MOVIE TITLE RECOMMENDER SYSTEM

A Project Report

Submitted by

PALADUGULA SAI PRASANTH – 1700267C203
M V S S S CHAKRI KRISHNA -1700144C203
JAVVAJI AKHIL CHOWDARY – 1700216C203

Batch - 2017 CSE-1

Course
MACHINE LEARNING and DATA MINING - CSE2702

SCHOOL OF ENGINEERING & TECHNOLOGY
BML MUNJAL UNIVERSITY

April 2020

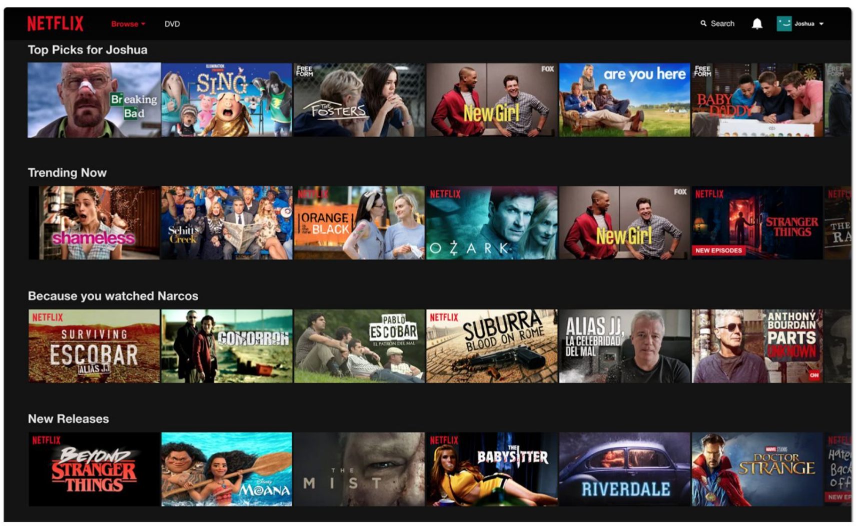
ACKNOWLEDGEMENT

The success and final results of this project required a few steering and help from many people and I am enormously privileged to have been given this all along the final touch of our challenge. All that we simply have finished is most effective because of such supervision and assistance and we might not neglect to thank them. It is always a massive pleasure to remind the fantastic people of our elegance for their sincere steerage I received to uphold my presentation similarly to helping us to assist beautify myself with my presentation competencies. First of all we would love to thank our father and mother for giving encouragement, enthusiasm and valuable assistance to us. Without all this, we would now not have been able to try this presentation well. Secondly, thanks to Mr.Atul Mishra for giving us the opportunity to undergo this shape of rigorous training. He moreover gave us a lot of guidance and help. Thirdly, we need to express our non-public thanks to the University officials to set the curriculum so dynamic which enables me to end up a higher competitor in this global making me a higher man or woman every day. A small sheet isn't enough for us to express the useful resource and steerage we have obtained from all of the above cited for almost all of the work I did.

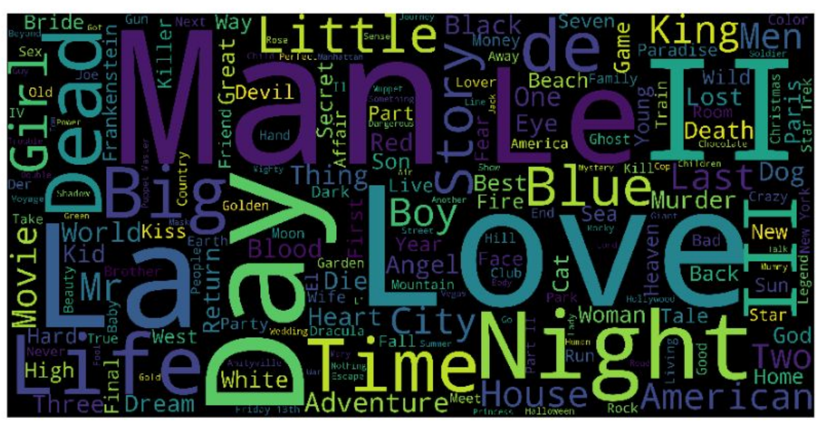
Movie Title Recommender System

Abstract –

A Movie Recommendation System filters the information with the help of various algorithms and recommends the most apt movies to users. It first captures the previous performance of the user and on the basis of that performance it recommends movies which the users may like to watch. If an absolutely new user visits an e-commerce website, that site won't have any previous history of that client. So how does the site approach prescribing items to the client in such a situation? One potential arrangement could be to suggest the top of the line items, for example the items which are high sought after. Other probable solution can be to recommend the products which would make them the at most profit to the business. Generally Three prime approaches are used for this recommendation systems. The first kind of Recommendation System is Content Based Recommendation System where we compute the movies that are similar to that movie and if the user watches that particular movie to which we calculated the similar movies, we will recommend the similar movies to him. The second type is Popularity Based Recommendation System in which we recommend movies based on the Popularity which means we use the data which gives us the number of people watched a particular movie and we will the collect the data from highest views to the lowest and we can recommend the users and obviously most of the recommended movies will be of highest popularity (Highest views). This type of Recommendation System is observed on YouTube. The third Recommendation system is Collaborative Based Recommendation System in which we find a similar user to a particular user and will recommend the movies to the user that his similar user watched and vice versa.



Genres:

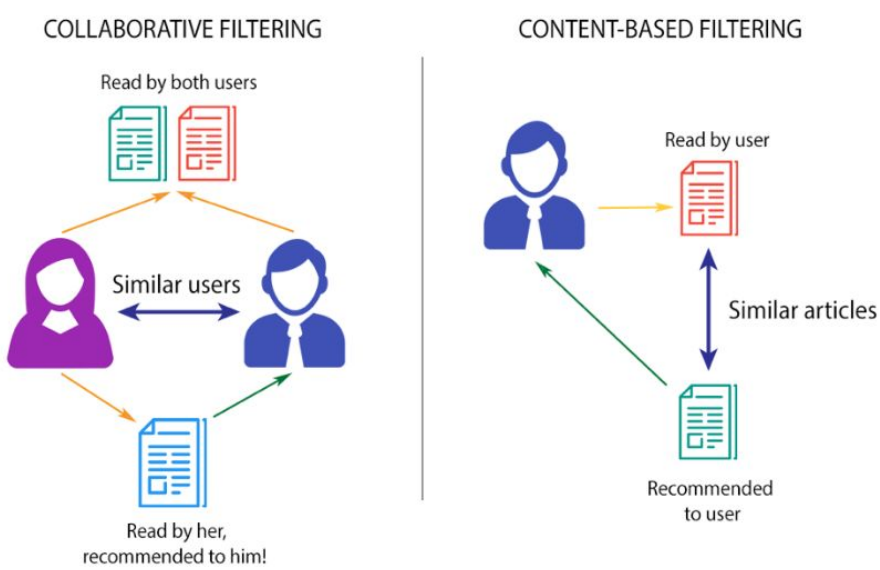


Movie Title Recommender System

I. INTRODUCTION

A Movie Recommendation System is a sort of facts filtering approach which attempts to calculate the alternatives of a consumer, and build indications based totally on those choices. Massive variety of recommendation systems are available. These have come to be progressively more present day extra than the last few years and are a gift applied in the largest part of online structures that we use. The content material cloth of such structures varies from films, tune, books and movies, to friends and tales on social media structures, to merchandise on E-trade internet sites, to population on professional and courting websites, to look for fallout lower back on Google. Frequently, the ones systems are successful to accumulate data approximately a customers alternatives, and might use this statistics to expand their guidelines in the future. For example, Facebook can supervise your verbal exchange with notable stories for your feed in order to look at what form of testimonies attraction to you. At times, the recommender structures can craft upgrades based totally on the actions of a big range of population.Due to the advances in recommender structures, clients frequently assume proper pointers. They have a small threshold for services that aren't capable of making suitable hints. If a song streaming app is not capable of calculating and playing a tune that the person likes, then the man or woman will just forestall the usage of it. This has caused an excessive significance with the useful resource of tech groups on civilizing their advice structures. However, the difficulty is greater elaborate than it appears. Each man or woman has various possibilities and likes. What's greater, even the revelation of a solitary client can range contingent upon a large number of variables, for instance, us of a mind, season, or form of movement the patron is doing. For instance, the sort of song one might probably want to listen while working closer to varies relatively from the type of track he'd music in to even as making ready supper. Another hassle that concept frameworks want to light up is the investigation rather than abuse problem. They should take a look at our new areas to locate step by step about the patron, whilst reaping rewards as plenty as viable from what is as of now belief about the client. Three crucial methodologies are implemented for our recommender frameworks. The first kind of Recommendation System is Content Based Recommendation System where we compute the similar movies to a movie and if the user watches that particular movie to which we calculated the similar movies, we will recommend the similar movies to him. The second type is Popularity Based Recommendation System in which we recommend movies based on the Popularity which means we use the data which gives us the number of people watched a particular movie and we will the collect the data from highest views to the lowest and we can recommend the users and obviously most of the recommended movies will be of highest popularity (Highest views). This type of Recommendation System is observed on YouTube. The third Recommendation system is Collaborative Based Recommendation System in which we find a similar user to a particular user and will recommend the movies to the user that his similar user watched and vice versa.

.



Collaborative Filtering vs Content-Based Filtering

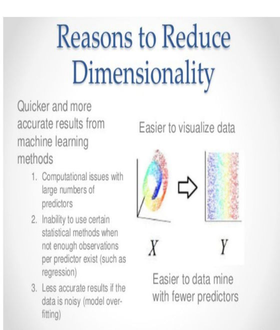
II. LITERATURE REVIEW

A movie recommendation system was proposed by DK Yadav and he uses collaborative filtering and records supplied by character.From the records collected, statistics are analyzed and a movie is normally suggested to the user that are organized with the film with maximum score first.

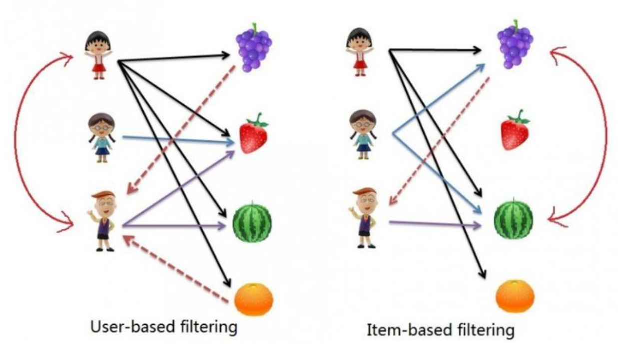
Luis M Capos has analyzed the conventional recommender structures which are collaborative filtering based recommendation systems & content based recommendation systems.Since every recommender has its own private defects he suggested a today's machine and this will be a mixture of both collaborative filtering & bayesian network.A hybrid device has been supplied with the resource of Harpreet Kaur. The gadget makes use of a combination of both content fabric further to collaborative filtering set of policies.The person - person relationship further to person - topic relation will act as a crucial position inside the recommendation system.The person unique facts or topic particular statistics is mixed or brought together to create a cluster with the aid of Utkarsh Gupta. The utilization of chameleon. This is an extraordinary technique based totally on the hierarchical clustering for the recommender system. For keeping a count on the rating of an item balloting tool is used which collects the votes from different users.This device which is proposed has less number of errors and does better clustering of similar items.Urszula Kużelewska says that clustering is a method to cope with recommender devices.There are 2 methods for the computation of clusters were furnished and checked thoroughly.The two collaborative filtering methods based on centroid and memory can be are used to check the efficiency the methods proposed. The prevent answer changed into an extensive growth in the efficiency of the movie recommendation systems at the identical time as compared to just centroid-based totally without a doubt technique.Costin-Gabriel Chiru suggested recommendation system , this is a device that uses the data of the customer to offer them film suggestions. The idea proposed will resolve the hassle of specific recommendations that are resulting by neglecting the data related to patrons. The mental profile of the person, the statistics of the genres the person is seeing and the data regarding film rankings from the internet is accumulated. These are based totally mostly on summations of similarities that are calculated. This is an advanced system which uses the recommender systems based on collaborative filtering & content filtering.For identifying the problem degree of every type by every trainee Hongli Lin recommended a technique known as the content boosted collaborative filtering (CBCF).The set of policies are splitted so that they become 2 types, 1st is the content based filtering that improves the score of the present trainees type facts and the 2nd is the collaborative filtering that offers the very last guess.

The CBCF set of rules includes the blessings of each CBF and CF, at the same time as at the identical time, overcoming each of their risks.

The collaborative filtering is a most common way for creating the recommendation systems which are intelligent and they give best output with more data collected from user.Many big mncs like netflix and amazon use this.There are two types one is User based while the other is Item based filtering.Both can be implemented again with the help of memory based and model based filtering.In memory based filtering we find the similarity by the help of ratings and reviews given by the users.The user based and item based are similar in terms of the concept but away too different in terms of mathematics.Amazon uses the item based filtering method.We use the K-Nearest Neighbours algorithm in memory based filtering which is almost similar to the cosine similarity.Here K is the number of neighbours from which we are taking the vote to confirm the confidence interval.So, K value is very essential here and we need to select it very carefully.We calculate the distances between test data and every row of training cases with the help of distance metrics like euclidean distance.After getting all the distances we arrange them in ascending order and take the k number of rows from them.Now take the most frequently occuring classes of these rows.The model based filtering has dimensionality reduction and matrix reduction algorithms which are helpful in the reduction.Here in dimensionality reduction we reduce the large matrix of user and item and this done by the help of matrix factorization.In matrix factorization we reduce a big matrix into smaller matrices which are products of each other with same dimensions.The reduced matrix here is the user and item matrix.After the reduction,rows of both matrices represent the user and item while columns represent their features or characteristics.We use the singular value decomposition (SVD) algorithm which is a famous algorithm to implement matrix factorization and this was found in a coding competition conducted by the Netflix.There are many other algorithms like PCA along with the different variations of it,NMF etc to do this matrix factorization.If we are using the neural networks for dimensionality reduction then autoencoders are very useful.



3 Reasons to Reduce Data's Dimensionality



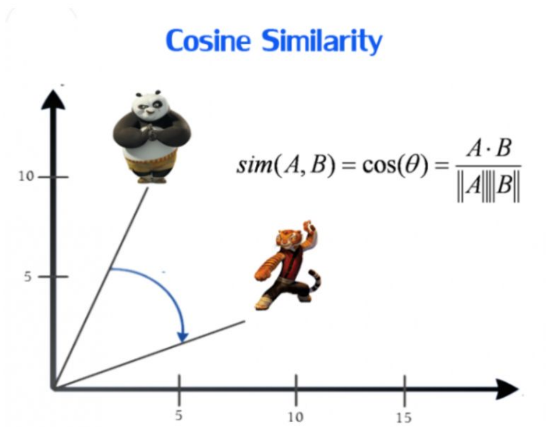
Movie Title Recommender System

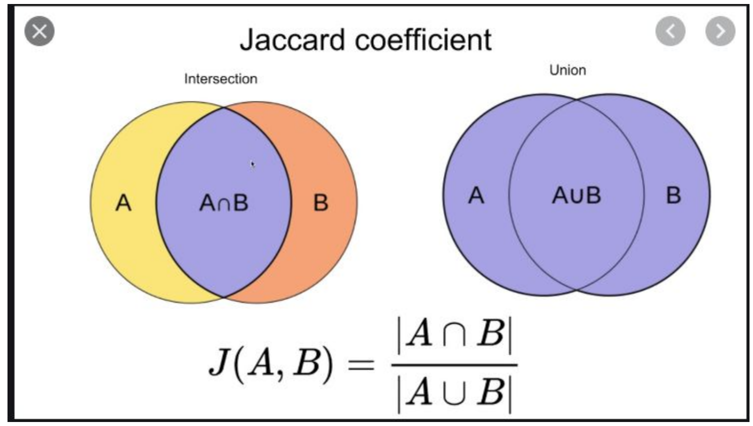
Suggestion frameworks are a settled field of study. The precision and significance of suggestions rely upon essential info information, which is the past conduct of clients and their companions. Past conduct, from the point of view of area and agenda proposal, could be area history. Clients' verifiable conduct can be profitably concentrated from their actions in informal communities. Area-based informal organizations (LBSNs) are an uncommon class of interpersonal organizations that permits clients to register in different areas. These systems, making geo-labeled or area labeled information, are in any case viewed as area-based social networks. There has been, as of late, a sensational climb in the amounts of area information data straightforwardly accessible on the web with the advancement of person to person communication locales, for example, Facebook Places, Google Plus, Google Latitude, Gowalla, Brightkite, Twitter, Flickr, Foursquare, and such like. Administrations reached out by LBSNs in this setting permit clients to share data as content, picture, and video. Despite imperatives on the accessibility and utilization of area data as a result of safety efforts, there are huge quantities of clients who effortlessly outfit individual data in geo-labeled information. A considerable lot of these LBSNs permit wannabes to accumulate client information by different methods. Some of them give Application Program Interface (API) for get-together the information, while some others permit the clients to share their information with other long range interpersonal communication locales like Twitter or Facebook so others can see them or concentrate them absent a lot of imperatives.

III. METHODOLOGY

The methodology describes the way we implement the project. So now, I'm mentioning the path that we have traversed in order to complete this project. As we already mentioned that we are going to implement a Content Based Recommendation System plus Collaborative Based Recommendation System from which we also can interpret the comparisons between these systems not only theoretically that we already know but also practically. Whenever we write a code , we start with importing required libraries through which we are going to work. Same thing I've done here. I imported pandas, numpy, and random. I also imported count vectorizer and cosine similarity from sk learn. After importing the required libraries, we will have to read the dataset. After reading the dataset, we can observe that the data set contains so many attributes in which some of them can be excluded when recommending movies. So in my dataset, I am going to consider a few attributes which are director, cast, genres, keywords. Considering these attributes we are going to find some similar movies to particular movies and if a particular user watches that particular movie to which we have obtained similar movies , we are going to recommend the similar movies to him since he watched that movie to which these movies are similar. Data preprocessing is done as we will replace the missing values with NaN. As I already mentioned, the dataset contains various attributes but here we are concerned about only some of them mentioned above. So for that we are combining these 4 attributes that we considered to make them look together for further proceedings. For creating a combined string for these 4 attributes we are going to create a function for that. Now from the text that we have from these 4 attributes, we have to transform them into a vector and for that we are going to use count_matrix function which is a part of the sklearn library. Now when we calculate the cosine similarity for this matrix or vector obtained we can observe the similarity scores for the movies. In the matrix we have movies both in row and column and the scores are allocated to the movies in the row corresponding to those movies in the column and if there is the same movie in both row and column intersecting, then the similarity score will be 1. We use auxiliary function to fetch the title of the movie and index of it using index and title auxiliary function. Now in order to find the similar movies to a particular movie, we have to fetch a movie that the user likes or watches and will have to see the similar movies to that movie and we are going to recommend the similar movies to the movie that he liked or watched.

Coming to a similarly calculating method, if User A watches a certain number of movies and User B watches some certain number of movies. Here we calculate the similarity between the users by considering the intersection , that is their common movies and the union, that is total movies watched by both. We calculate the cardinality for both union and intersection and we divide the cardinality if intersection with cardinality of union to get a score in between 0 and 1 which is referred to as similarity score and the name of this method of finding similarity is Jaccard Similarity. We can calculate the cardinality by importing math functions beforehand. By using this Jaccard Similarity, when there are some n number of users and if we wanted to recommend a movie to a particular user, then we calculate the similarity score for this particular user with all the users and find the highest similarity score which indicates that these two users are similar and then from that similar user found, we remove the common movies watched (union) and then recommend the remaining movies.

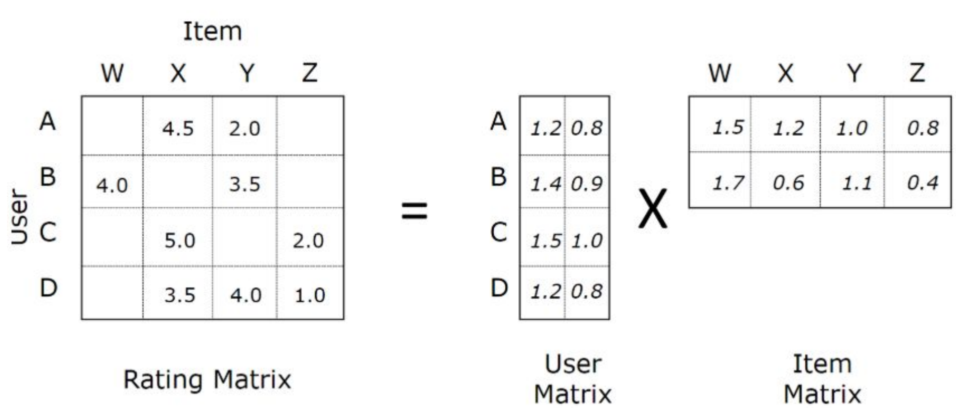




After Reading the dataset, Here are the average RMSE, MAE and total execution time of various algorithms (with their default parameters) on a 3-fold cross-validation procedure. We will use RMSE as our accuracy metric for the predictions.We will be comparing SVD, NMF, Normal Predictor, KNN Basic and will be using the one which will have least RMSE value.Some understanding on the algorithms before we start applying.

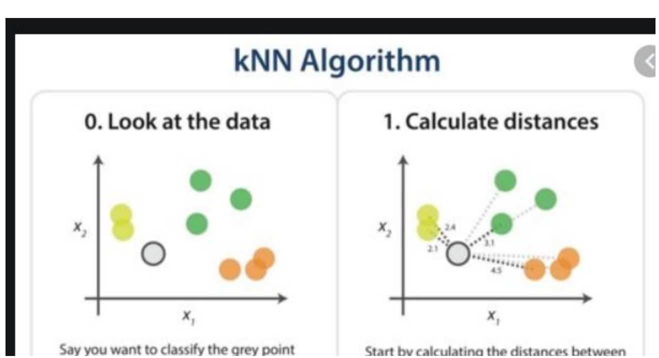
1: Normal Predictor: It predicts randomly and assumed to be normal. It's a basic algorithm that does not do much work but that is still useful for comparing accuracies.

2: SVD: It got popularized during the Netflix prize and is a Matrix Factorized algorithm. If baselines are not used, it is equivalent to PMF.



3: NMF: It is based on Non-negative matrix factorization and is similar to SVD.

4: KNN Basic: This is a ALgo used for Collaborative Filtering



Algorithm	test_rmse	fit_time	test_time
SVD	0.880220	4.403916	0.223411
NMF	0.934801	4.789203	0.184360
KNNBasic	0.958908	0.084770	2.151071
NormalPredictor	1.422818	0.107039	0.244199

Now coming to the Training & Testing part: We use the train_test_split() to sample a train set and a testset with given sizes, and use the accuracy metric of rmse. We'll then use the fit() method which will train the algorithm on the train set, and the test() method which will return the predictions made from the testset. Let's check how good or bad our predictions are: The following function will create a pandas data frame which will consist of these columns:

UID: user-id

iid: item id

Rui: the rating given by the user

est: rating estimated by the model

Iu: No of items rated by the user

UI: number of users that have rated this item

err: abs difference between predicted rating and the actual rating.

We are now going to show best and worst predictions.

best_predictions

	uid	iid	rui	est	details	Iu	Ui	err
1325	475	1210	5.0	5.0	{'was_impossible': False}	113	146	0.0
18982	25	5952	5.0	5.0	{'was_impossible': False}	17	131	0.0
7043	348	50	5.0	5.0	{'was_impossible': False}	40	156	0.0
4963	452	1221	5.0	5.0	{'was_impossible': False}	142	94	0.0
3946	597	260	5.0	5.0	{'was_impossible': False}	335	188	0.0
14154	122	1136	5.0	5.0	{'was_impossible': False}	222	107	0.0
23654	30	1198	5.0	5.0	{'was_impossible': False}	27	150	0.0
10666	106	318	5.0	5.0	{'was_impossible': False}	26	233	0.0
22458	122	608	5.0	5.0	{'was_impossible': False}	222	133	0.0
9136	452	1089	5.0	5.0	{'was_impossible': False}	142	96	0.0

worst_predictions

	uid	iid	rui	est	details	Iu	Ui	err
5754	182	7153	1.0	4.621940	{'was_impossible': False}	692	143	3.621940
19010	111	593	0.5	4.132474	{'was_impossible': False}	498	205	3.632474
20428	154	86644	0.5	4.200164	{'was_impossible': False}	26	6	3.700164
19876	542	1732	0.5	4.260783	{'was_impossible': False}	83	82	3.760783
19490	426	47	0.5	4.284836	{'was_impossible': False}	53	148	3.784836
11965	580	1250	0.5	4.298141	{'was_impossible': False}	332	33	3.798141
13456	580	1207	0.5	4.301877	{'was_impossible': False}	332	41	3.801877
19979	105	4027	0.5	4.357468	{'was_impossible': False}	555	74	3.857468
20582	256	5618	0.5	4.548539	{'was_impossible': False}	124	69	4.048539
13965	543	89904	0.5	4.954417	{'was_impossible': False}	55	7	4.454417

The worst predictions look pretty surprising. Let's look in more detail at item "3996", rated 0.5, our SVD algorithm predicts 4.4. It turns out, most of the ratings that Item received were between "3 and 5",only 1% of the users rated "0.5" and one "2.5"below 3. It seems that for each prediction, the users are some kind of outlier and the item has been rated very less number of times. K Recommendations. Recall and precision at K :Recall and precision are the classical evaluation metric and are used to evaluate the binary metric and so we have to convert our rating which is scaled from (1-5) into a binary problem relevant and not relevant items. Conversion to binary. To do the translation we have to select an arbitrary value on which we can say any rating above that will be considered relevant. There are many methods on selecting that value but for now, we will select 3.5 as the threshold, which means any true rating above 3.5 will be considered relevant and below will be not relevant.

Deciding 'k' : In recommendation systems, we are interested in showing the top N items to users and so the best is to compute precision and recall on top N values instead of calculating on all the items. Definition of Relevant and Recommended

Relevant: True Rating > = 3.5

Irrelevant: True Rating < 3.5

Recommended item: Predicted Rating > = 3.5

Not Recommended item: Predicted Rating > = 3.5

Definition of Precision and Recall

Precision: It tries to answer "What proportion of positive identifications was actually correct?"

Recall: It tries to answer "What proportion of actual positives were identified correctly?"

f1 score = 2*P*R/(P+R) where P is Precision and R is Recall

The below function computes precision and recall and F1 score as explained above

	threshold	tp	fp	tn	fn	Precision	Recall	F1
0	0.0	25209	0	0	0	1.000000	1.000000	1.000000
1	0.5	25209	0	0	0	1.000000	1.000000	1.000000
2	1.0	24861	348	0	0	0.986195	1.000000	0.993050
3	1.5	24149	1042	10	8	0.958636	0.999669	0.978723
4	2.0	23586	1406	115	102	0.943742	0.995694	0.969022
5	2.5	21138	2764	719	588	0.884361	0.972936	0.926536
6	3.0	18003	2510	2361	2335	0.877639	0.885190	0.881398
7	3.5	10532	2687	7149	4841	0.796732	0.685097	0.736710
8	4.0	4017	936	12271	7985	0.811024	0.334694	0.473843
9	4.5	580	261	19598	4770	0.689655	0.108411	0.187369
10	5.0	27	5	22016	3161	0.843750	0.008469	0.016770

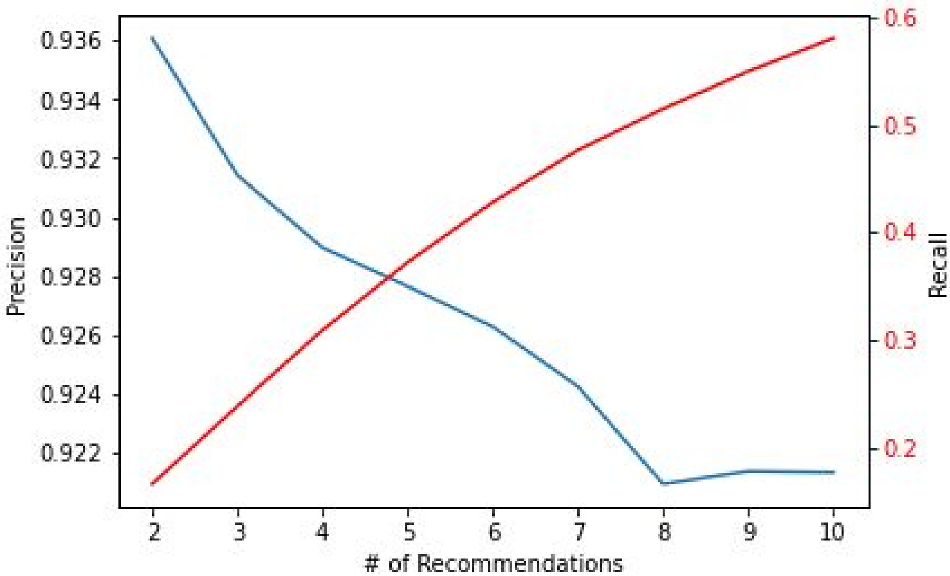
As per the results above, the optimal value for threshold is 2.5. The next step is to find the optimal K value, and to find it we have to first calculate precision and recall for all the K values(2-10) having threshold value 2.5. Below is the function to calculate precision and recall @ K. As the graph states, Precision drops significantly when K=4.So we will consider the value of K to be 4. Now as we know the optimal number of recommendations to provide, it's time to give recommendations to users. To do so we have to predict ratings for the movies which user has not yet watched. Here we will be using the build_anti_testset() method to get the data for the testset as we have to predict ratings for the (user, item) pairs which are not present. Now we have to sort all the predictions made. As we have all the predicted ratings, We'll subset to only top K movies for every user, where K is 4. Now we have a dataframe which consists of top 4 movies recommended to every user. Let's try one example and find recommendations for user 67. Here in the output we got the ratings but we need names of the movies, so now let us extract the names. Now as we have the movie-id's to be recommended, Let's find out the movie details of those id's by reading the movie data. Let's check the user history to see whether the given recommendations are similar or not. Now we can compare the results with the user history and see how relevant the recommendations are. Above is the user history and below the recommended movies. As the history of the user tells that the user mostly likes movies that are mixed that means he prefers crime,thriller,drama,comedy and we are recommending movies that are crime,thriller,drama because those are highly preferred by the user, which means we are recommending the right movies to the user.

IV. PERFORMANCE

At every Threshold level, we computed Recall, Precision, and F1 Score and from the outputs of threshold values ranging from 0 to 5, we found optimal threshold value to be 2.5 and then computed the recall precision for all the K values having threshold value as 2.5.

From the output, we see that Precision drops significantly when K=4.So we will consider the value of K to be 4. Also, we compared user history with the recommendation made and we observed a lot of similarities which signifies our prediction is accurate.

Precision and Recall graph:




```
The Jaccard similarity performance:
Jaccard Similarity btw u1 and u2 : 0.10437912417516497
Jaccard Similarity btw u1 and u3 : 0.12725399278722307
Jaccard Similarity btw u1 and u4 : 0.1276005547850208
Jaccard Similarity btw u1 and u5 : 0.10764705882352942
```

VIII. RESULT

User history:

Movie Title Recommender System

	movieId	title	genres
46	50	Usual Suspects, The (1995)	Crime|Mystery|Thriller
841	1104	Streetcar Named Desire, A (1951)	Drama
1616	2160	Rosemary's Baby (1968)	Drama|Horror|Thriller
2462	3275	Boondock Saints, The (2000)	Action|Crime|Drama|Thriller

Recommended Movies:

	movieId	title	genres
254	293	Léon: The Professional (a.k.a. The Professiona...	Action|Crime|Drama|Thriller
257	296	Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller
277	318	Shawshank Redemption, The (1994)	Crime|Drama
314	356	Forrest Gump (1994)	Comedy|Drama|Romance|War

Using Jaccard Similarity:

```
Top 5 similar movies to Spectre are:
Deadfall
The Devil's Own
AWOL-72
Jack Reacher
The One
Lethal Weapon 3
Random selected title for u1 : ['Elizabeth: The Golden Age', 'Tank Girl', 'The Barbarians', '[Rec]', 'Operation Chromite', 'The Lovely Bones', 'Roadside Romeo']
Random selected title for u2 : ['The Last Exorcism', 'Duplicity', 'Warriors of Virtue', 'Aloha', 'The Lovers', 'Street Kings']
Random selected title for u3 : ['Deceptive Practice: The Mysteries and Mentors of Ricky Jay', 'Flightplan', 'Disturbia', 'Narc', 'Insidious: Chapter 2']
Random selected title for u4 : ['Superman', 'Alice Through the Looking Glass', 'Snow Angels', 'Tracker', 'A Guy Thing', 'Dazed and Confused']
Random selected title for u5 : ['Elizabethtown', 'DragonHeart', 'The Messenger: The Story of Joan of Arc', 'Antibirth', 'Stripes']
The highest similarity is between u1 and u2
Recommended movies for u1 on basis of similarity score between u1 and u2 are :
['Operation Chromite', '[Rec]', 'Roadside Romeo', 'Tank Girl', 'The Lovely Bones', 'Elizabeth: The Golden Age', 'The Barbarians']
```

IX. CONCLUSION AND FUTURE SCOPE

We have combined the Content Based Recommendation System and Collaborative Based Recommendation System to implement our project. We believe that each algorithm has its own kind of limitations and this combination may reduce the limitations that these are facing individually. Similarity methods were used to make a better recommendation system increasing the accuracy and precision. In future , we can develop this by adding clustering and some other featured algorithms of similarity for better performance. Our methodology can be further extended to other domains like to recommend feed, songs, news, books, e-commerce sites,etc…

REFERENCES

1) https://realpython.com/build-recommendation-engine-collaborative-filtering/#algorithms-based-on-k-nearest-neighbours-k-nn
2) https://www.researchgate.net/publication/324763207_A_Hybrid_Approach_using_Collaborative_filtering_and_Content_based_Filtering_for_Recommender_System
3) https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243
4) https://surprise.readthedocs.io/en/stable/FAQ.html
5) https://surprise.readthedocs.io/en/stable/dataset.html
6) https://towardsdatascience.com/paper-summary-matrix-factorization-techniques-for-recommender-systems-82d1a7ace74
7) https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html
8) https://www.themarketingtechnologist.co/a-recommendation-system-for-blogs-content-based-similarity-part-2/
9) https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1
10) https://medium.com/@connectwithghosh/simple-matrix-factorization-example-on-the-movielens-dataset-using-pyspark-9b7e3f567536
11) https://medium.com/@m_n_malaeb/recall-and-precision-at-k-for-recommender-systems-618483226c54
12) https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c
13) https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall?hl=ru
14) https://surprise.readthedocs.io/en/stable/matrix_factorization.html#surprise.prediction_algorithms.matrix_factorization.SVDpp
15) https://realpython.com/build-recommendation-engine-collaborative-filtering/