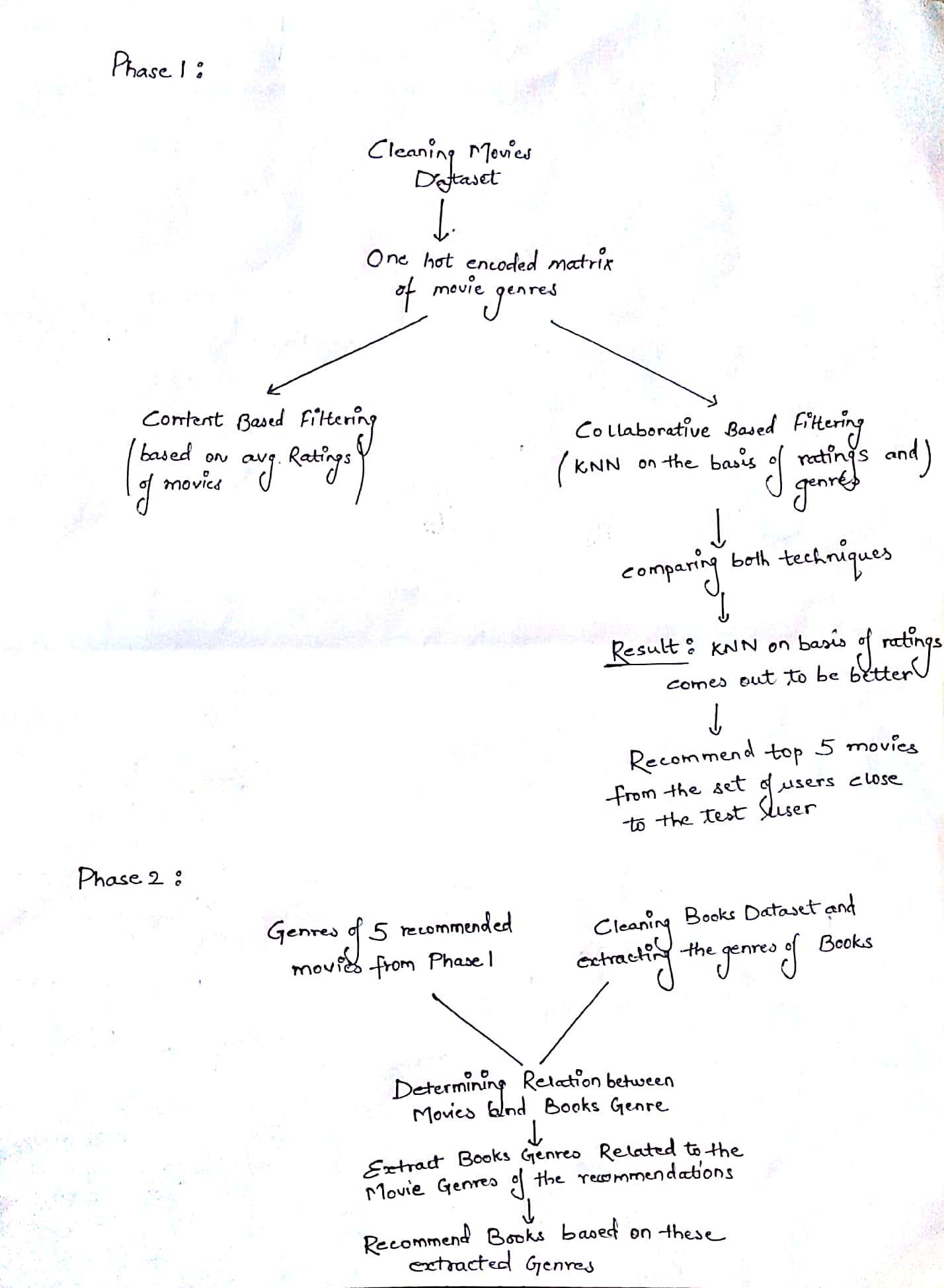
**Learning phase**

It applies a **Collaborative Filtering and** **KNN** algorithms to filter and exploit the user’s features from the feedback gathered in information collection phase.

**Prediction/recommendation phase**

We use the top rated movies out of the recommended movies for the test user to recommend the books. Then, we make a bucket of genres of these recommended movies. On the basis of this bucket we find the relation between these and the genres of the books, which in turn recommends user the books. The books can further be filtered on the basis of author, year of publication, rating, etc.

**Implemented Methodology**

****

**Phase 1**

When the user signs up on the website for the first time and we don’t know about his likings or disliking so to overcome this cold freeze problem we came up with a survey form in order to get all details about the user. Our form basically consists of two sections. in section 1 we are asking users about the liking on the basis of genres and in section 2 we get to know about the era in which they are interested.

There are basically 2 files in the movies dataset namely movies and ratings, the former one consists of the list of movies, their title and their genres while the former one consists of the ratings of the users given to different movies and their respective timestamp. For the preprocessing part we created a one hot encoded matrix with genres as columns values for different movie IDs. This would further be used in gathering information about the closeness of users.

There are three types of recommendation techniques namely content based, collaborative and hybrid filtering. The collaborative filtering is further divided into two techniques, model based and memory based. Model based comprises of different pre-defined model like clustering and association techniques while the memory based filtering technique is further classified into user based and item based. In our project we are using the User based technique.

For the top picks column of the website we are using content based filtering, where we calculated the mean ratings of every movie in the dataset and later printed the highest rated movies. Now for recommending movies according to the user’s preferences that are already stored in the dataset, we have used collaborative filtering using K nearest Neighbors technique which calculates the closeness of the test user to all the other train users and then selects the top closest users among them. This closeness score is basically the Euclidean distance between the ratings given by different users for the same movie. Say for user 161 the following 7 users have the least Euclidean score, we then create a full list of all the movies rated by these 7 users and extract a common list of movies which have already been watched by the test user and recommend the highest rated of the remaining movies to him.

Another way for calculating the closeness is based on Genres. Here firstly we grouped the data by user IDs so that we have the result as follows. Here as we can see that user 1 has watched 18 movies related to mystery genre 9 to war and so on. The distance or score between user 1 and user 2 is calculated as the square root of sum of squares of differences of the individual genres. the train to test ratio is 80:20

We then validated our results by using the standard KNN from the sklearn library, and observed that the root mean square error for our model was 0.115 which is in the acceptable range

Again we calculate the full list, the common list and the recommendations by taking genres as a factor in Euclidean score.

Now we compare both the techniques. The technique with the higher number of common movies gives us the better result because it shows that the train users are better correlated with the test users. The graph is the proof that the model with rating as a closeness factor gives us the better recommendation as their common movies are higher in number than other one.

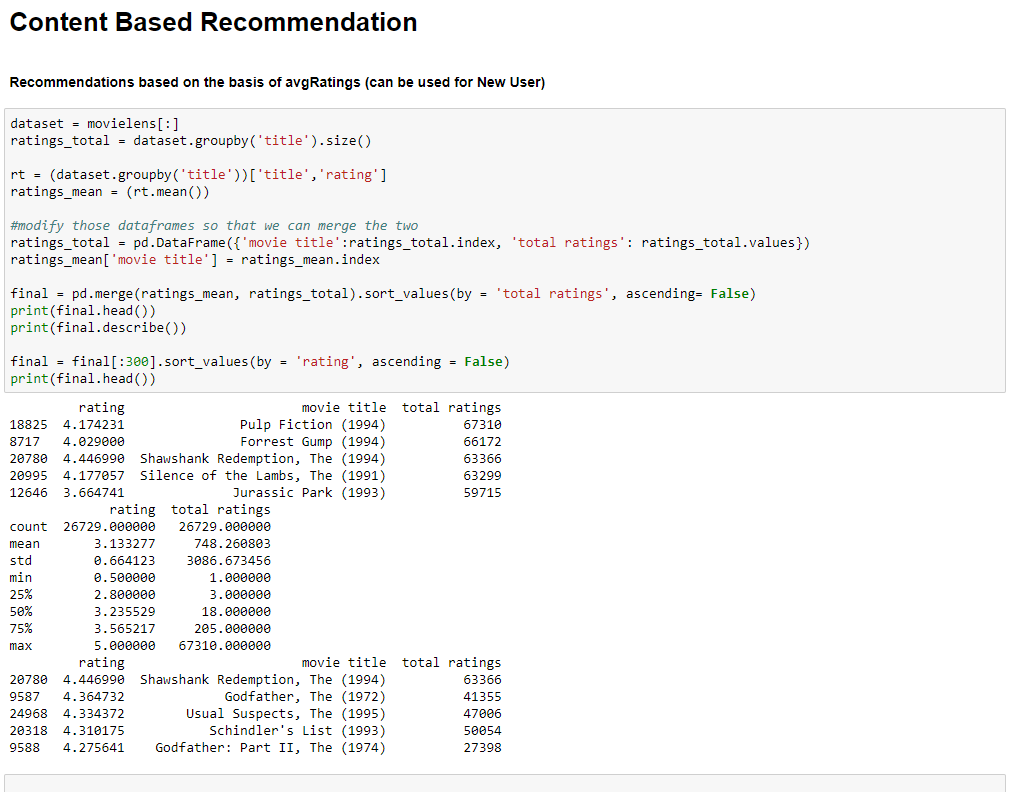
**Phase 2**

Now we recommend books, for that we firstly read the books dataset which has following attributes: Wiki id, title, author, publication date, genre

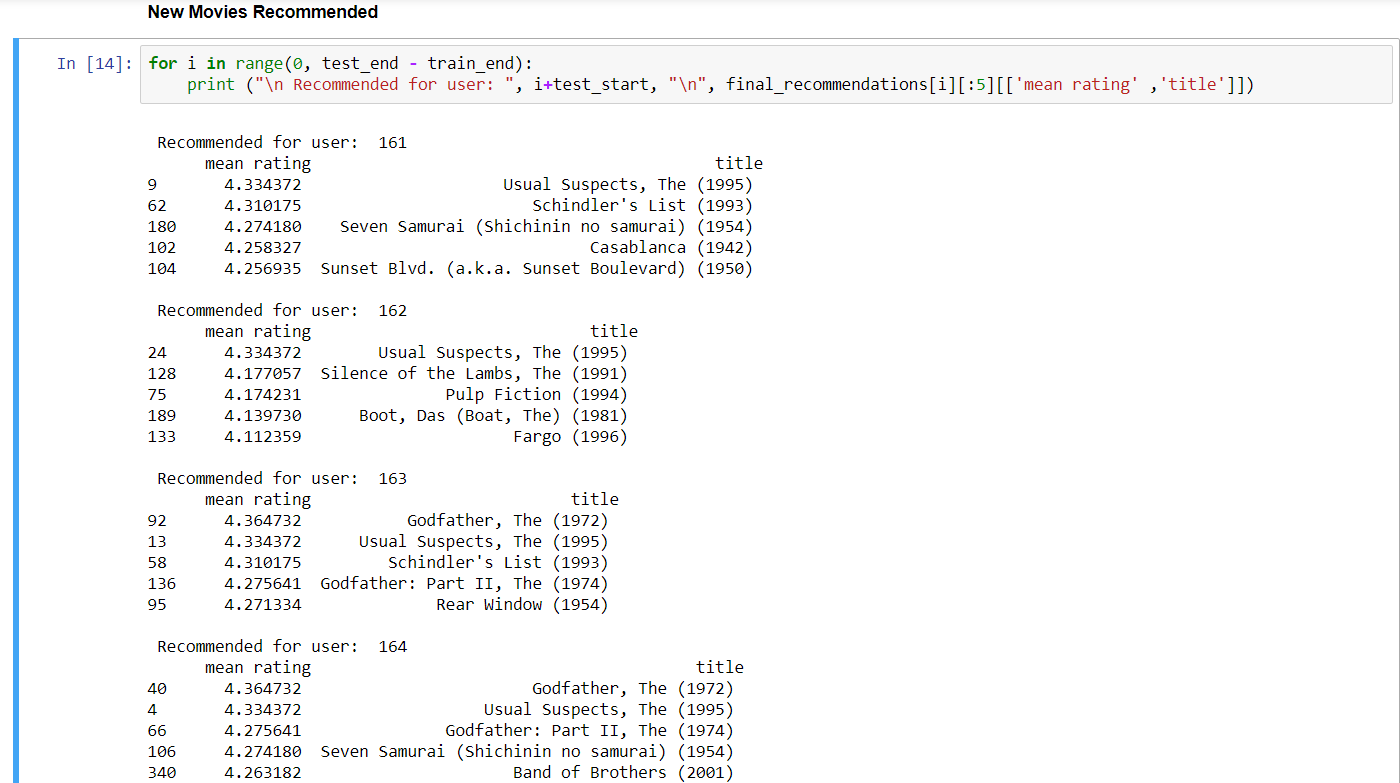
We calculate the total no of unique genres in the books dataset which comes out to be 227, and the unique genres in the movie dataset were 19. We now calculate the relationship between the books and the movies genres using NLP. Using this library, we calculate the synsets/synonyms of each genre in the movie set and calculate its similarity score with each genre in books set. The genre with highest score is returned as its relation.

We filter the final books recommendations by using these book genres, the top 5 books related to max number of these genres are then recommended.

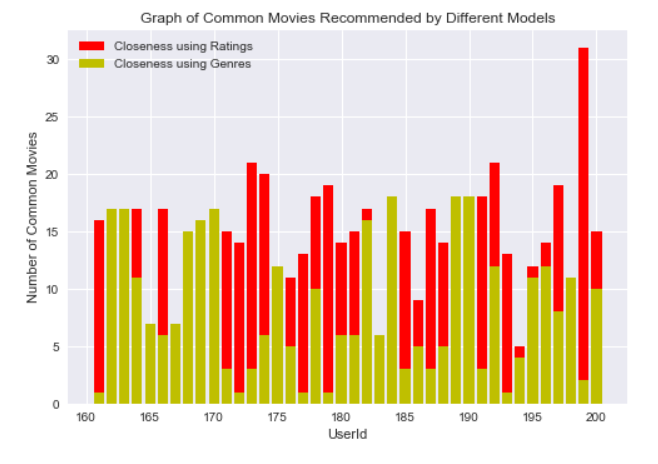
**Result**

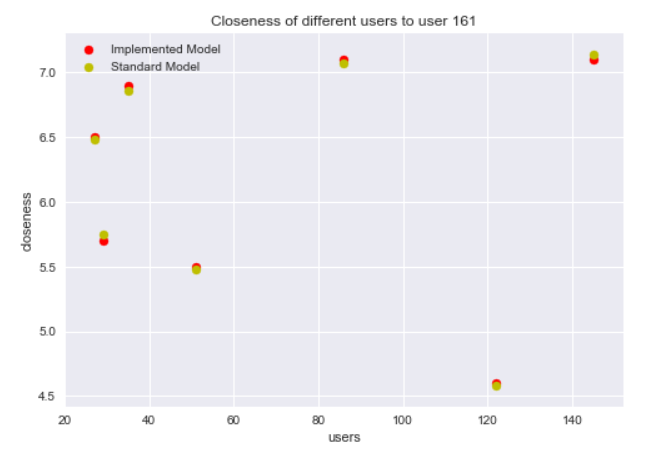


**Collaborative Based Recommendations**

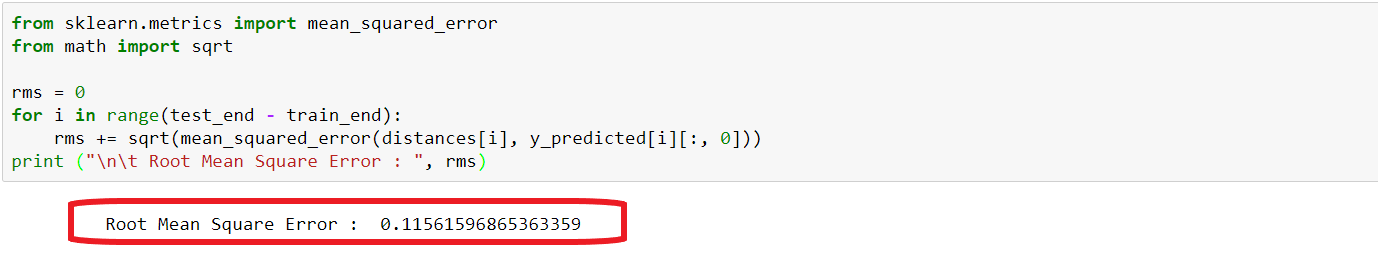


After using both the techniques i.e. Collaborative filtering over genres and Collaborative filtering over Movie Ratings we conclude that Collaborative filtering over movie ratings is a better approach.





The RMSE value of this technique on comparing with the standard tool comes out to be 0.115 (which is in the acceptable range)



**Future Scope**

Recommendation systems of the future will work in e-commerce to offer a more visceral, immersive and well-rounded experience for every step of a customer’s journey. One company that is already getting there is [Boomtrain](http://boomtrain.com/?utm_campaign=ecomm&utm_medium=mktg&utm_source=quora_recoeng).

Another industry booming up these days is of targeted advertisements where companies uses these techniques to show users only those sort of adds which they like and are then earning from it.

**Bibliography**

[1] J.A. Konstan, J. Riedl **Recommender systems: from algorithms to user experience** User Model User-Adapt Interact, 22 (2012), pp. 101-123

[2] C. Pan, W. Li **Research paper recommendation with topic analysis**

In Computer Design and Applications IEEE, 4 (2010), pp. V4-264

[3] Pu P, Chen L, Hu R. **A user-centric evaluation framework for recommender systems.**

In: Proceedings of the fifth ACM conference on Recommender Systems (RecSys’11), ACM, New York, NY, USA; 2011. p. 57–164.

[4] P. Resnick, H.R. Varian **Recommender system’s**, Commun ACM, 40 (3) (1997), pp. 56-58.

[5] A.M. Acilar, A. Arslan, **A collaborative filtering method based on Artificial Immune Network**, Exp Syst Appl, 36 (4) (2009), pp. 8324-8332

[6] L.S. Chen, F.H. Hsu, M.C. Chen, Y.C. Hsu **Developing recommender systems with the consideration of product profitability for sellers** , Int J Inform Sci, 178 (4) (2008), pp. 1032-1048

[7] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez **Recommender systems survey** Knowledge-Based Syst, 46 (2013), pp. 109-132

[8] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl **Evaluating collaborative filtering recommender systems** ACM Trans Inform Syst, 22 (1) (2004), pp. 5-53

[9] G. Adomavicius, A. Tuzhilin **Toward the next generation of recommender system. A survey of the state-of-the-art and possible extensions** IEEE Trans Knowl Data Eng, 17 (6) (2005), pp. 734-749