ECE219 Project 3: Collaborative Filtering

Firnaz Ahamed (104943091) Larsan Aro Brian Arockiam(904943186)
Hariharan Shamugavadivel(804946029) Shoban Narayan Ramesh(604741670)

March 31, 2018

1 Introduction

The usage of Internet is at its highest right now. Almost all everyday tasks are either directly or indirectly impacted by content on the Internet. But not all content would be suitable for all types of audience and this has led to the rapid development of Recommendation Systems and Algorithms. The Recommendation Systems thrive on the feedback that users give. This feedback is acquired even without the user knowing it based on his search trails and his likes and dislikes. In this project we are building a recommendation system based on collaborative filtering models which uses the collaborative powers of the ratings provided my multiple users to provide recommendations.

2 Dataset

The dataset which was used for this project was the 'MovieLens' dataset. The dataset has various attributes such as ratings and genres, but in this project we will be using only the ratings to build the recommendation system. The movie ratings of various users are available in the dataset which will be used to build the recommender.

3 Exploratory Data Analysis (Extra)

3.1 Distribution of average ratings

First, lets take a look at how the ratings are distributed for each movie, by looking at the histogram of the average ratings attained by each movie.

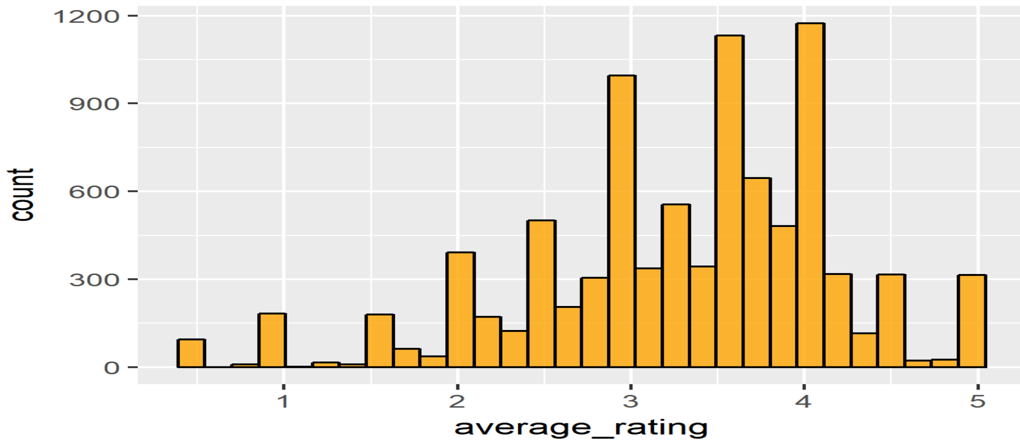

Figure 1: Histogram of average ratings received by movies

The histogram shows that most of the movies have an average rating between 3 and 4. There are around 300 movies which have an average rating of 5, but its highly unlikely that those movies have gotten ratings from many users. Its usually movies with ratings from a small number of users which have the perfect average rating value of 5. True to our intuition, if you look at the maximum number of ratings for the movie with an average rating of 5, it turns out to be just 4 user ratings.

3.2 Highest rated and most viewed movies

Lets look at some statistics about the best movies the most viewed movies and the highest rated movies. Since the average ratings parameter may be skewed, lets put a threshold of at least 50 ratings, in the search for highest rated movies.

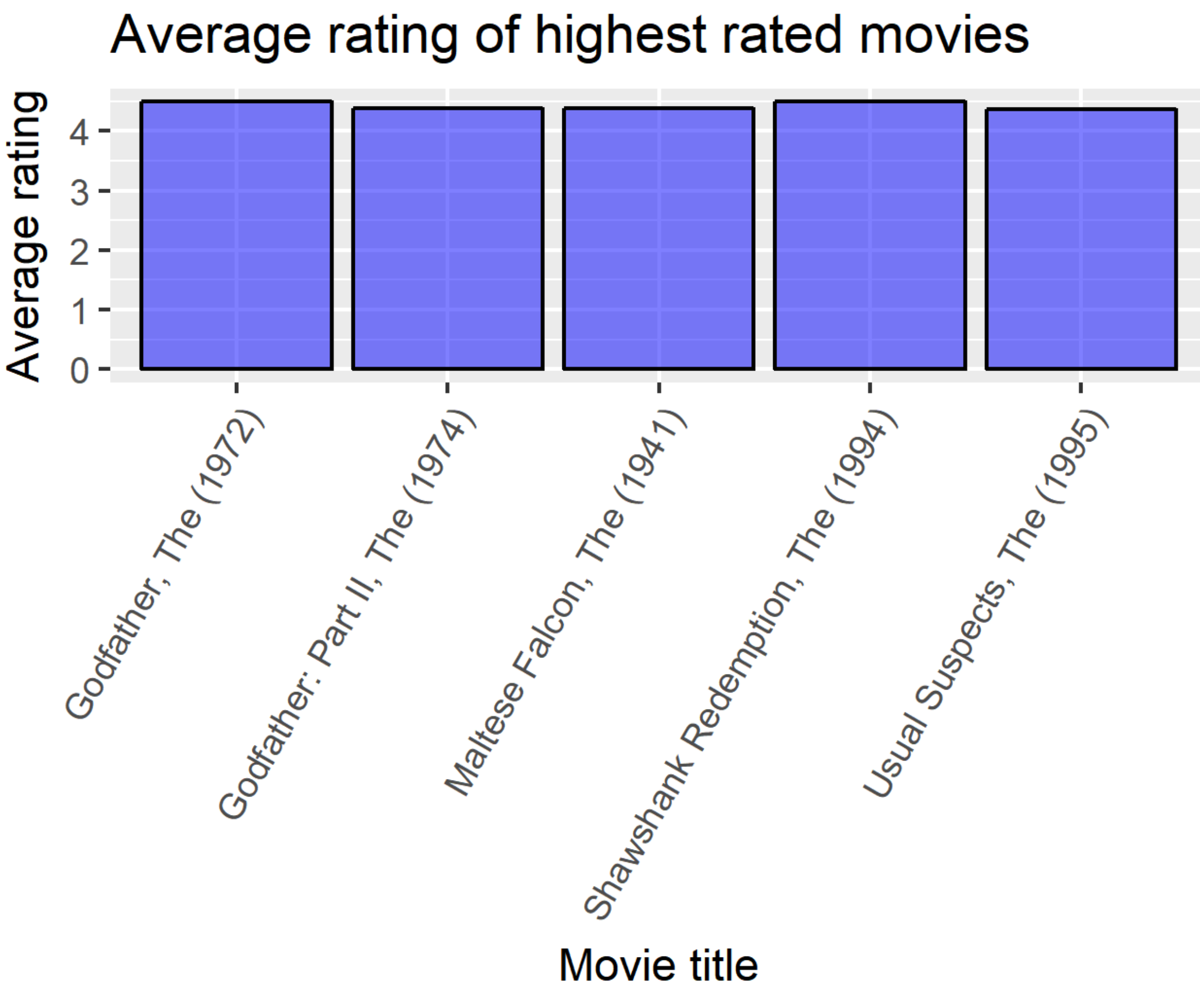

Figure 2: Average ratings of the highest rated movies

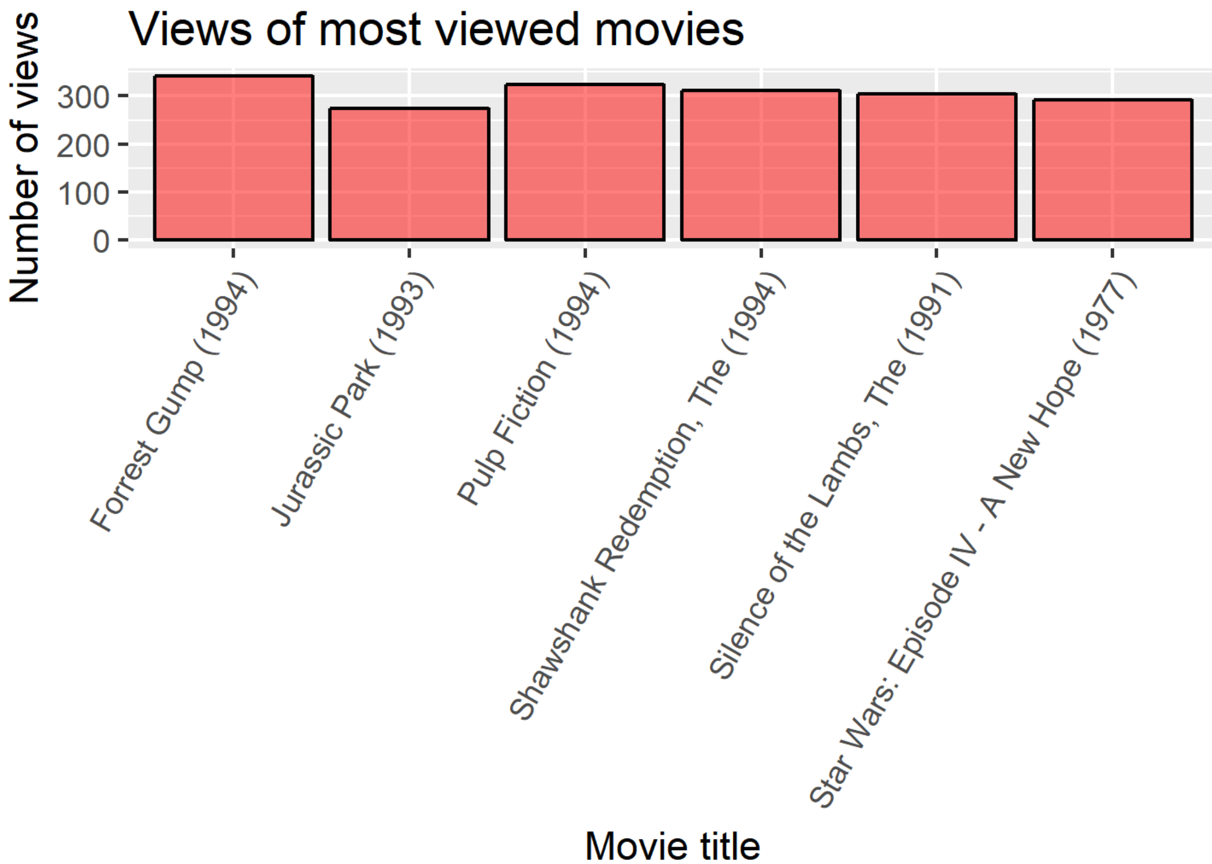

Figure 3: Number of views of the most viewed movies

4 Analyzing the Dataset

4.1 Question 1: Sparsity

The sparsity of the dataset is defined as the ratio of number of available ratings to the number of possible ratings.

$$\text{Sparsity}=0.016439$$

4.2 Question 2: Histogram of the Rating Values

The Histogram shows the frequency of the rating values. The shape of the histogram suggests that most(almost 80%) movies are rated between 3 and 5. A very few movies are rated very low. A large number of movies have been rated 4.

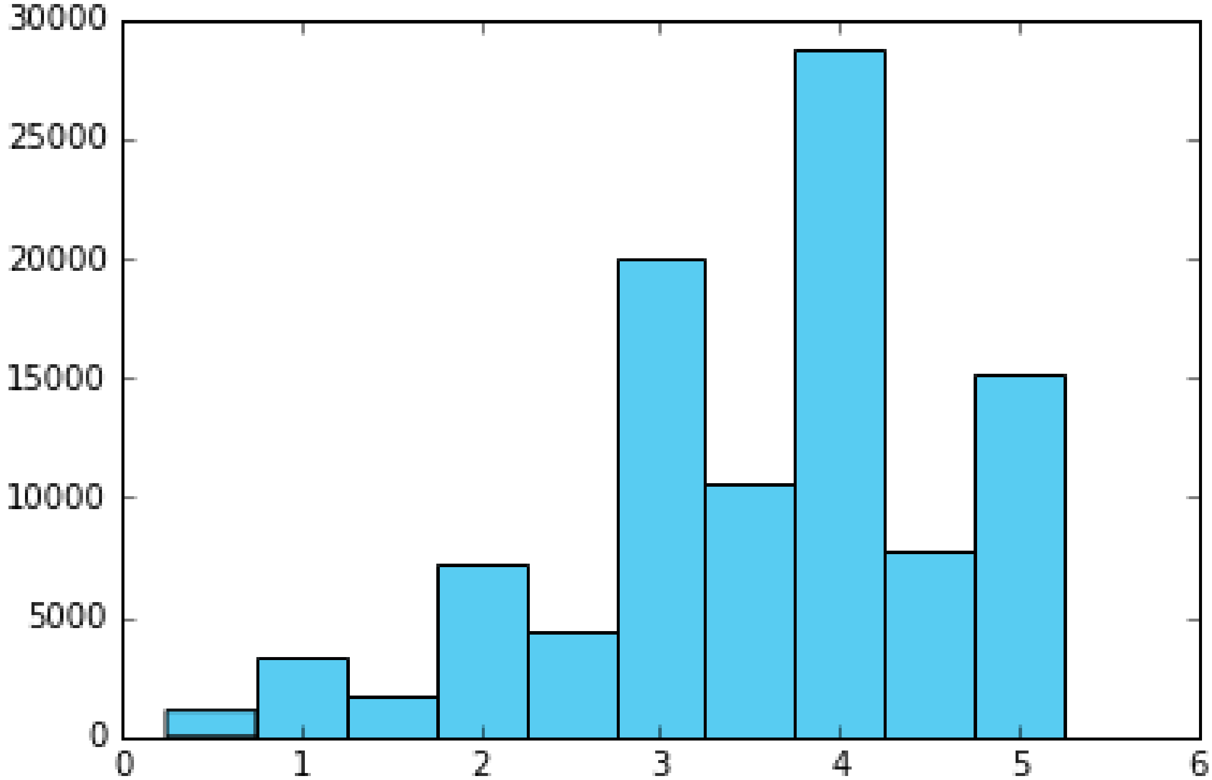

Figure 4: Histogram of Number of Movies Vs Ratings

4.3 Question 3: Distribution of Ratings among movies

The plot shows the number of ratings received by each movie vs the movie index. The x-axis is arranged with decreasing number of ratings.
From the plot it is seen that a very few movies(popular movies) tend to receive a large number of ratings (be it good or bad) and there is a rapid decrease in the number of ratings for less popular movies. More than half of the movies received less than 5 ratings. This is also a good indication of the sparsity of the ratings matrix.

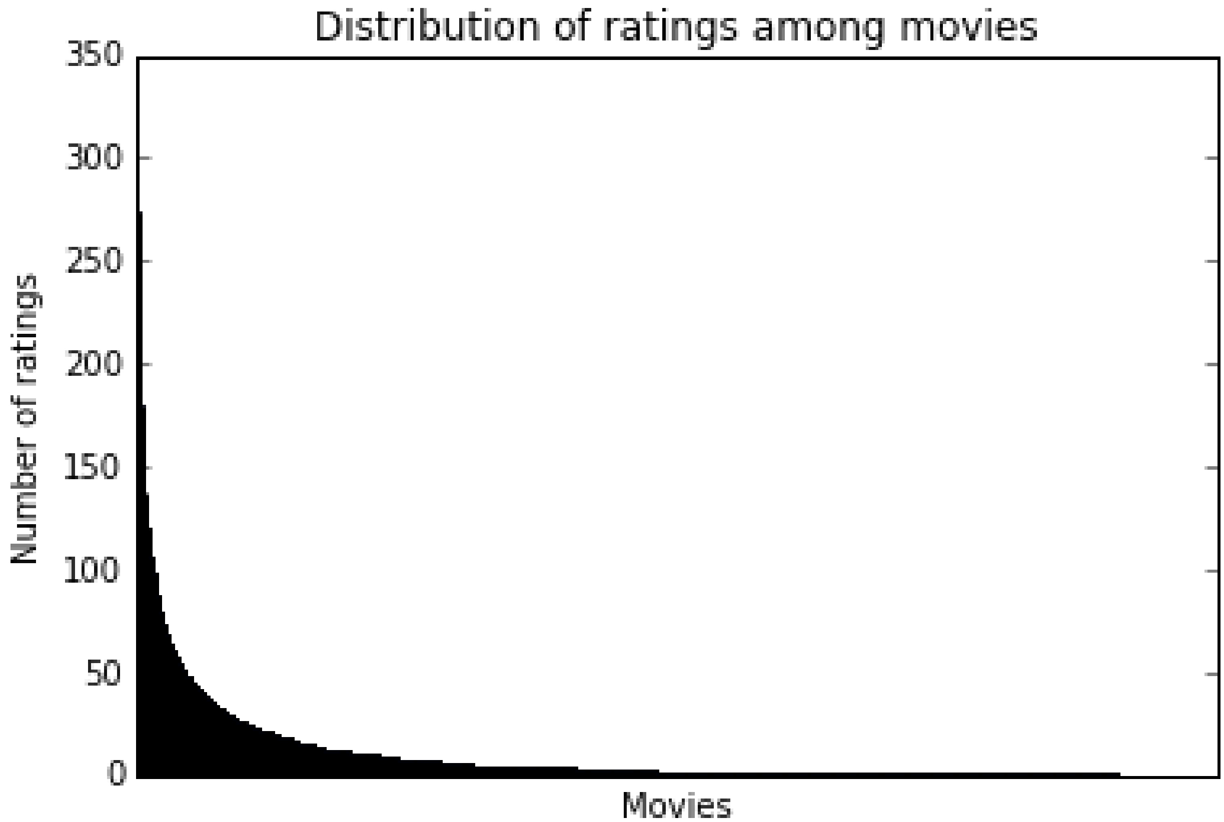

Figure 5: Number of ratings Vs Movie Index

4.4 Question 4: Distribution of Ratings among users

The plot shows the number of ratings each user has given, arranged in decreasing frequency. The plot indicates that very few users actively rate movies compared to others. More than half of the users have given less than 100 ratings. Once again this is a good indication of the sparsity of the ratings matrix.

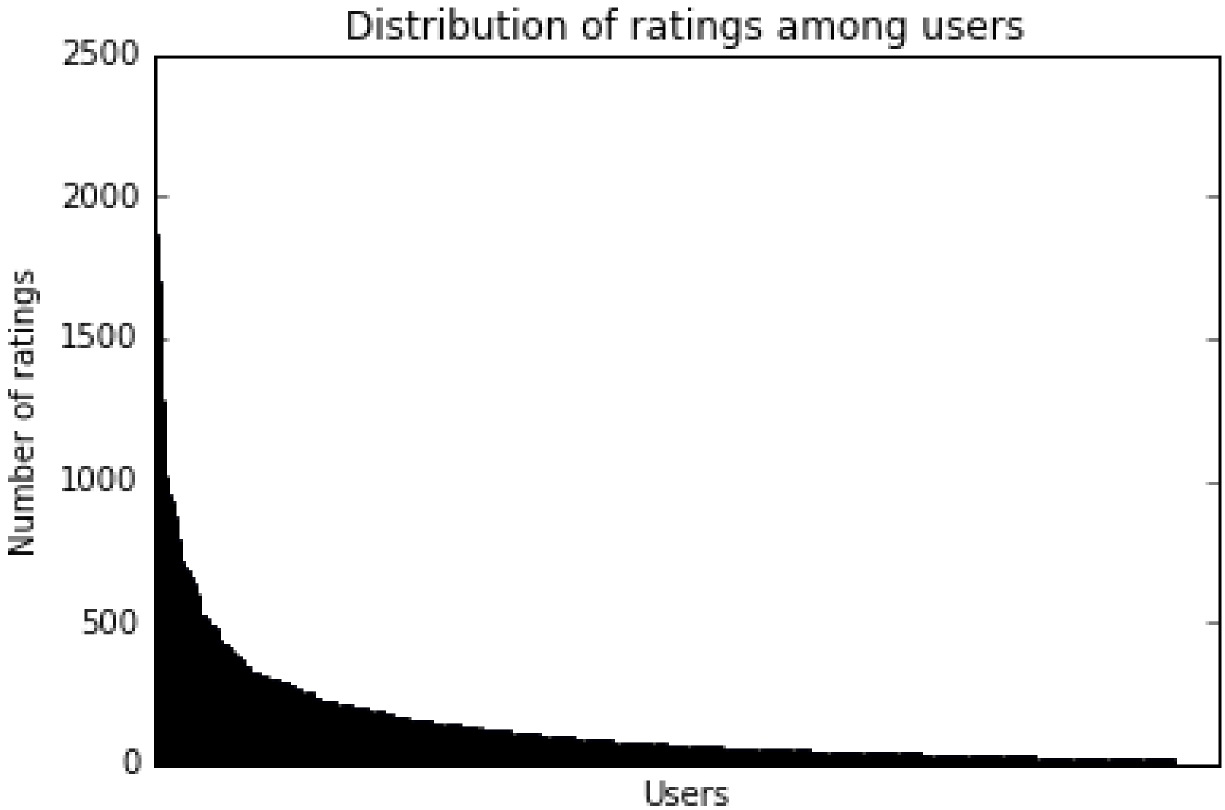

Figure 6: Number of ratings Vs User Index

4.5 Question 5: Implications of distribution of ratings among movies(Figure 2)

The exponentially decaying curve seen in the distribution of ratings among movies with a long tail implies that the ratings matrix will be sparse. Since only a few movies have received many ratings, there will be a lot of columns in the ratings matrix with missing values. With such sparse data, imputing the missing values becomes a challenge and hence methods such as collaborative filtering have to be employed to fill the missing data. The long tail in the distribution also indicates that a huge number of movies have received very little ratings. Those are the unpopular movies which have been seen by very few users. Predicting the ratings for such movies is a challenge as well since calculating the similarity will be based on the extremely small number of ratings.

4.6 Question 6: Distribution of variance of ratings

The plot shows the variance of user ratings binned into 0.5 variance bins. It is seen from the plot that the variance of most ratings are in the bin 0-1. This indicates that most users rate the movies with similar rating which suggests that most users do not give varied ratings. In a certain way this might add bias. For example, a certain user might always give a rating of 5 for all the movies he has seen. This will lead to a biased data distribution. It will be preferable to have some variance in the ratings though not too much.

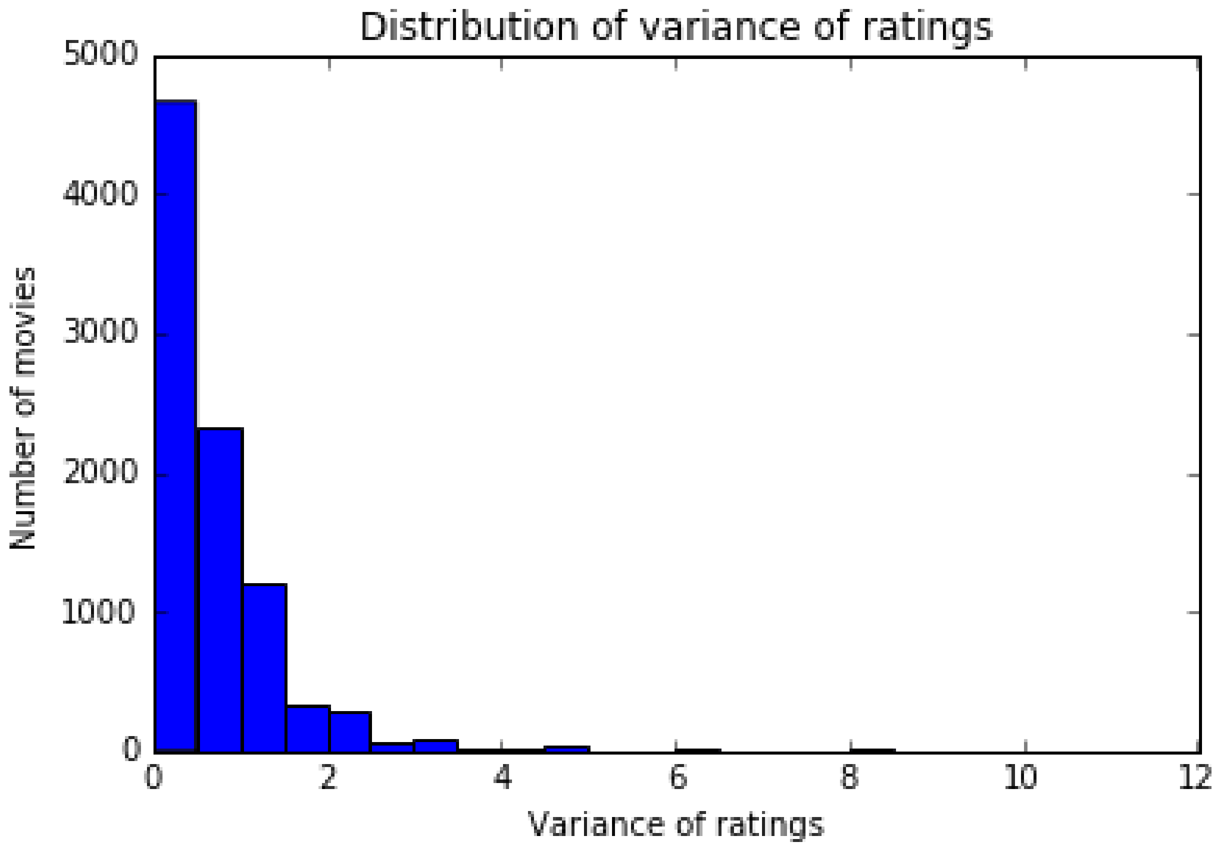

Figure 7: Number of ratings Vs Variance of ratings

4.7 Question 7

$$\mu_u = \frac{Sum\,of\,ratings\,given\,by\,user}{Number\,of\,ratings\,given\,by\,user} = \frac{\sum_{I_u} r_{uk}}{|I_u|} \tag{1}$$

4.8 Question 8

$I_u \cap I_v$ refers to the intersection of the movies seen by the two users. Concretely, it is the set of movies rated by both the users I_u and I_v. Since the ratings matrix is sparse, there is a possibility that $I_u \cap I_v = 0$. It implies that there is a likelihood that two users do not have any commonly seen movies.

4.9 Question 9

Mean centering the raw ratings helps to remove the effect of ratings given by biased users. As stated earlier when discussing the distribution of the variance of ratings given, there may be users who give only high ratings or only low ratings. It is necessary to nullify the effect of ratings given by such biased users. By removing the average rating given by the user, we are effectively taking only the variation in the ratings that the user has given, thereby nullifying the bias. Concretely, the importance of normalizing the data lies in the mitigation of the effect of biased ratings. For example, if a person decides to give only ratings of 5, after normalization, all his ratings will be changed to 0. Lets look at the heat map of the ratings before and after the normalization process for a sample of 30 users and 30 movies, to visualize this effect.

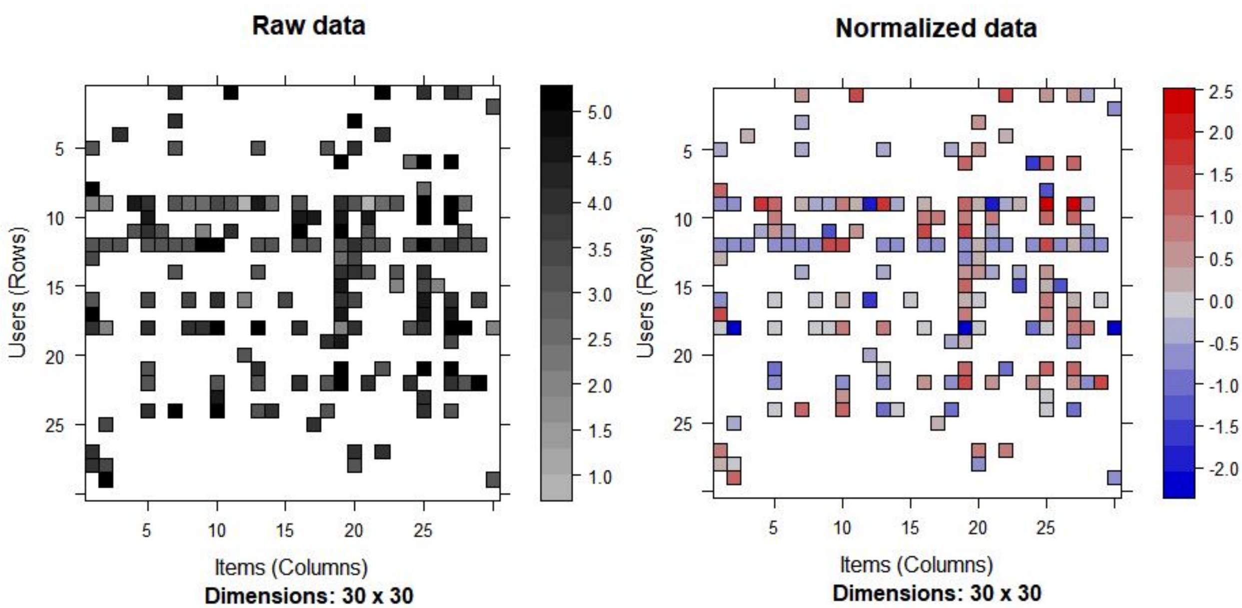

Figure 8: Heat map of data before and after normalization

4.10 Question 10

Testing a k-NN Collaborative filter with 10-fold cross-validation

MAE - Mean Absolute Error, measures the average magnitude of the errors in a set of predictions, without considering their direction. Its the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

RMSE - Root Mean Square Error is a quadratic scoring rule that also measures the average magnitude of the error. Its the square root of the average of squared differences between prediction and actual observation.

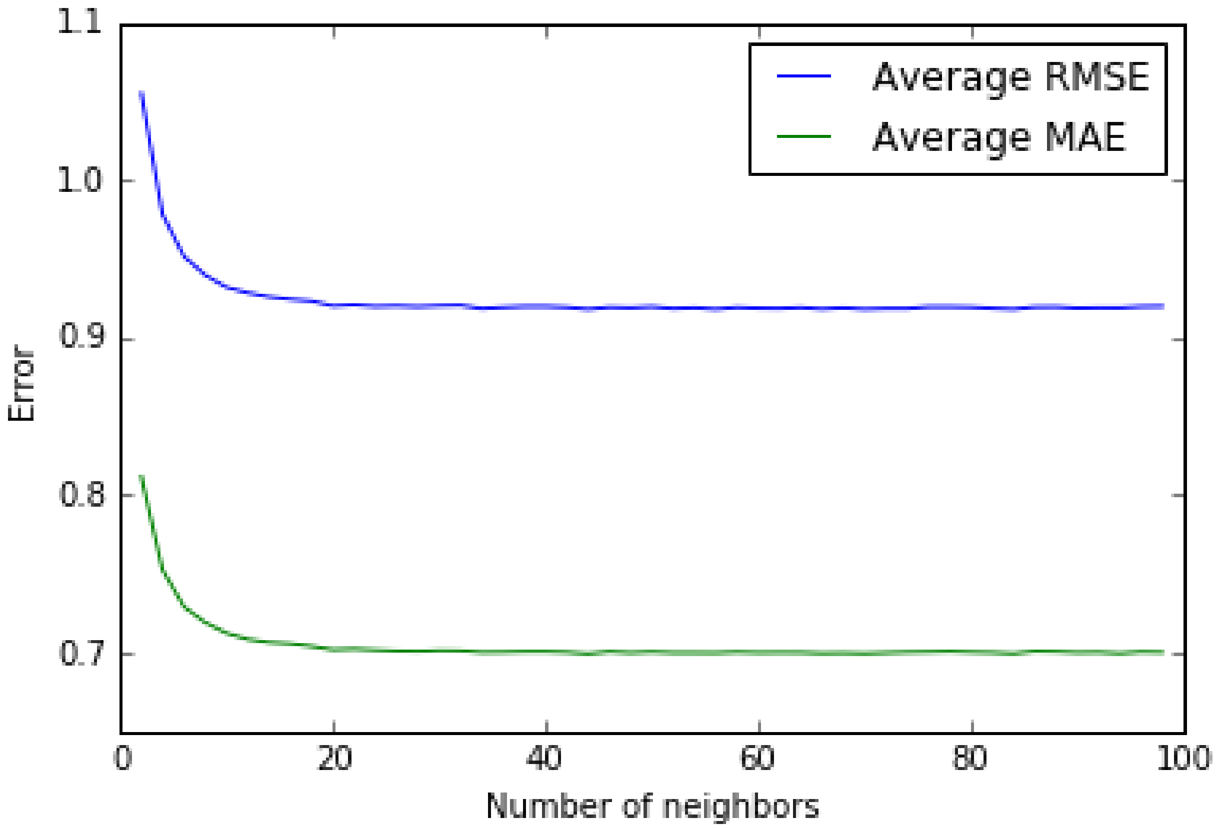

Figure 9: Average RMSE and MAE Vs Number of Neighbors(k) - All Movies - k-NN Collaborative filter

From the above understanding of the two errors, it can be seen from the figure 9 that MAE is lower in its value than RMSE due to the fact it takes the absolute magnitude and not the direction. From this it can be understood that the use of RMSE is more appropriate to evaluate the performance of the filters as it takes into account of the large errors which are undesirable.

4.11 Question 11

From the Figure 9 it can be seen that the **minimum number of neighbors(k)** at which the average RMSE and MAE converge to steady state is **20**.

Steady state values of average RMSE error:0.92
Steady state values of average MAE error:0.7

4.12 Question 12

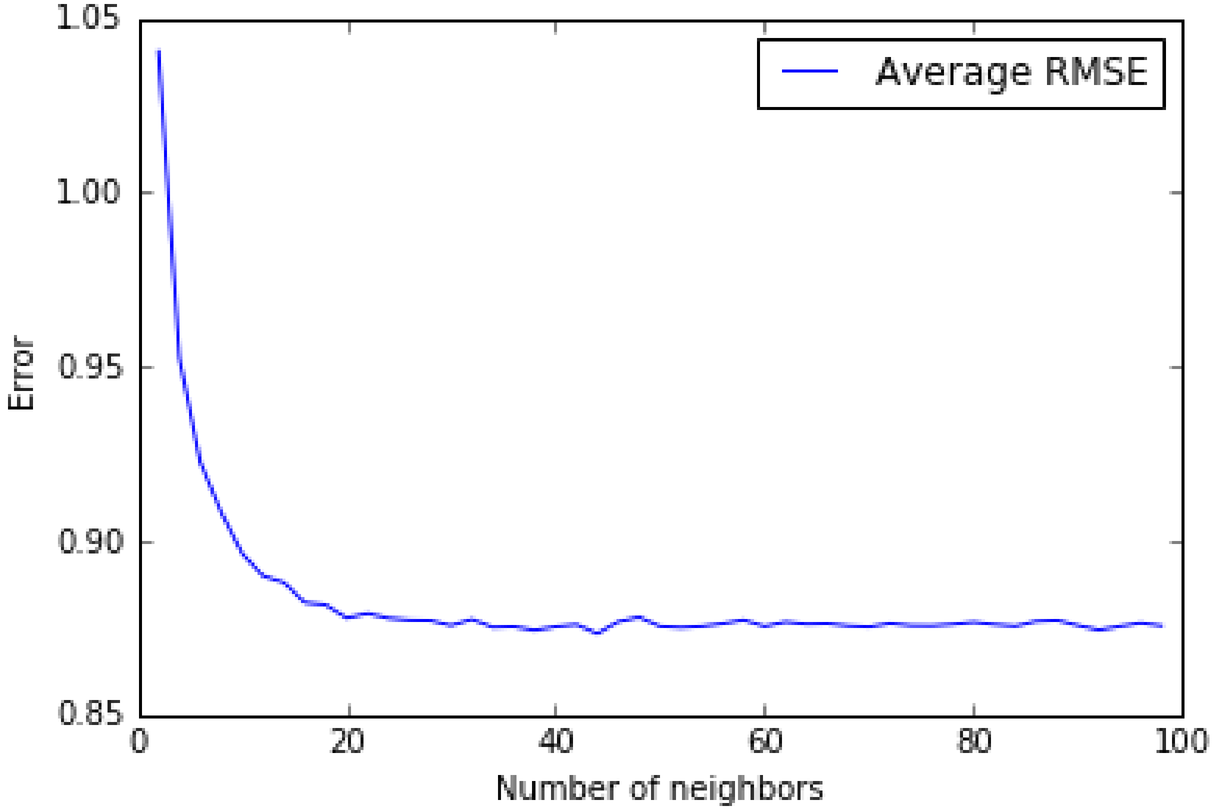

Figure 10: Average RMSE Vs Number of Neighbors(k) - Popular Movies - k-NN Collaborative filter

Evaluating k-NN collaborative filtering on popular movie trimmed set

The figure 10 is a plot of of the RMSE values averaged over the 10 cross folds on y-axis Vs the number of neighbors on the x-axis. The RMSE values obtained correspond to the performance of a k-NN collaborative filter applied on a popular movie trimmed set.

Interpretation It can be seen from the figure 10 that the average RMSE for predicting the rating of a movie from the popular movie trimmed set is exponentially decreasing with increase in number of neighbors on using a k-NN collaborative filter.

It can also be seen that the RMSE values reach a steady state of **0.8739** as the number of neighbors increase beyond 20, which implies that the filter is useful in predicting the ratings of popular movies when the number of neighbors are more than 20, for the given data set.

The minimum average RMSE in predicting the rating of a movie in the popular movie trimmed set is **0.8739 for k = 20**.

4.13 Question 13

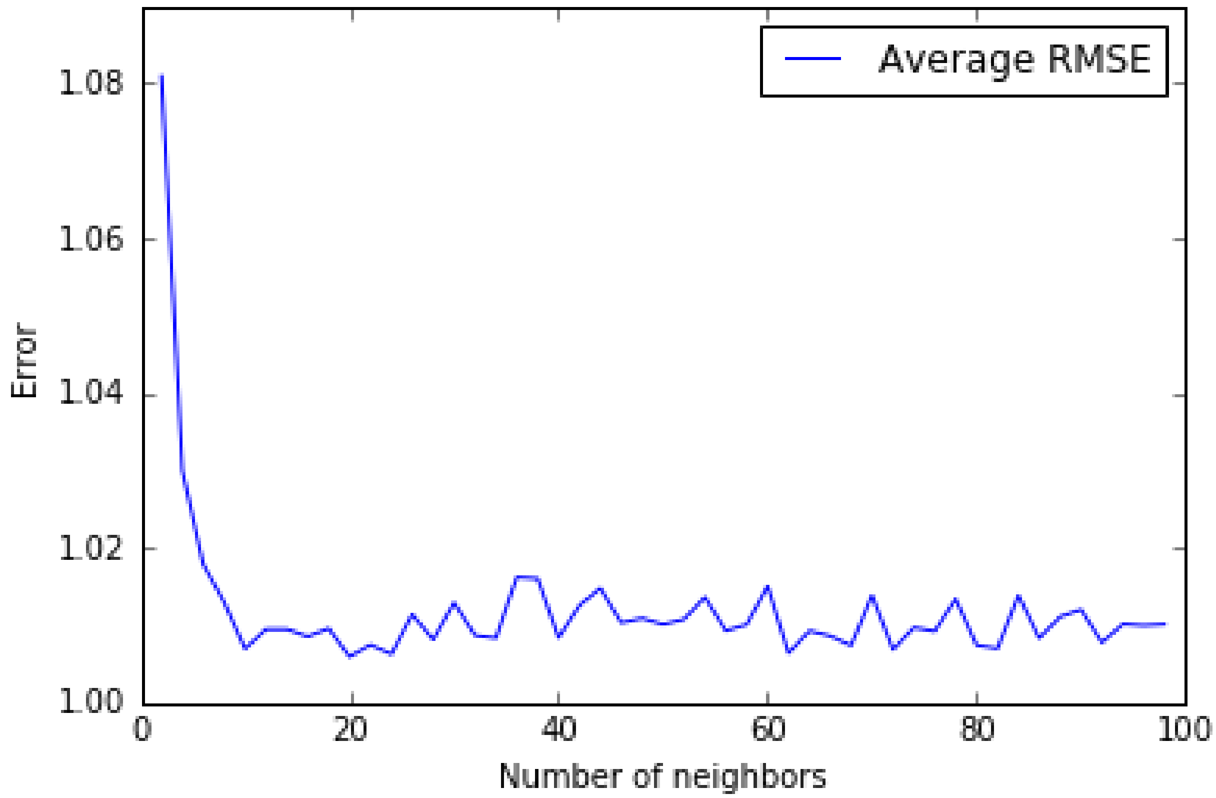

Figure 11: Average RMSE Vs Number of Neighbors(k) - Unpopular Movies - k-NN Collaborative filter

Evaluating k-NN collaborative filtering on unpopular movie trimmed set

The minimum average RMSE in predicting the rating of a movie in the unpopular movie trimmed set is **1.00596 for k = 18**.

Interpretation As seen from the graph there is more variations in this graph as compared to the popular movie trimmed graph. This is due to the fact that the unpopular movies have received very little ratings and hence may contain more variations. It does not contain enough ratings to get a smoothened effect.

4.14 Question 14

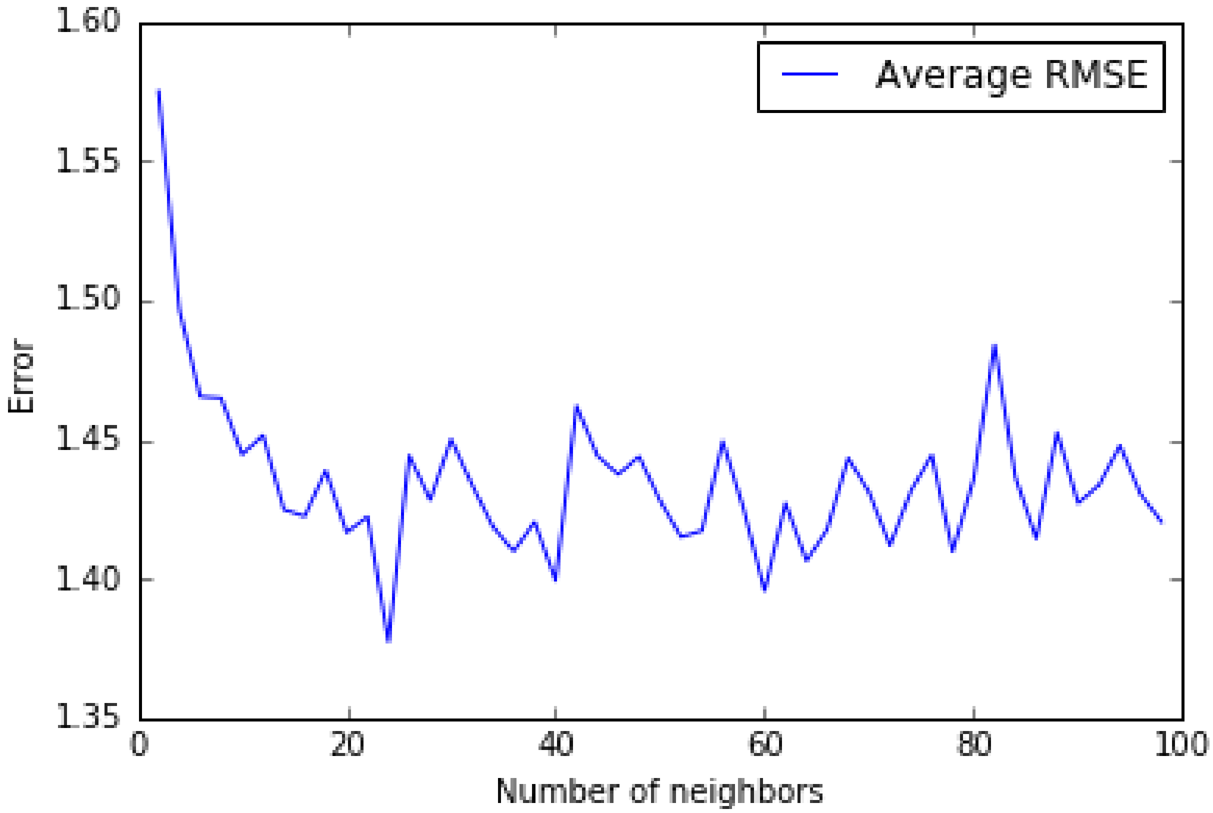

Figure 12: Average RMSE Vs Number of Neighbors(k) - High variance Movies - k-NN Collaborative filter

Evaluating k-NN collaborative filtering on high variance movie trimmed set

The minimum average RMSE in predicting the rating of a movie in the high variance movie trimmed set is **1.3773 for k = 22**.

Interpretation Here there are far more variations in the graph as compared to the popular and unpopular trimmed movie data sets. This is due to the fact that we are using only the movies that have high variance, thereby displaying erratic behavior. However, the best value of number of neighbors is still around 20.

4.15 Question 15

ROC Curve for k-NN collaborative filter for different thresholds

4.15.1 ROC Curve with threshold = 2.5

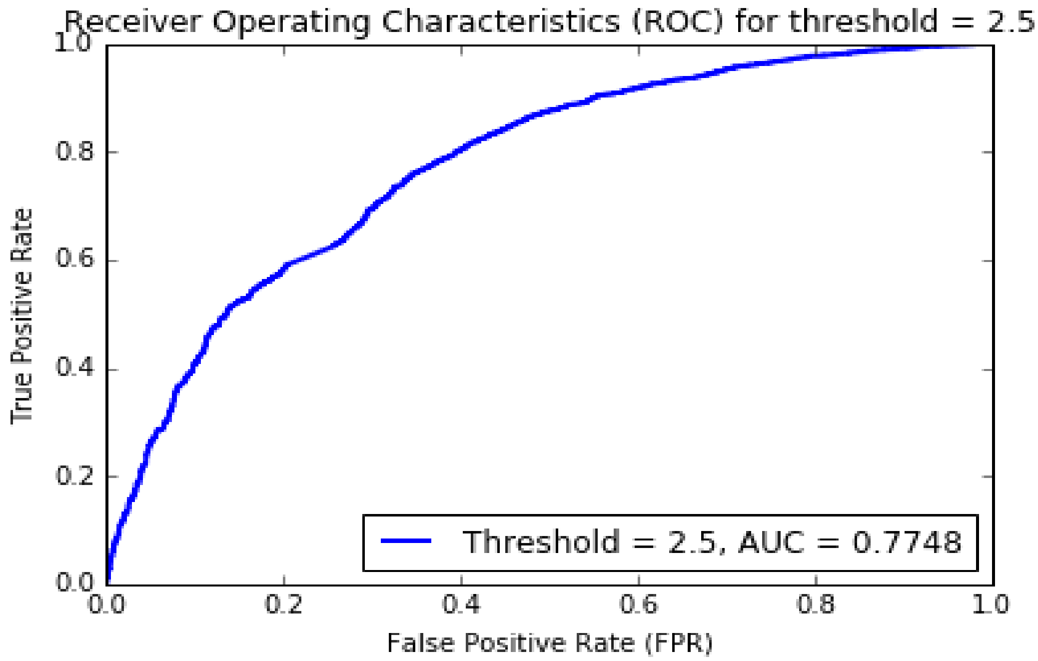

Figure 13: ROC Curve for k-NN Collaborative filter with threshold=2.5, k=20

4.15.2 ROC Curve with threshold = 3

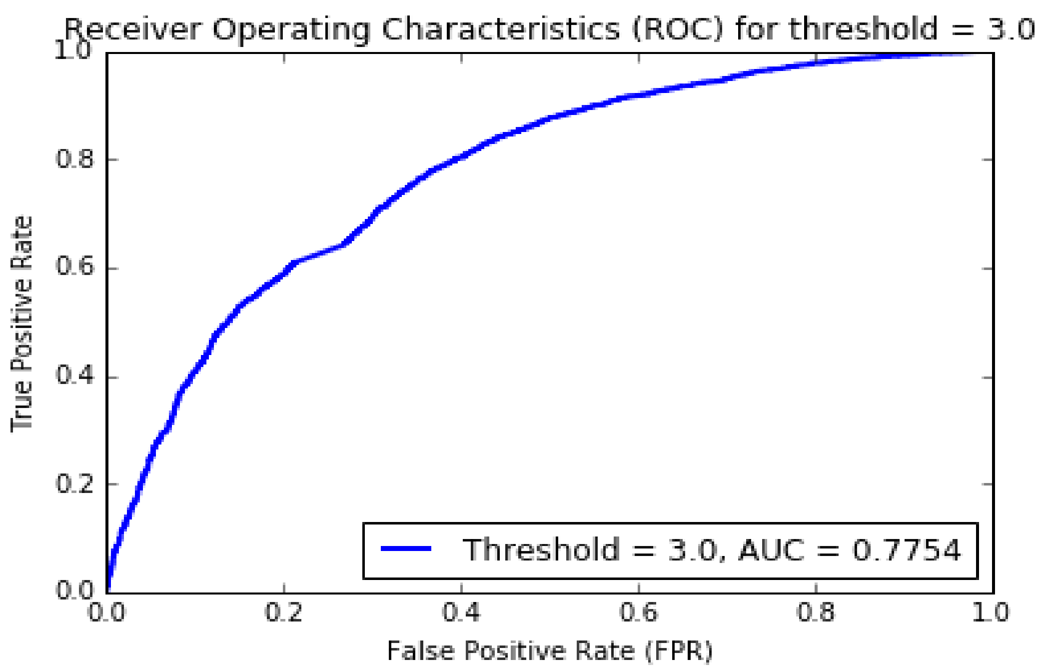

Figure 14: ROC Curve for k-NN Collaborative filter with threshold=3, k=20

4.15.3 ROC Curve with threshold = 3.5

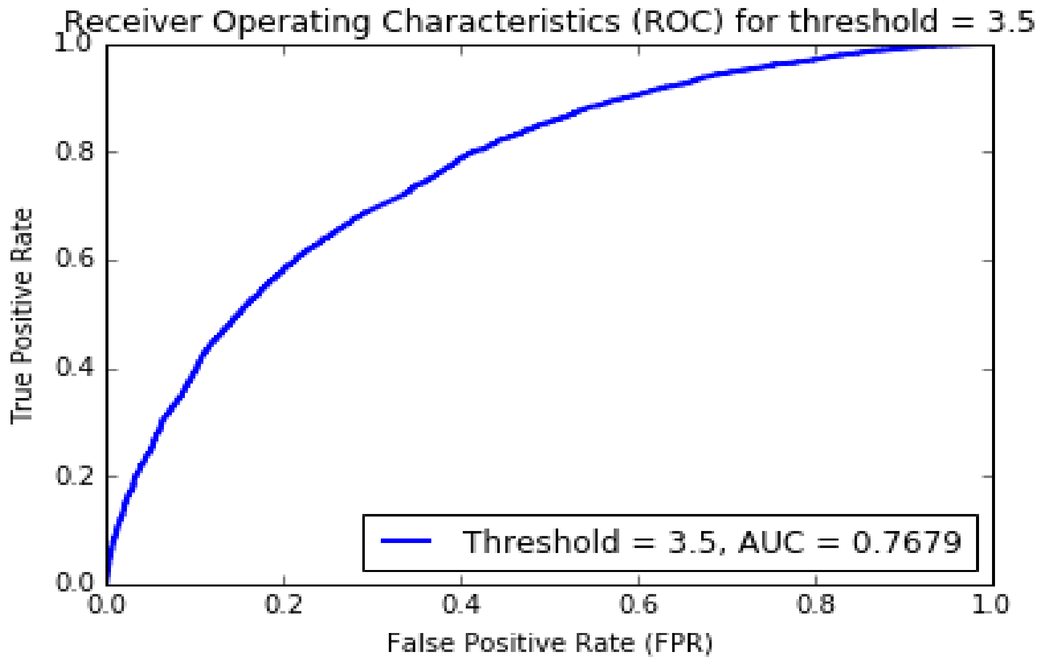

Figure 15: ROC Curve for k-NN Collaborative filter with threshold=3.5, k=20

4.15.4 ROC Curve with threshold = 4

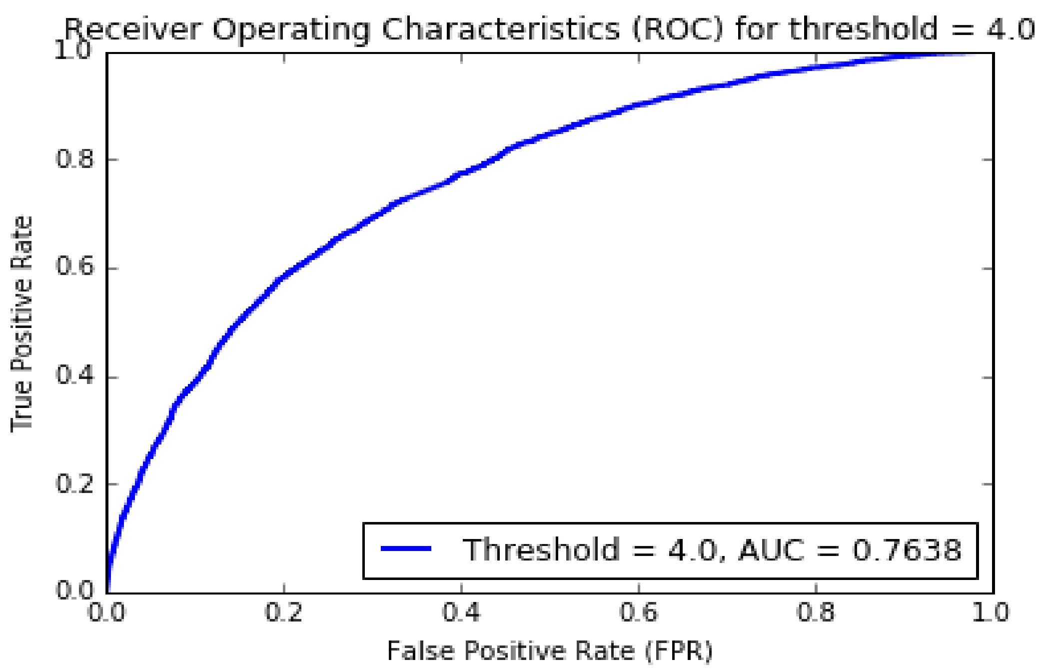

Figure 16: ROC Curve for k-NN Collaborative filter with threshold=4, k=20

Threshold	Area under the curve
2.5	0.7748
3	0.7754
3.5	0.7679
4	0.7638

Interpretation There is just a marginal difference in performance for the different thresholds. By just comparing the area under the curve for the different thresholds, it can be seen that setting the threshold to be 3 produces the best results. Hence, 3 can be used as a threshold setting for all future problems and it will produce the best performance.

4.16 Question 16

The given objective function is not convex. Our argument is as follows. If it were convex, then every projection of the function should be convex. Projecting it on to the matrix components, we find that it is not convex which can be proved using first principles. However, the objective function is convex just U, keeping V fixed or in V keeping U fixed. This can be proved by writing it as a summation of multiple convex functions. The objective function written as a least squares is:

$$\|\widetilde{R} - \widetilde{U}\widetilde{V}\|^2 \tag{2}$$

where

$$\widetilde{R} = [W_{11}r_{11}, W_{21}r_{21}, ...] \tag{3}$$

4.17 Question 17

Implementation We design a Non-Negative Matrix Factorization colloborative filter in this section. The performance is evaluated using a 10-fold cross validation.Also, average RMSE and average MAE is computed for different values of latent factors using this filter. Number of latent factors are varied from 2 to 50 in step sizes of 2 and then we compute RMSE and MAE. We do a 10-fold cross validation and for each latent factor, we split the entire data set into training and test sets.

NMF and KFold, train_test_split of model_selection under surprise is used. The learning rate is set to 0.005 (default) while the regularization is 0.02. As we increase this regularization parameter, we can prevent overfitting and force the model to generalize better. However, it is prudent to keep the regularization parameter under reasonable limits so that we do not underfit and prevent the model from learning at all.

Testing a NNMF-based Collaborative filter with 10-fold cross-validation

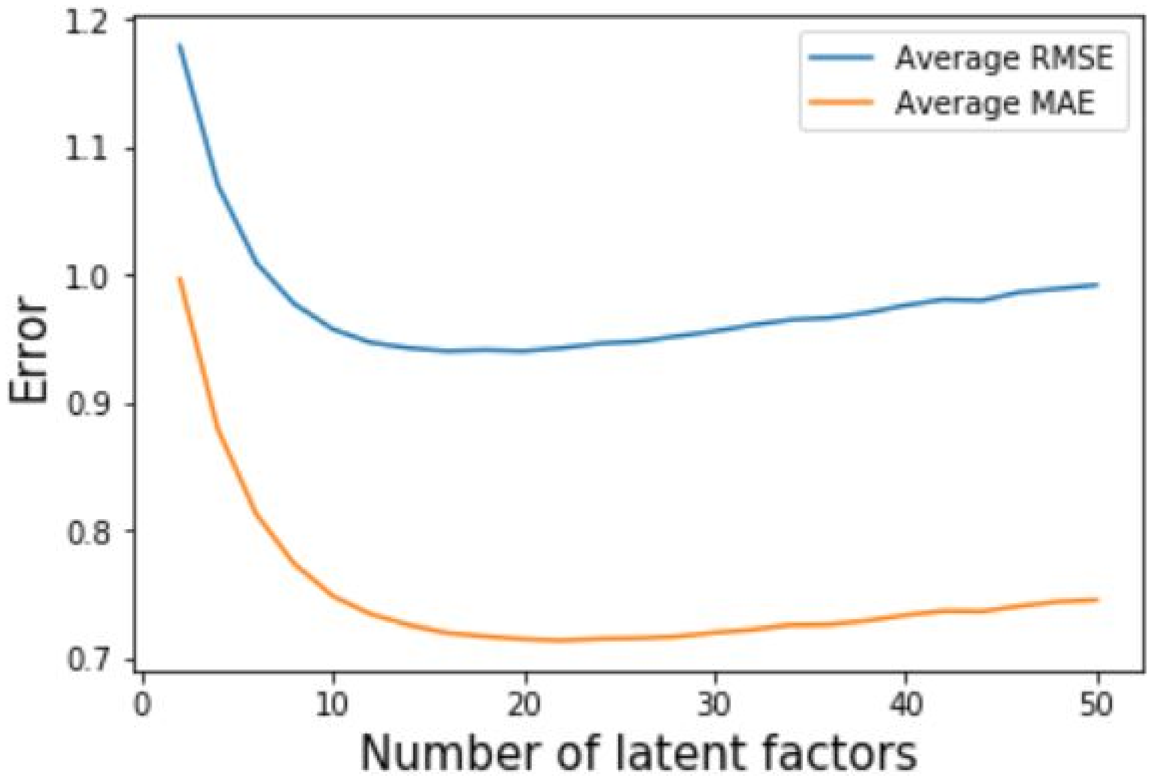

Figure 17: Average RMSE and MAE Vs Number of Latent Factors) - All Movies- NNMF-based Collaborative filter

Interpretation It can be seen that Average RMSE is higher than Average MAE. This can be verified with Jensen's inequality. Also, intuitively since square of RMSE is effectively the sum of mean square and square of standard deviation.

4.18 Question 18

From the Figure:17, it can be seen that the optimal latent factor with lowest average RMSE and MAE is 20.

Minimum average RMSE error : 0.975
Minimum average MAE error : 0.712

Number of Genres - 20
Yes, the number of genres and the optimal latent factor is the same - 20.

4.19 Question 19

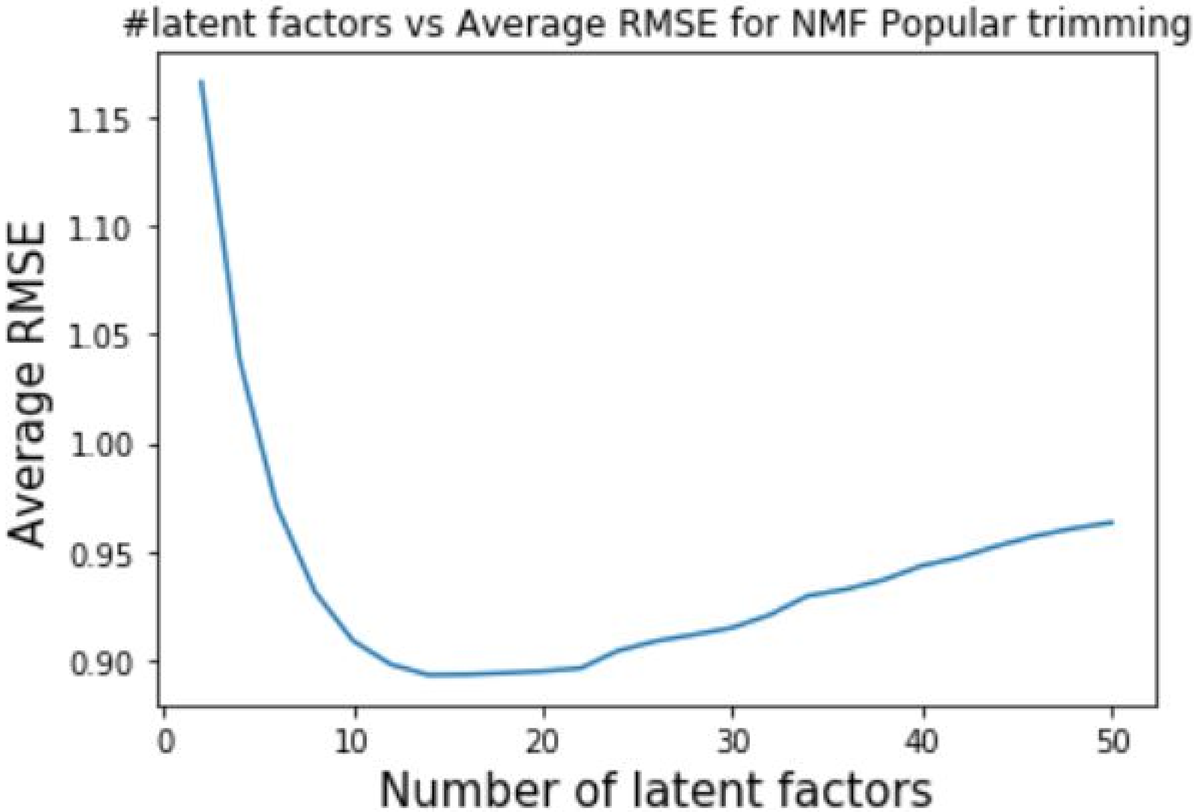

Figure 18: Average RMSE Vs Number of Latent Factors(k) - Popular Movies NNMF-based Collaborative filter

Evaluating NNMF collaborative filtering on popular movie trimmed set The minimum average RMSE in predicting the rating of a movie in the popular movie trimmed set is **0.8929 for k = 14**.

Interpretation As the number of latent factors increase, the average RMSE initially decreasing by a huge margin and then starts to increase again after a point. The minimum RMSE can be reached when the number of latent factors is between 10 to 20. This corresponds to the genres of the popular trimmed dataset. Since there is lesser variance in the popular trimmed dataset, we are able to get a smooth curve for this.

4.20 Question 20

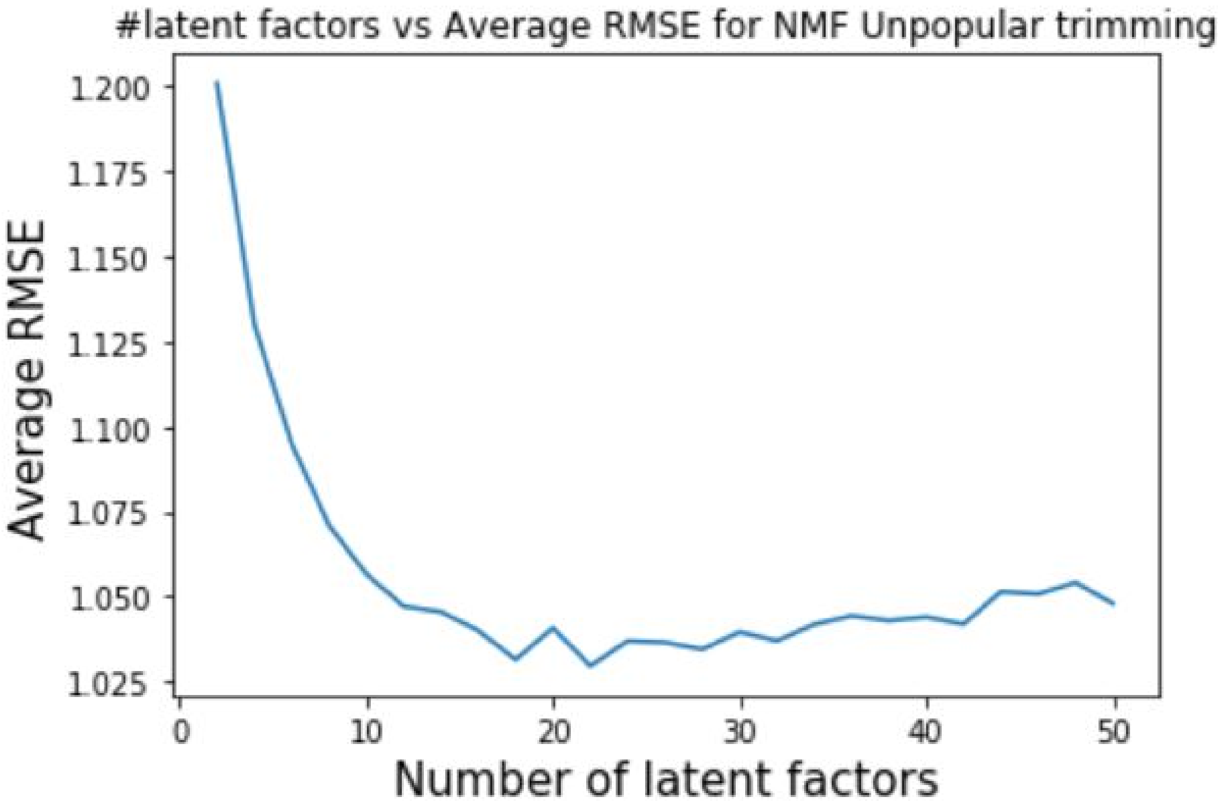

Figure 19: Average RMSE Vs Number of Latent Factors(k) - Unpopular Movies NNMF-based Collaborative filter

Evaluating NNMF collaborative filtering on unpopular movie trimmed set

The minimum average RMSE in predicting the rating of a movie in the unpopular movie trimmed set is **1.0297 for k = 22**.

Interpretation As the number of latent factors increase, the average RMSE initially decreasing by a huge margin and then starts to achieve a steady state with small fluctuations. The minimum RMSE can be reached when the number of latent factors is larger than 18. Since there is more variance in the unpopular trimmed dataset, we see more fluctuations here.

4.21 Question 21

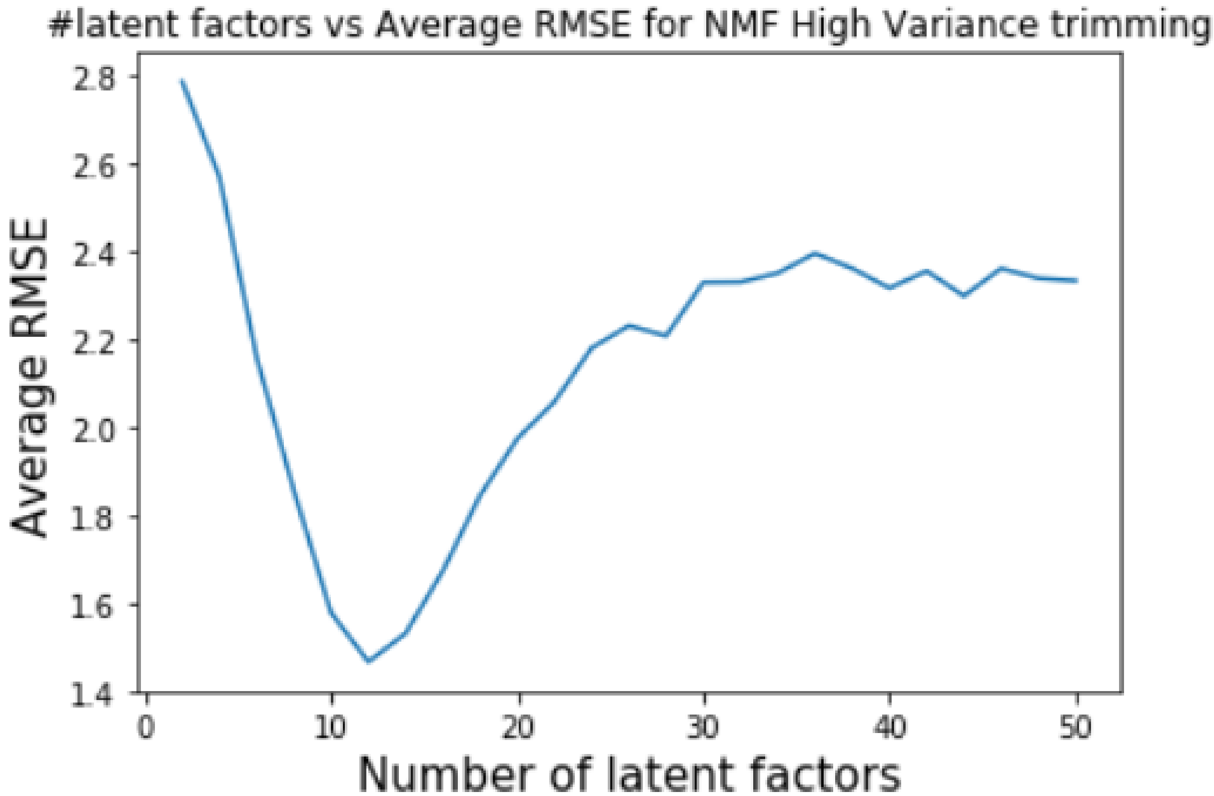

Figure 20: Average RMSE Vs Number of latent factors - High variance Movies - NNMF Collaborative filter

Evaluating NNMF collaborative filtering on high variance movie trimmed set

The minimum average RMSE in predicting the rating of a movie in the High variance movie trimmed set is **1.4068 for k = 12**.

Interpretation As the number of latent factors increase, the average RMSE initially decreasing by a huge margin and then starts to increase drastically before reaching a steady state. The minimum RMSE can be reached when the number of latent factors is around 12. Since high variance trimmed dataset has large variance, we are able to see a high variation in the average RMSE as the number of latent factors change.

4.22 Question 22

ROC Curve for NNMF collaborative filter for different thresholds

4.22.1 ROC curve with threshold = 2.5

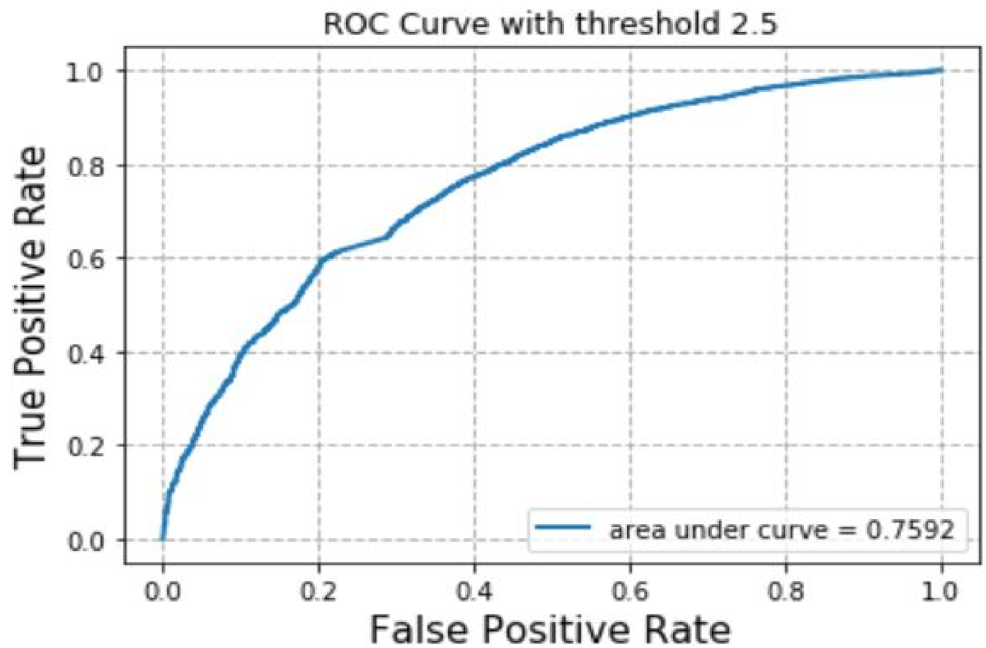

Figure 21: ROC Curve for NNMF Collaborative filter with threshold=2.5, k=20

4.22.2 ROC curve with threshold = 3

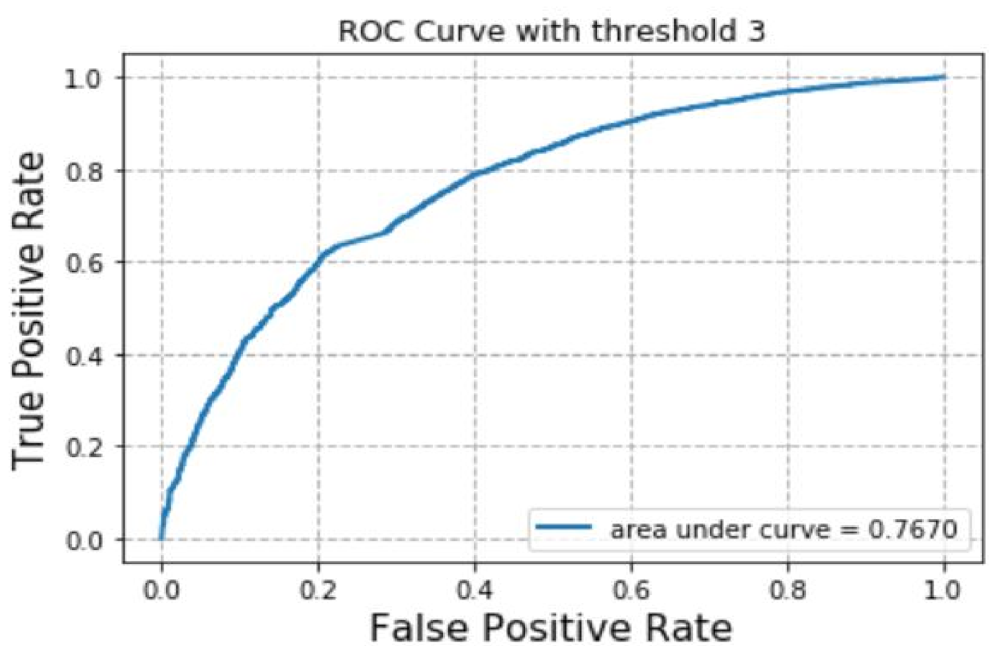

Figure 22: ROC Curve for NNMF Collaborative filter with threshold=3, k=20

4.22.3 ROC curve with threshold = 3.5

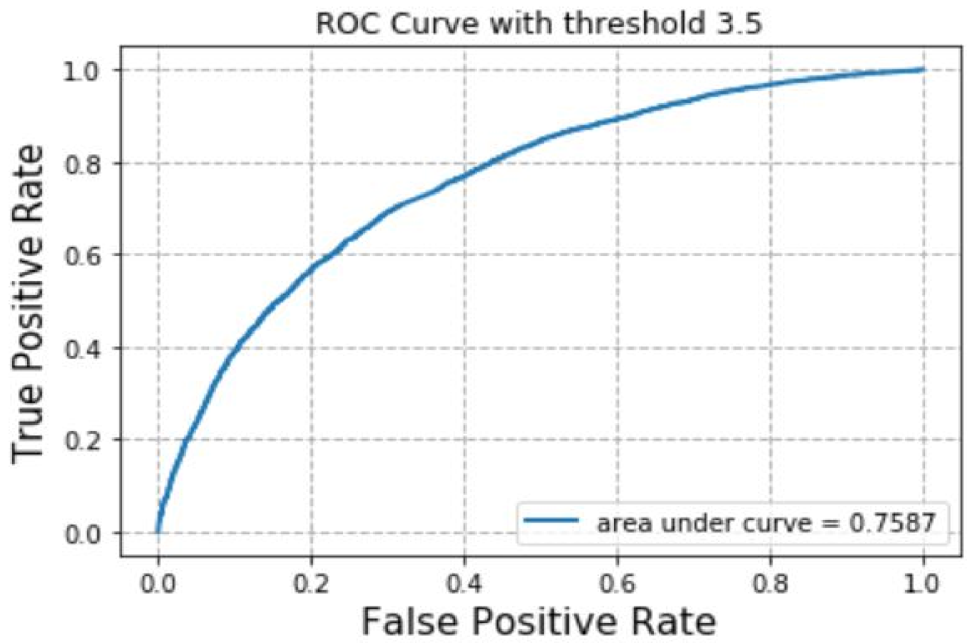

Figure 23: ROC Curve for NNMF Collaborative filter with threshold=3.5, k=20

4.22.4 ROC curve with threshold = 4

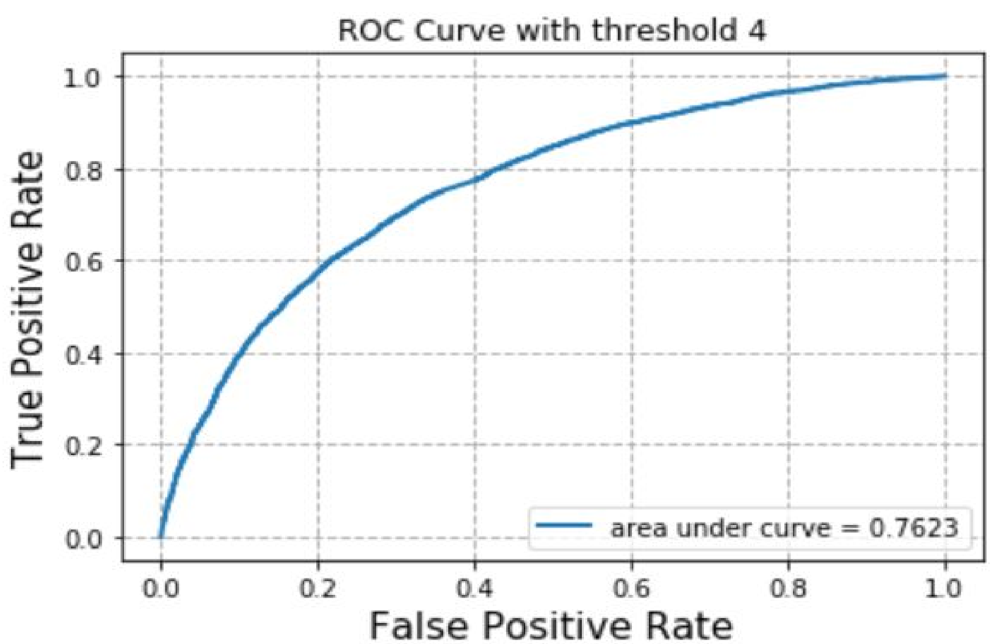

Figure 24: ROC Curve for NNMF Collaborative filter with threshold=4, k=20

Threshold	Area under the curve
2.5	0.7592
3	0.7670
3.5	0.7587
4	0.7623

Interpretation There is just a marginal difference in performance for the different thresholds. By just comparing the area under the curve for the different thresholds, it can be seen that setting the threshold to be 3 produces the best results. Hence, 3 can be used as a threshold setting for all future problems and it will produce the best performance.

4.23 Question 23

Implementation In this part, we perform Non-negative matrix factorization on the ratings matrix to obtain user-latent factors interaction matrix and movie-latent factors interaction matrix. For the latter matrix, the movies are sorted in descending order and the corresponding genres are listed. Optimal latent factor of 20 was taken for this task. Here we list the genres of the top-10 movies for the first four columns of the movie-latent factors matrix.

Table 1: Genres of top-10 movies for different latent factors

<table>
<tr><th>Column 1</th><th>Column 2</th><th>Column 3</th><th>Column 4</th></tr>
<tr><td>Comedy|Drama|Romance</td><td>Adventure|Children|Fantasy</td><td>Comedy|Horror</td><td>Action|Crime|Drama</td></tr>
<tr><td>Comedy</td><td>Adventure|Children</td><td>Horror</td><td>Crime|Drama|Thriller</td></tr>
<tr><td>Comedy</td><td>Adventure|Children</td><td>Crime|Thriller</td><td>Crime|Drama|Horror</td></tr>
<tr><td>Comedy</td><td>Adventure|Children</td><td>Horror</td><td>Drama</td></tr>
<tr><td>Comedy|Drama</td><td>Action|Drama|Sci-Fi</td><td>Drama|Thriller</td><td>Crime|Drama|Romance</td></tr>
<tr><td>Drama|Musical</td><td>Adventure|Children|Fantasy</td><td>Drama</td><td>Action|Crime</td></tr>
<tr><td>Comedy</td><td>Adventure|Children</td><td>Comedy|Horror</td><td>Documentary</td></tr>
<tr><td>Drama</td><td>Adventure|Children|Fantasy</td><td>Drama|Romance</td><td>Action|Drama</td></tr>
<tr><td>Comedy|Drama|Romance</td><td>Action|Thriller</td><td>Horror</td><td>Crime|Thriller</td></tr>
<tr><td>Comedy|Fantasy|Romance</td><td>Comedy|Crime|Thriller</td><td>Horror</td><td>Action|Crime|Drama</td></tr>
</table>

Interpretation It can be easily seen that each column, the top-10 movies belong to similar genres. For example, the first column has mostly movies in the genre Comedy. The second column in Adventure, the third column in Horror and the fourth column in Crime. This indicates that the latent factors are closely related to the number of genres and we achieve a good filtering using NMF which can help in aggregating similar genre movies and provide a precise recommendation. If a user, tends to like more movies of the same genre, then other top movies can be recommended based on this grouping.

4.24 Question 24

Implementation We design a Matrix Factorization with bias based colloborative filter in this section. The performance is evaluated using a 10-fold cross validation.Also, average RMSE and average MAE is computed for different values of latent factors using this filter. Number of latent factors are varied from 2 to 50 in step sizes of 2 and then we compute RMSE and MAE. We do a 10-fold cross validation and for each latent factor, we split the entire data set into training and test sets.

SVD and KFold, train_test_split of model_selection under surprise is used. The learning rate is set to 0.005 (default) while the regularization is 0.02. As we increase this regularization parameter, we can prevent overfitting and force the model to generalize better. However, it is prudent to keep the regularization parameter under reasonable limits so that we do not underfit and prevent the model from learning at all.

As it can be seen below, we did not observe a proper pattern with change in RMSE/MAE with change in number of latent factors like we observed for kNN or NMF. The change is not in any inferable correlation with the number of latent factors. So, to observe how Matrix Factorization works better, we tried changing the regularization, learning rate and initial mean as hyperparameters and observed that the highest change occurred with change in initial mean.

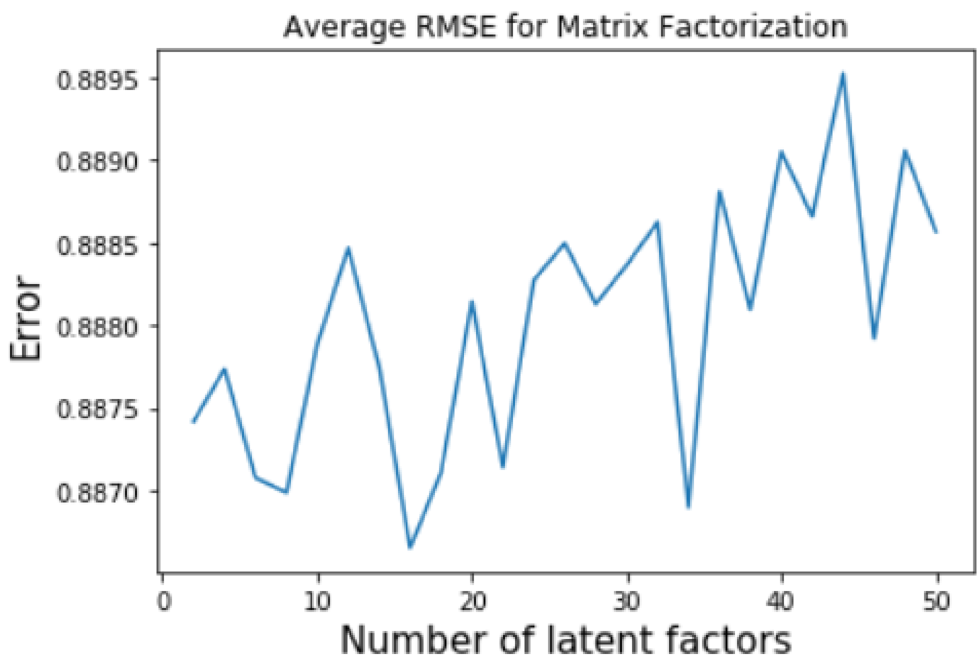

Figure 25: Average RMSE Vs Number of Latent Factors(k) - MF with bias

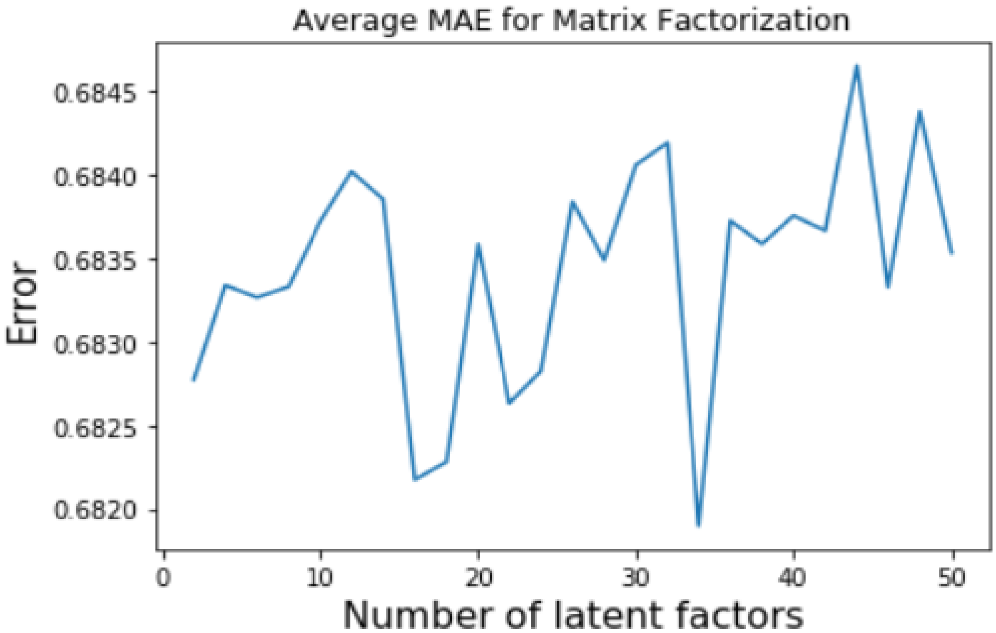

Figure 26: Average MAE Vs Number of Latent Factors(k) - MF with bias

Initial mean (init_mean) is the mean of the normal distribution for factor vectors initialization and by default it is zero. We changed it in steps of 0.5 till 5 and observed the error values. Here, we report it for the best case = 1.5.

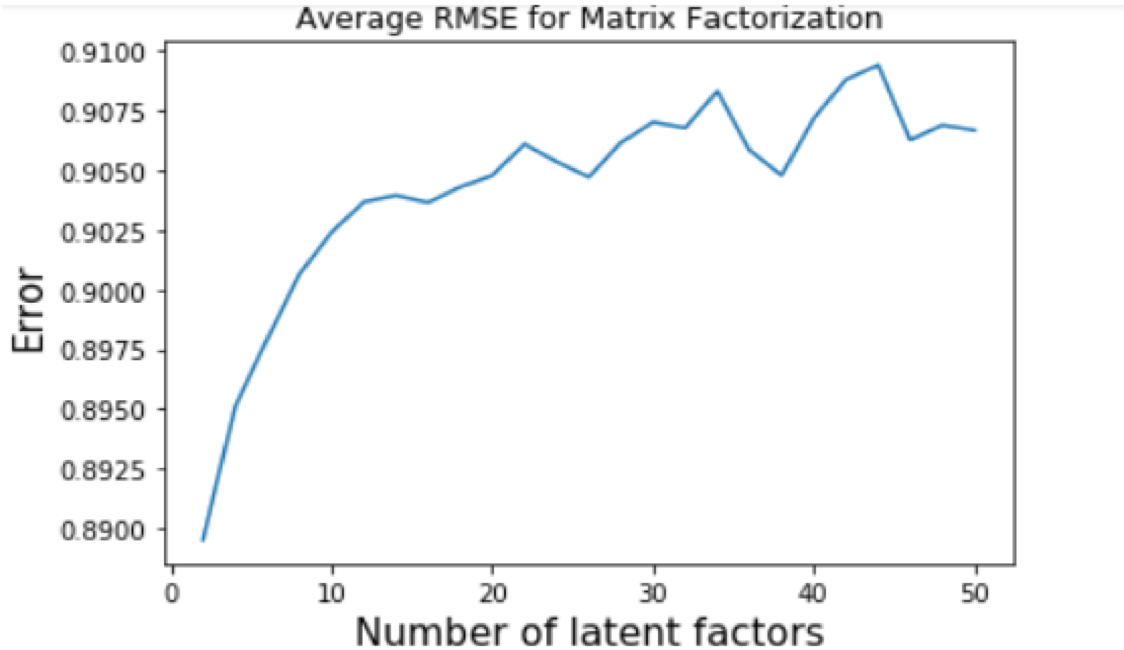

Figure 27: Average RMSE Vs Number of Latent Factors(k) - MF with bias - initial_mean = 1.5

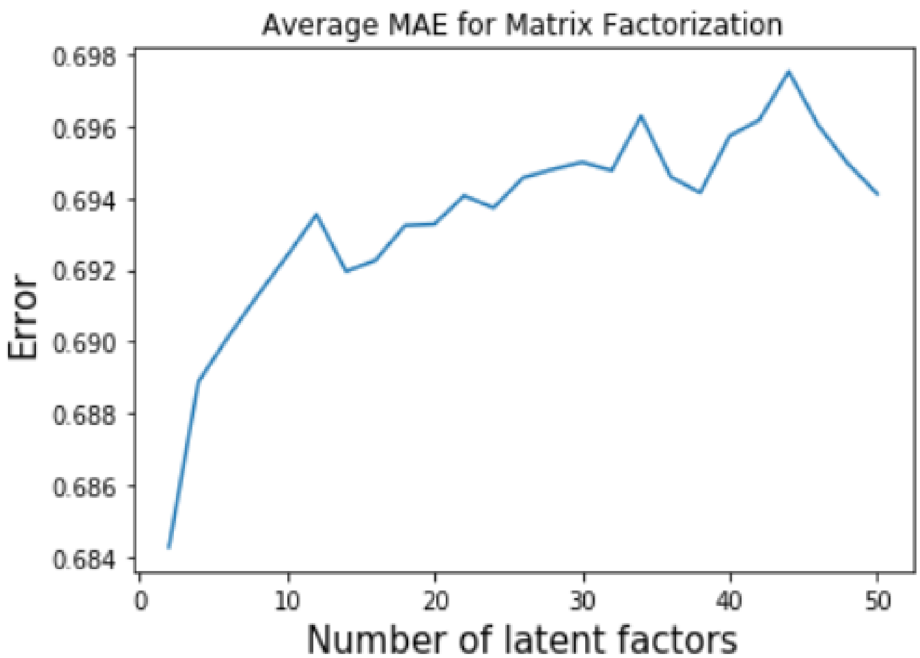

Figure 28: Average MAE Vs Number of Latent Factors(k) - MF with bias - initial_mean = 1.5

It can be observed that the graphs have become smoother than the default case. We can see that with increase in latent factors, the RMSE and MAE error increases and starts to saturate with high latent factors. We also set the initial mean to be the global mean of all the ratings in the training set and test the same. The global mean was observed to be 3.5450318322722576.

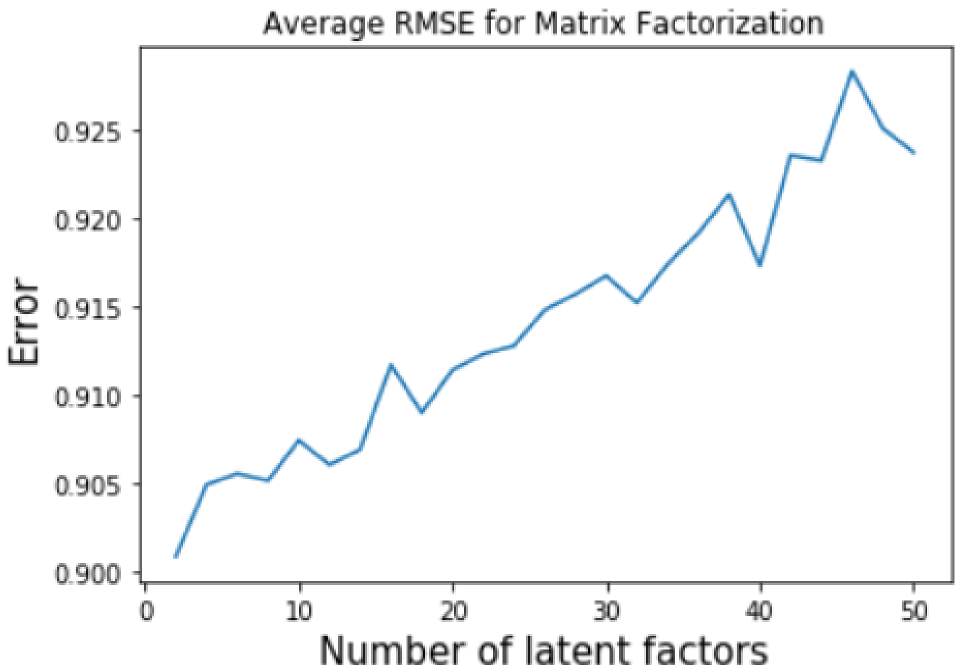

Figure 29: Average RMSE Vs Number of Latent Factors(k) - MF with bias - initial_mean = global_mean

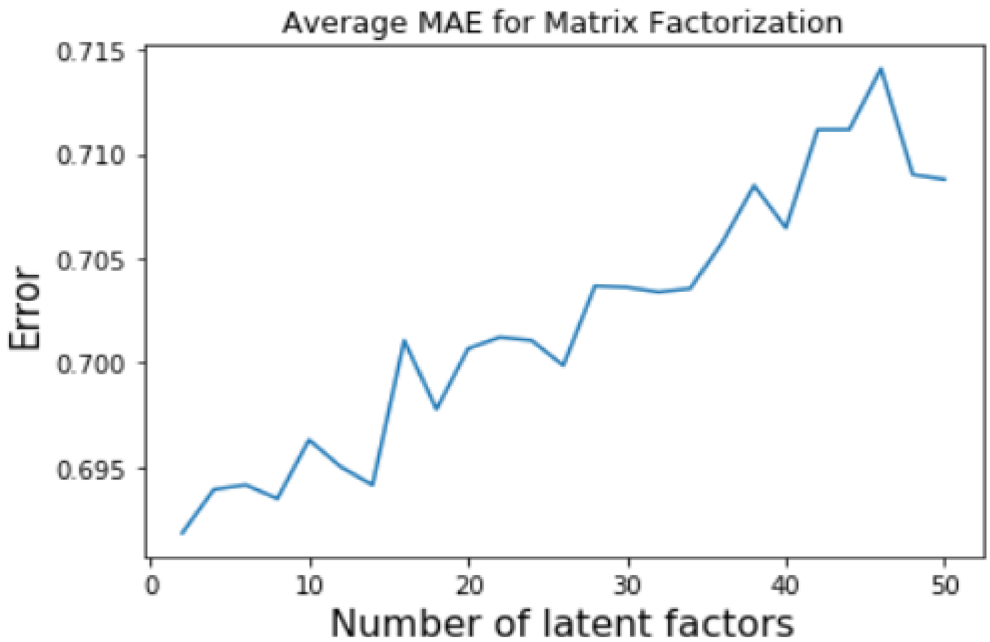

Figure 30: Average MAE Vs Number of Latent Factors(k) - MF with bias - initial_mean = global_mean

4.25 Question 25

From the figures for the default case, it can be seen that the optimal latent factor with lowest average RMSE is 16 and MAE is 34.

Minimum average RMSE error : 0.885
Minimum average MAE error : 0.682

For the case with init_mean = 1.5, we see that,
Minimum average RMSE error at k = 2 : 0.89
Minimum average MAE error at k = 2 : 0.684

For the case with init_mean = global_mean, we see that,
Minimum average RMSE error at k = 2 : 0.90
Minimum average MAE error at k = 2 : 0.695

4.26 Question 26

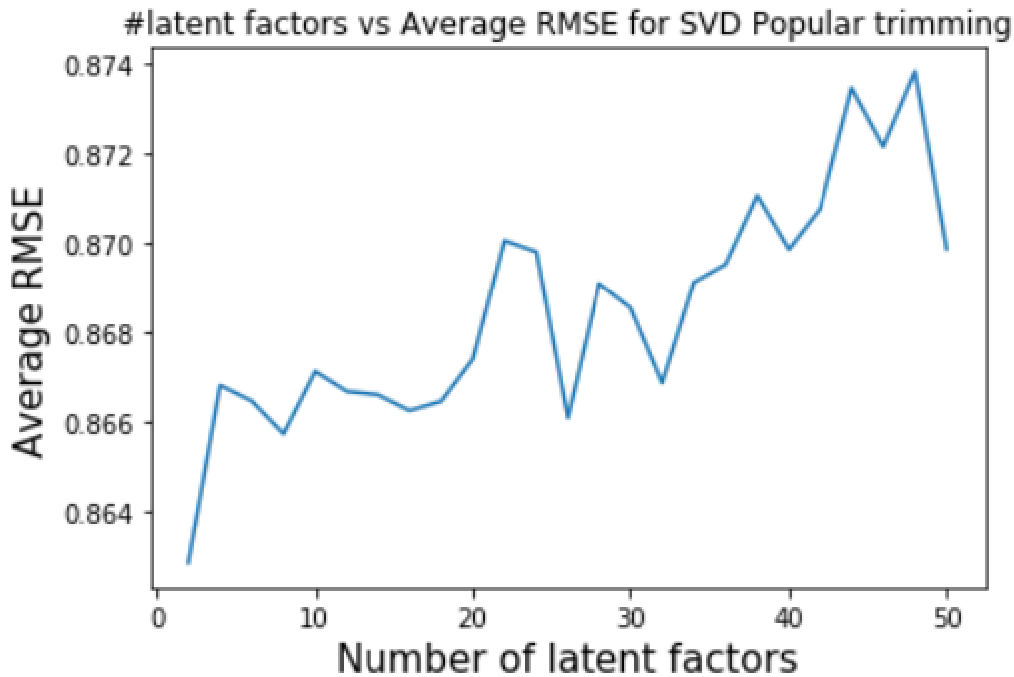

Figure 31: Average RMSE Vs Number of Latent Factors(k) - MF with bias - Popular movies

Evaluating MF with bias collaborative filtering on popular movie trimmed set

The minimum average RMSE in predicting the rating of a movie in the popular movie trimmed set is **0.863 for k = 2**.

4.27 Question 27

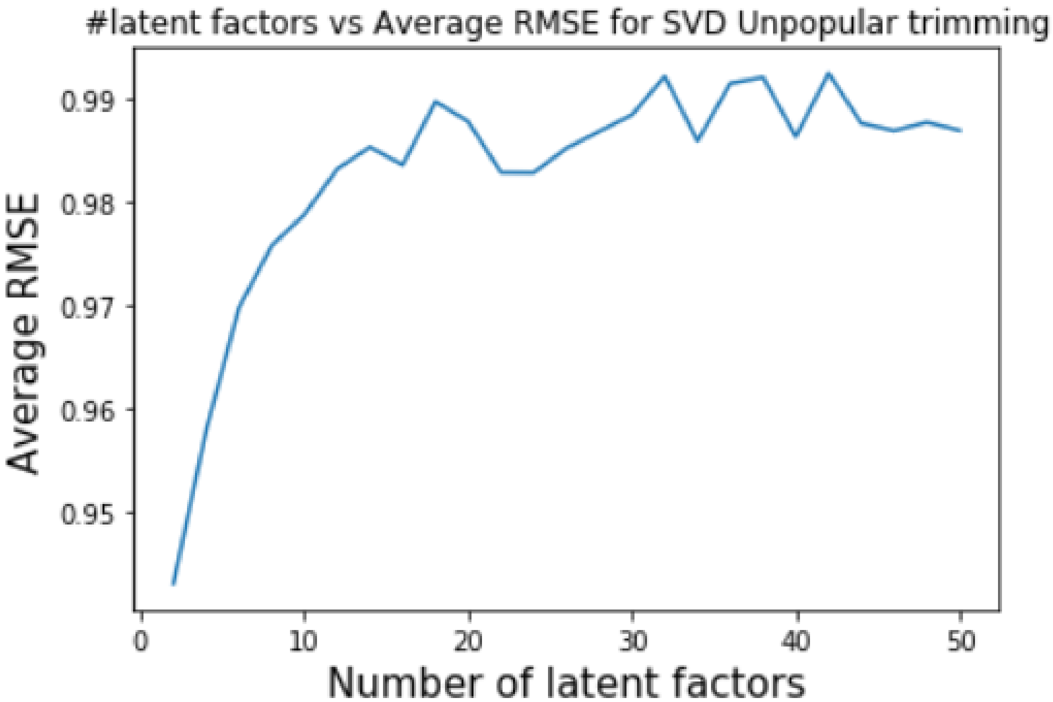

Figure 32: Average RMSE Vs Number of Latent Factors(k) - MF with bias - Unpopular movies

Evaluating MF with bias collaborative filtering on popular movie trimmed set

The minimum average RMSE in predicting the rating of a movie in the unpopular movie trimmed set is **0.943 for k = 2**.

4.28 Question 28

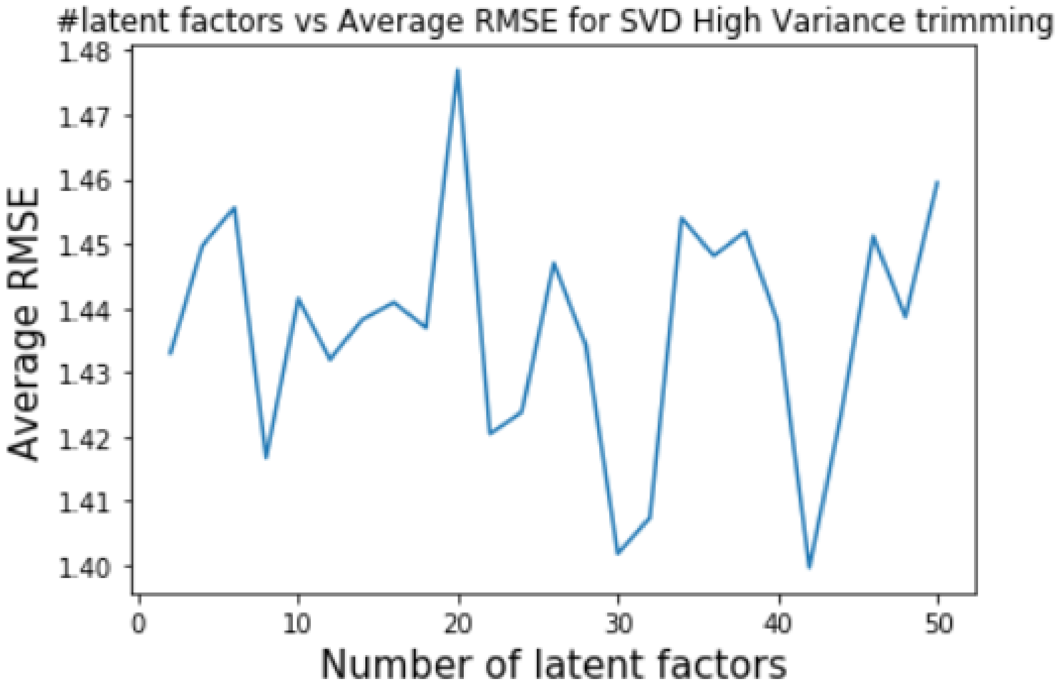

Figure 33: Average RMSE Vs Number of Latent Factors(k) - MF with bias - High variance trimming

Evaluating MF with bias collaborative filtering on high variance trimmed set

The minimum average RMSE in predicting the rating of a movie in the unpopular movie trimmed set is **1.394 for k = 42**.

4.29 Question 29

4.29.1 ROC curve with threshold = 2.5

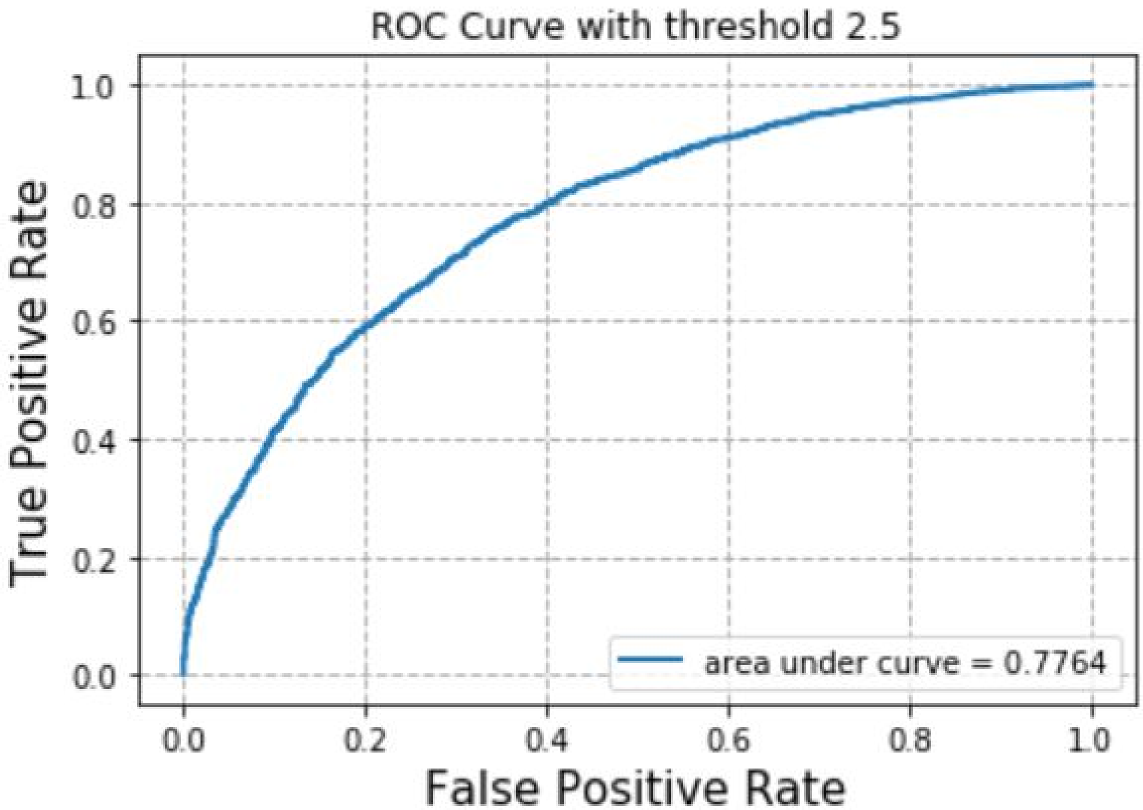

Figure 34: ROC Curve for MF with bias for threshold of 2.5

4.29.2 ROC curve with threshold = 3

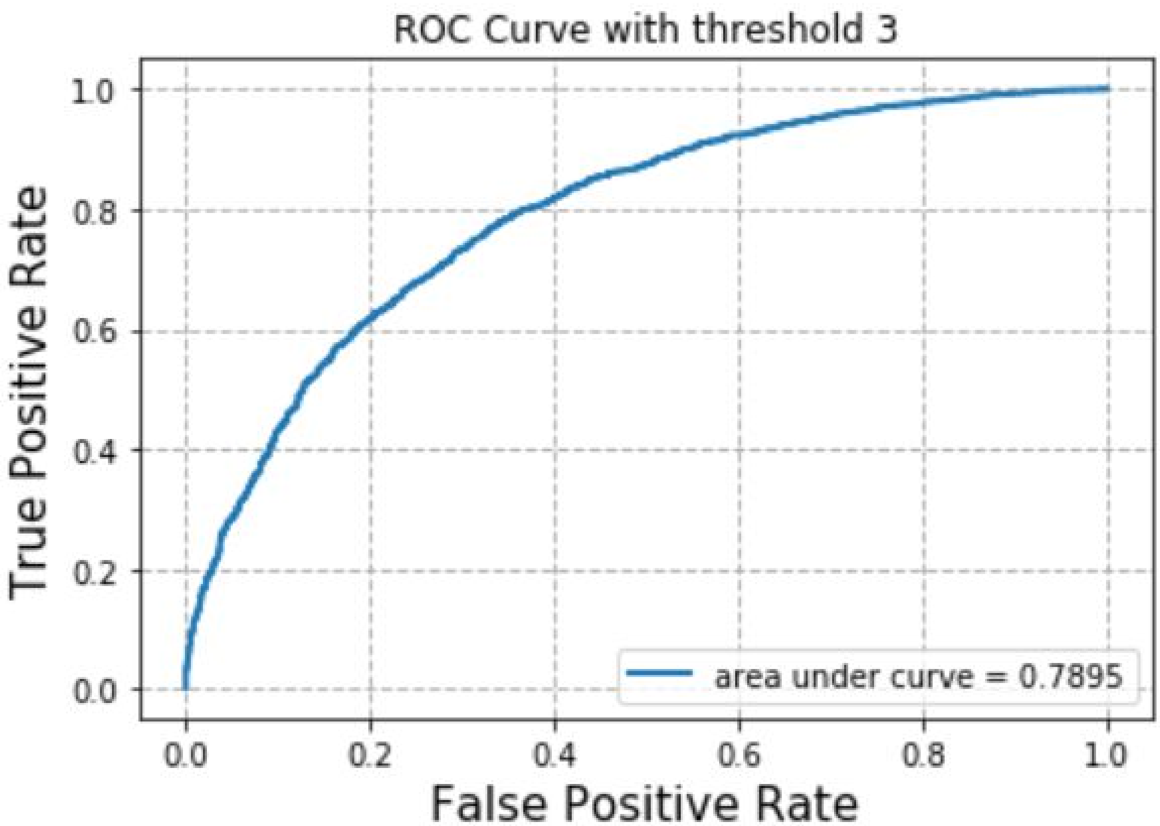

Figure 35: ROC Curve for MF with bias for threshold of 3

4.29.3 ROC curve with threshold = 3.5

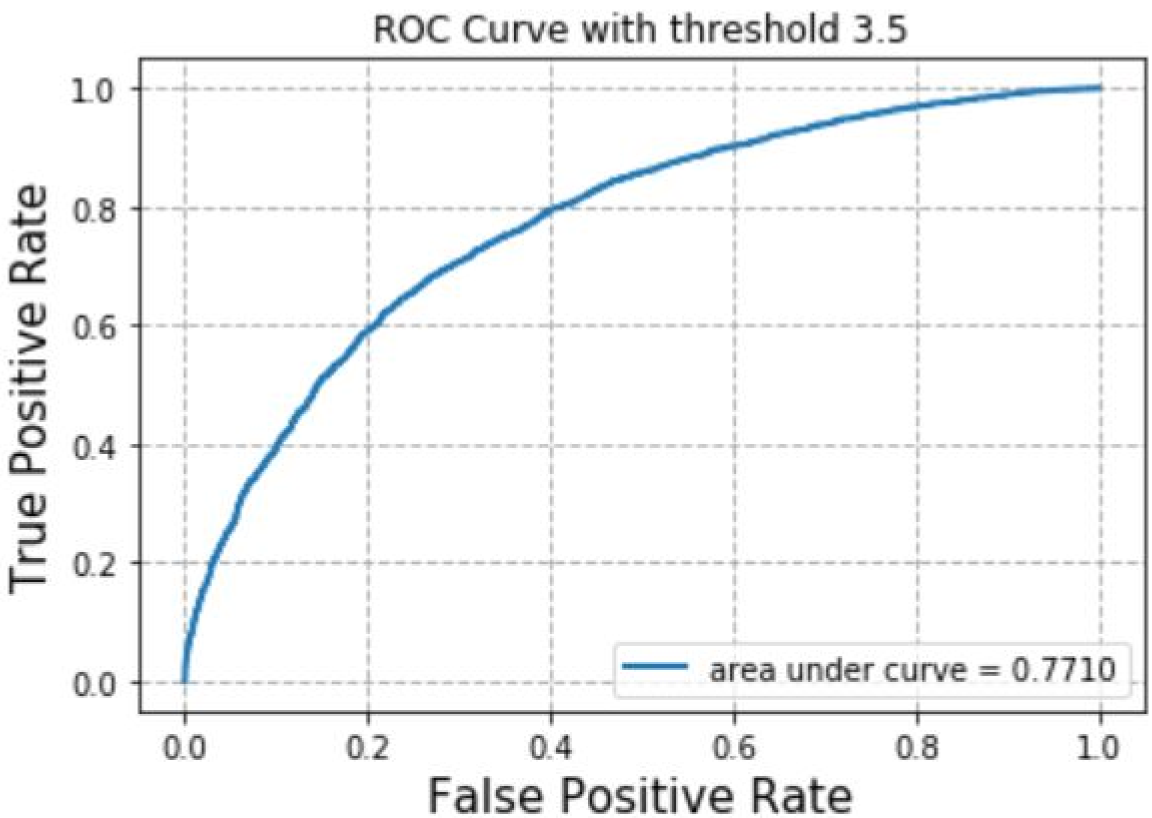

Figure 36: ROC Curve for MF with bias for threshold of 3.5

4.29.4 ROC curve with threshold = 4

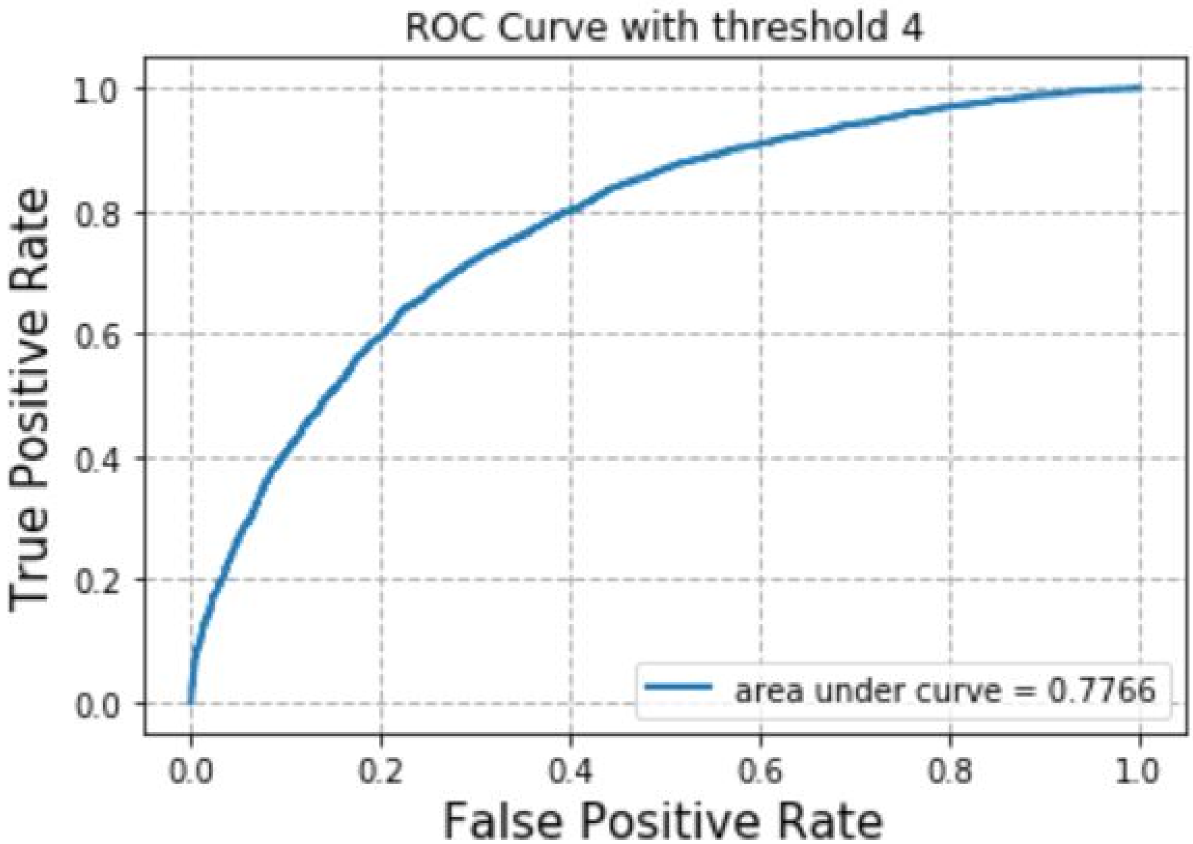

Figure 37: ROC Curve for MF with bias for threshold of 4

Threshold	Area under the curve
2.5	0.7764
3	0.7895
3.5	0.7710
4	0.7766

Interpretation There is just a marginal difference in performance for the different thresholds. By just comparing the area under the curve for the different thresholds, it can be seen that setting the threshold to be 3 produces the best results. Hence, 3 can be used as a threshold setting for all future problems and it will produce the best performance.

4.30 Question 30

Implementation The dataframe containing the ratings information was first grouped by the userID and the mean of the grouped data was stored in a list. A function was written which returned a list containing the ratings based on the mean ratings given by the user.

For each of the testsets, the predicted ratings were found by calling the function and the actual ratings were also taken in a list. The predicted ratings and the actual ratings were compared to determine the root mean squared errors.

The **average RMSE** for predicting the ratings of movies in the MovieLens dataset using **naive collaborative filter** with 10-fold cross validation is **0.9129**.

4.31 Question 31

Implementation For each of the testsets, the predicted ratings were found by calling the prediction function on the popular trimmed dataset and the actual ratings were also taken in a list. The predicted ratings and the actual ratings were compared to determine the root mean squared errors.
The **average RMSE** for predicting the ratings of movies in the popular movie trimmed set using **naive collaborative filter** with 10-fold cross validation is **0.8827**.

Interpretation As it was expected, the average RMSE for the popular trimmed is lower than normal since there is lesser variance for the ratings of the popular movies.

4.32 Question 32

Implementation For each of the testsets, the predicted ratings were found by calling the prediction function on the unpopular trimmed dataset and the actual ratings were also taken in a list. The predicted ratings and the actual ratings were compared to determine the root mean squared errors.

The **average RMSE** for predicting the ratings of movies in the unpopular movie trimmed set using **naive collaborative filter** with 10-fold cross validation is **0.9777**.

Interpretation The average RMSE for the unpopular trimmed is higher than normal since there is higher variance for the ratings of the unpopular movies due to the limited number of ratings.

4.33 Question 33

Implementation For each of the testsets, the predicted ratings were found by calling the prediction function on the high variance trimmed dataset and the actual ratings were also taken in a list. The predicted ratings and the actual ratings were compared to determine the root mean squared errors.

The **average RMSE** for predicting the ratings of movies in the high variance movie trimmed set using **naive collaborative filter** with 10-fold cross validation is **2.1144**.

Interpretation As expected, the average RMSE for the high variance trimmed data set is very large since there is a huge variance for the ratings of the high variance trimmed data set.

4.34 Question 34

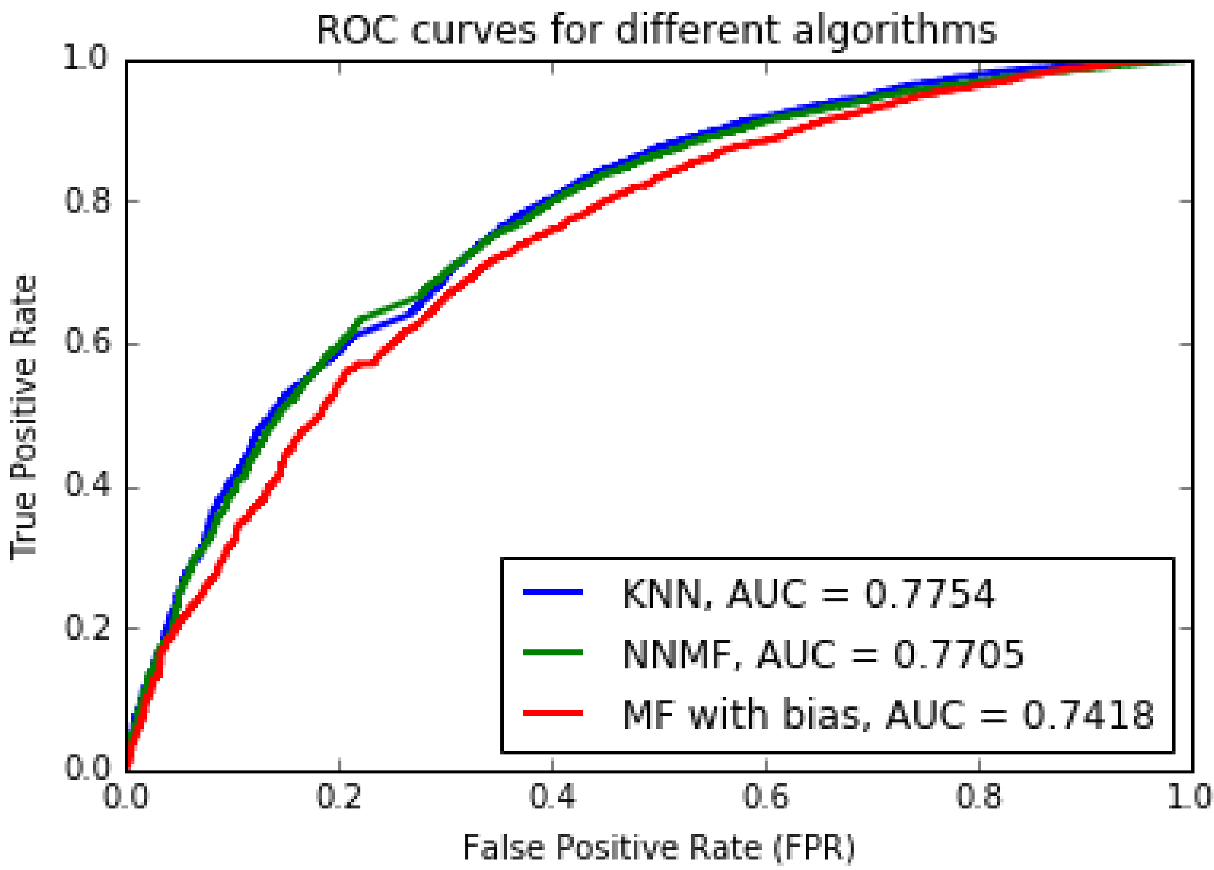

Figure 38: ROC curves for different algorithms

Interpretation As seen from the graph, there is just a marginal difference in performance between the different algorithms. MF with bias performs slightly poorer compared to the other two algorithms. The MF with bias algorithm varies drastically based on the different hyperparameters and this is just one of the results that was obtained for one of the hyper-parameter settings. Based on this graph, KNN and NMF perform better compared to MF with bias since they have a higher area under the curve for the ROC plots. Hence those two algorithms will be preferred for predicting the ratings of the movies.

4.35 Question 35

Precision In general, precision can be defined as the ratio of correctly predicted positive observations to the total predicted positive observations. In this specific case of recommendation system, precision can be understood as the fraction of recommendations that the user will actually like to the total number of recommendations.

Recall In general, recall can be defined as the ratio of correctly predicted positive observations to all the observations in the actual class. In this specific case of recommendation system, recall can be understood as the fraction of recommendations that the user will actually like to the total movies that the user likes.

4.36 Question 36

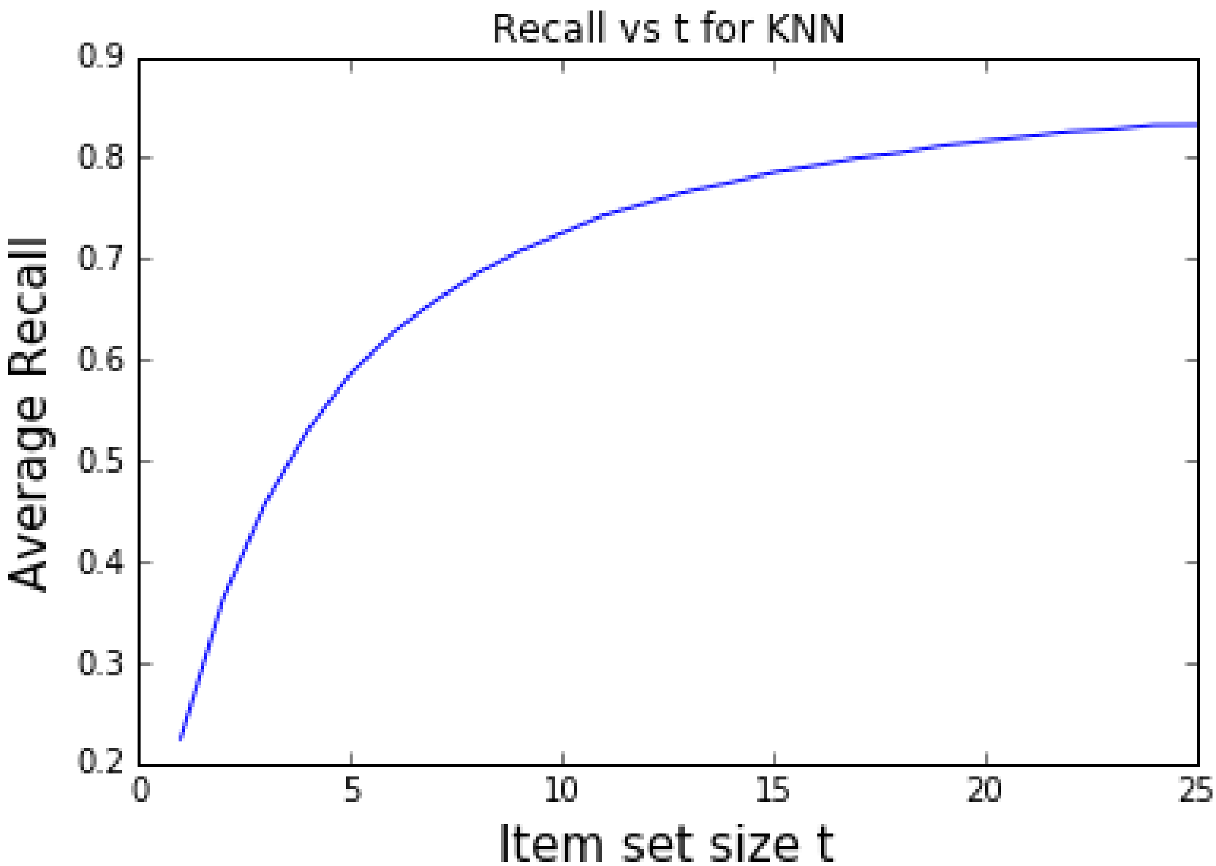

Figure 39: Average Recall Vs Item set size t for k-NN

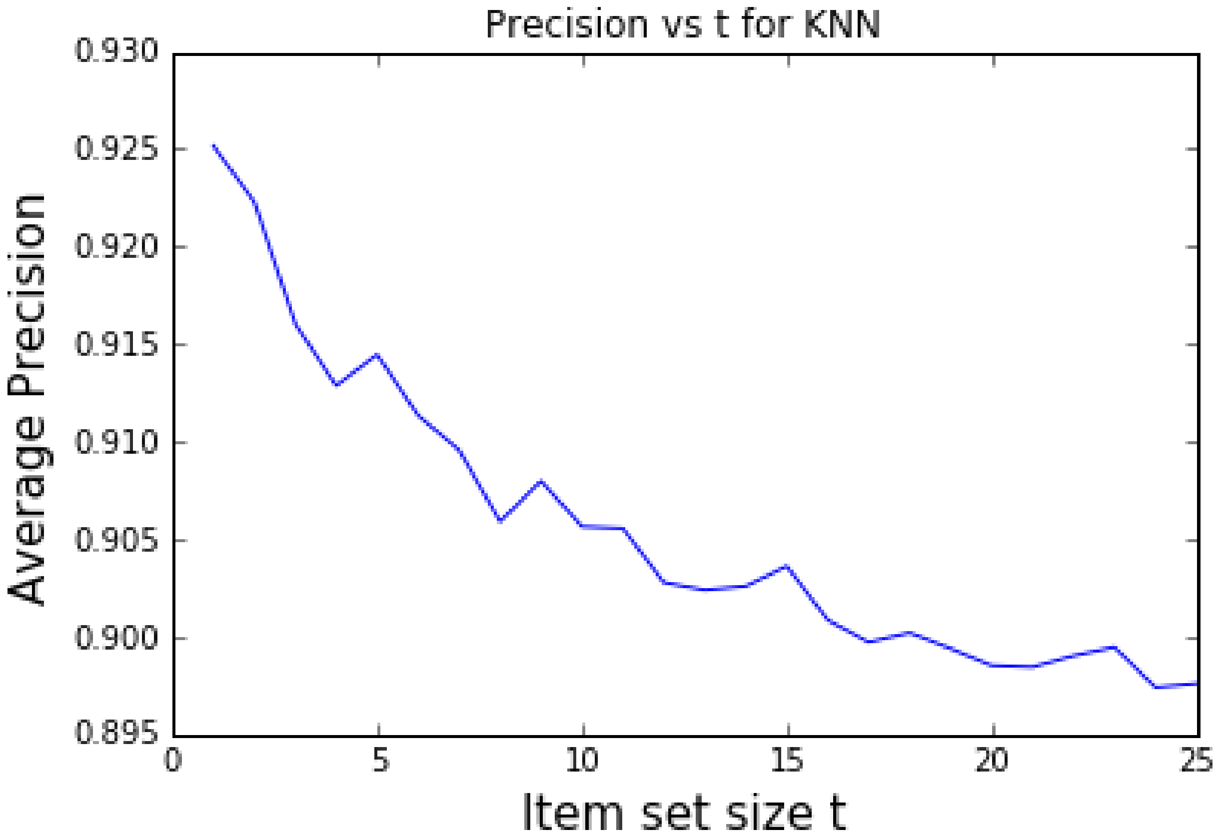

Figure 40: Average Precision Vs Item set size t for k-NN

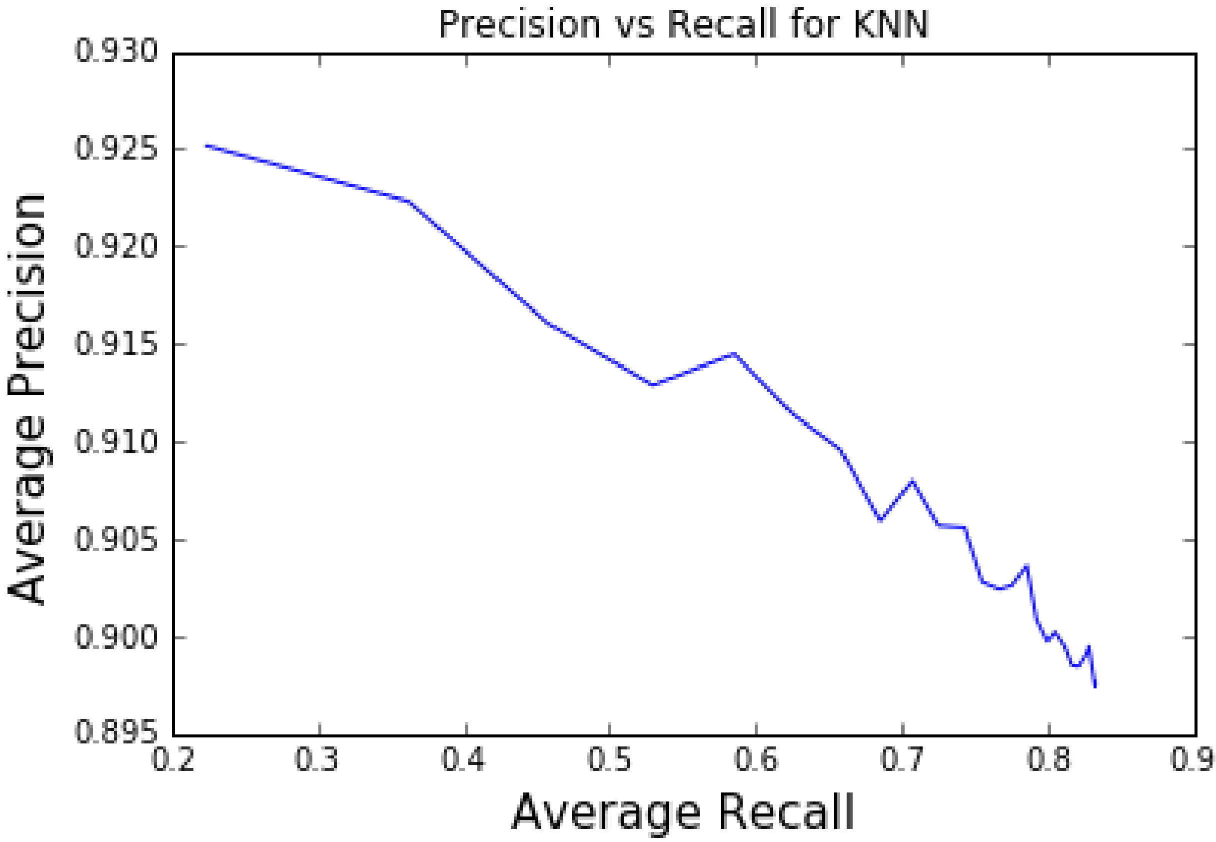

Figure 41: Average Precision Vs Average Recall for k-NN

Interpretation The first graph shows the change in recall as the recommendation size t increases. It shows that as the recommendation size increases, the recall increases and the increase is exponential. The second graph tells us that precision reduces as the item set size increases but the decrease in precision is small. The third graph is the precision vs recall curve which shows the trade-off between precision and recall and helps us to determine the ideal item set size that gives us the best achievable trade-off between precision and recall.

4.37 Question 37

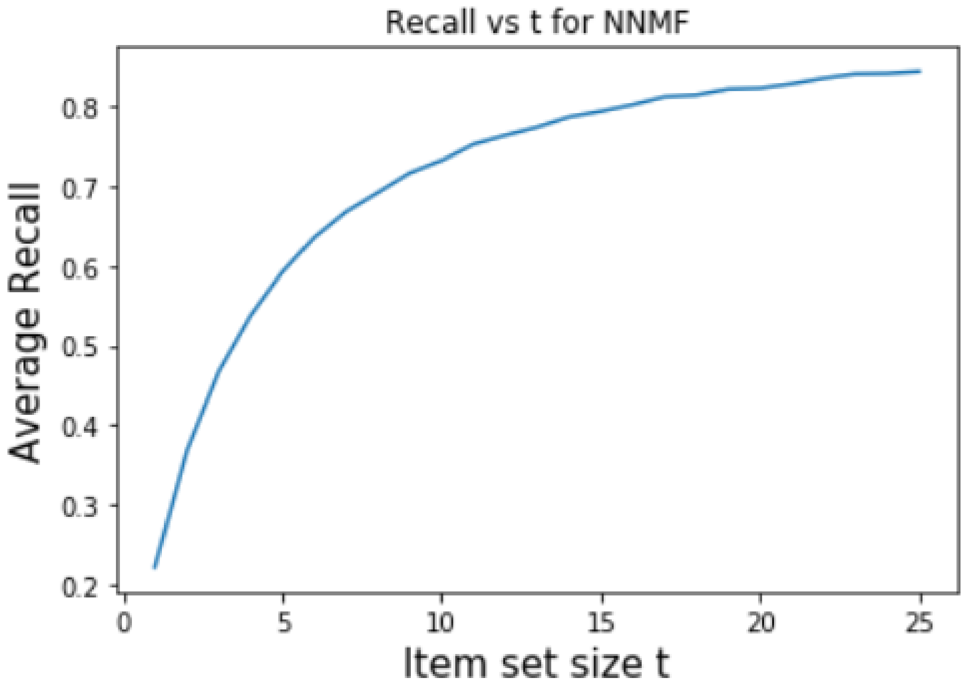

Figure 42: Average Recall Vs Item set size t for NNMF

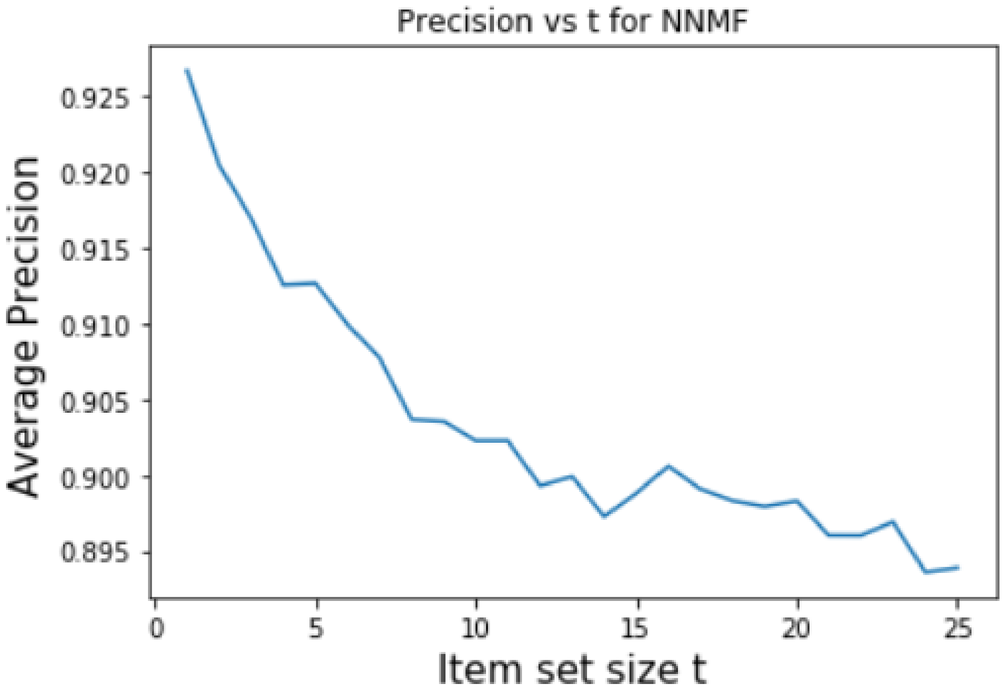

Figure 43: Average Precision Vs Item set size t for NNMF

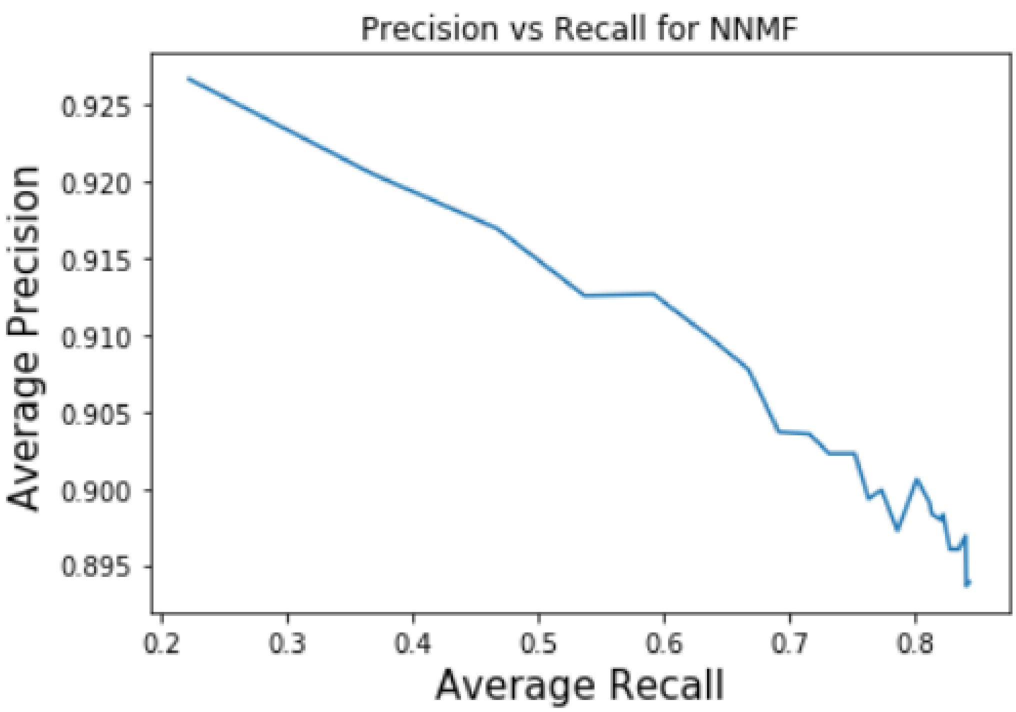

Figure 44: Average Precision Vs Average Recall for NNMF

Interpretation The first graph shows the change in recall as the recommendation size t increases. It shows that as the recommendation size increases, the recall increases and the increase is exponential. The second graph tells us that precision reduces as the item set size increases but the decrease in precision is small. The range of variation of recall is just 0.3. The third graph is the precision vs recall curve which shows the trade-off between precision and recall and helps us to determine the ideal item set size that gives us the best achievable trade-off between precision and recall.

4.38 Question 38

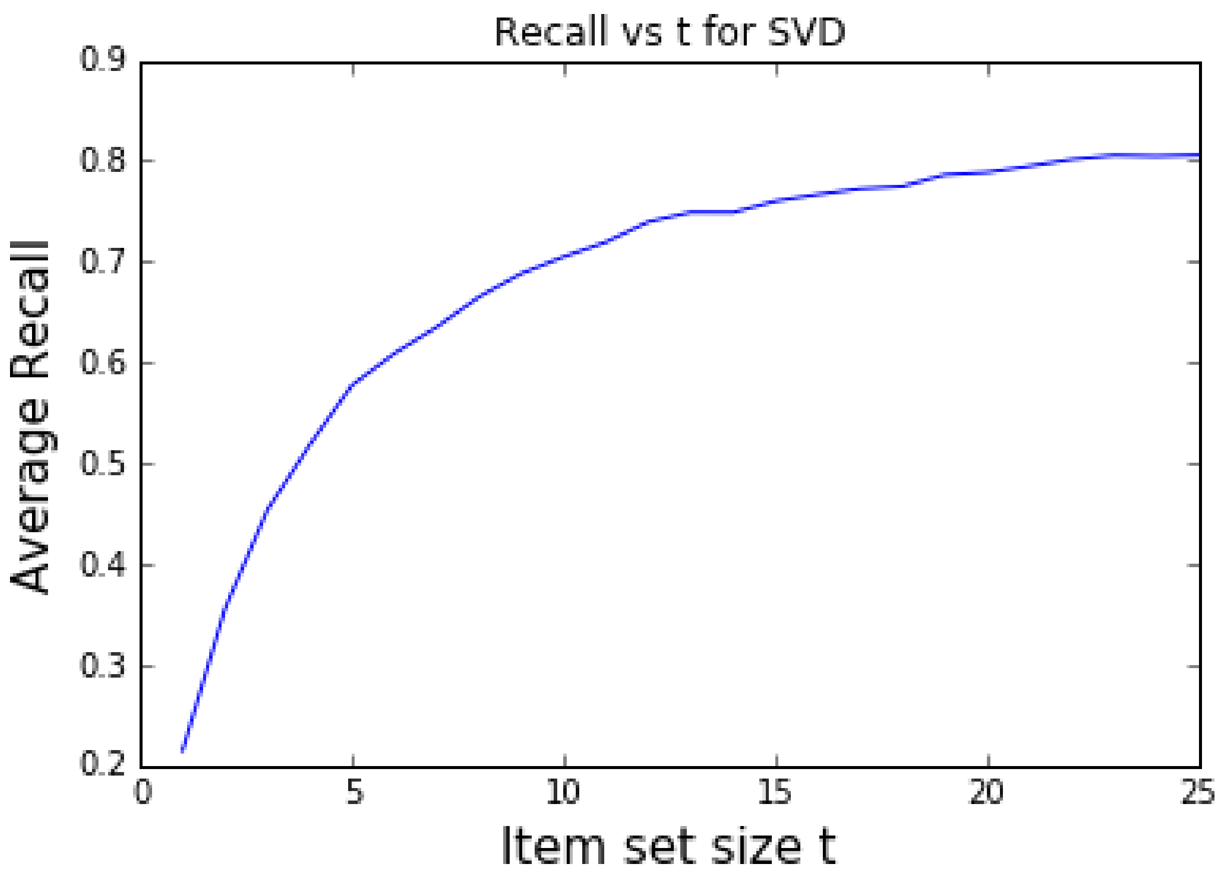

Figure 45: Average Recall Vs Item set size t for MF with bias

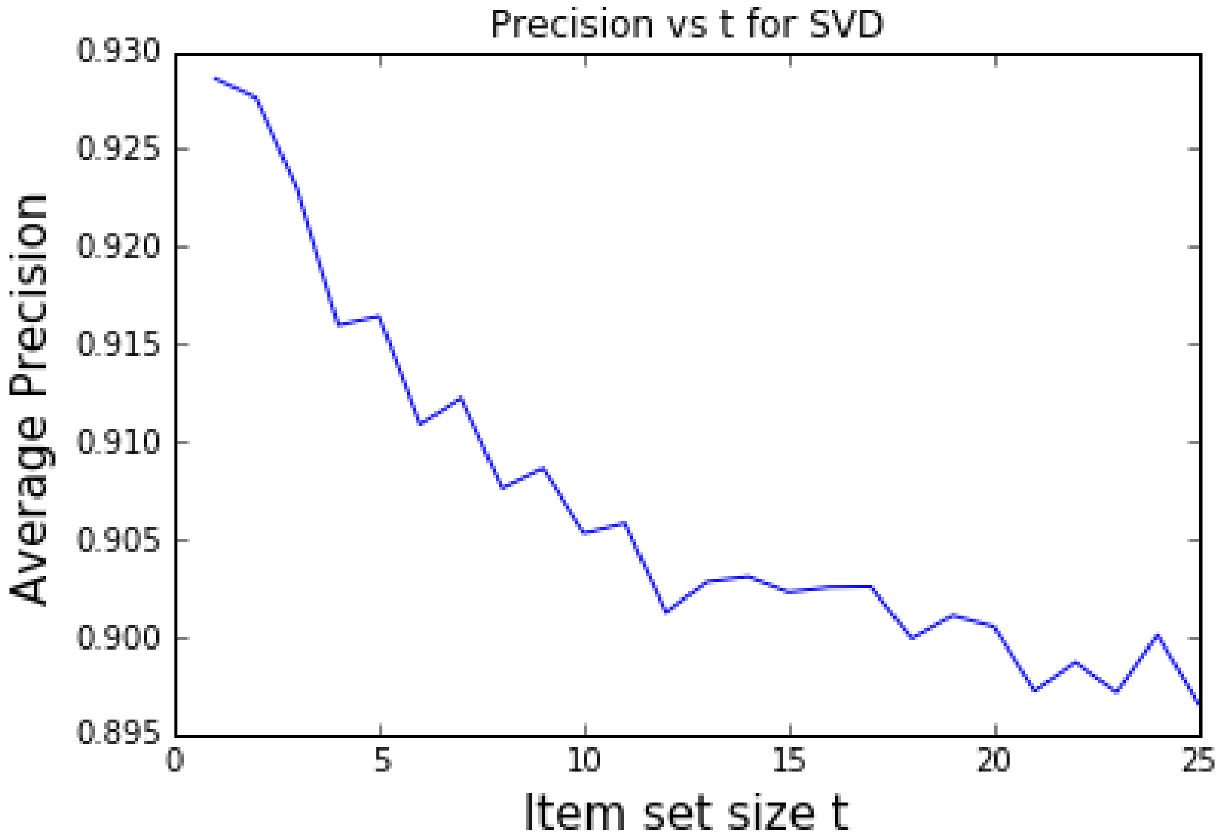

Figure 46: Average Precision Vs Item set size t for MF with bias

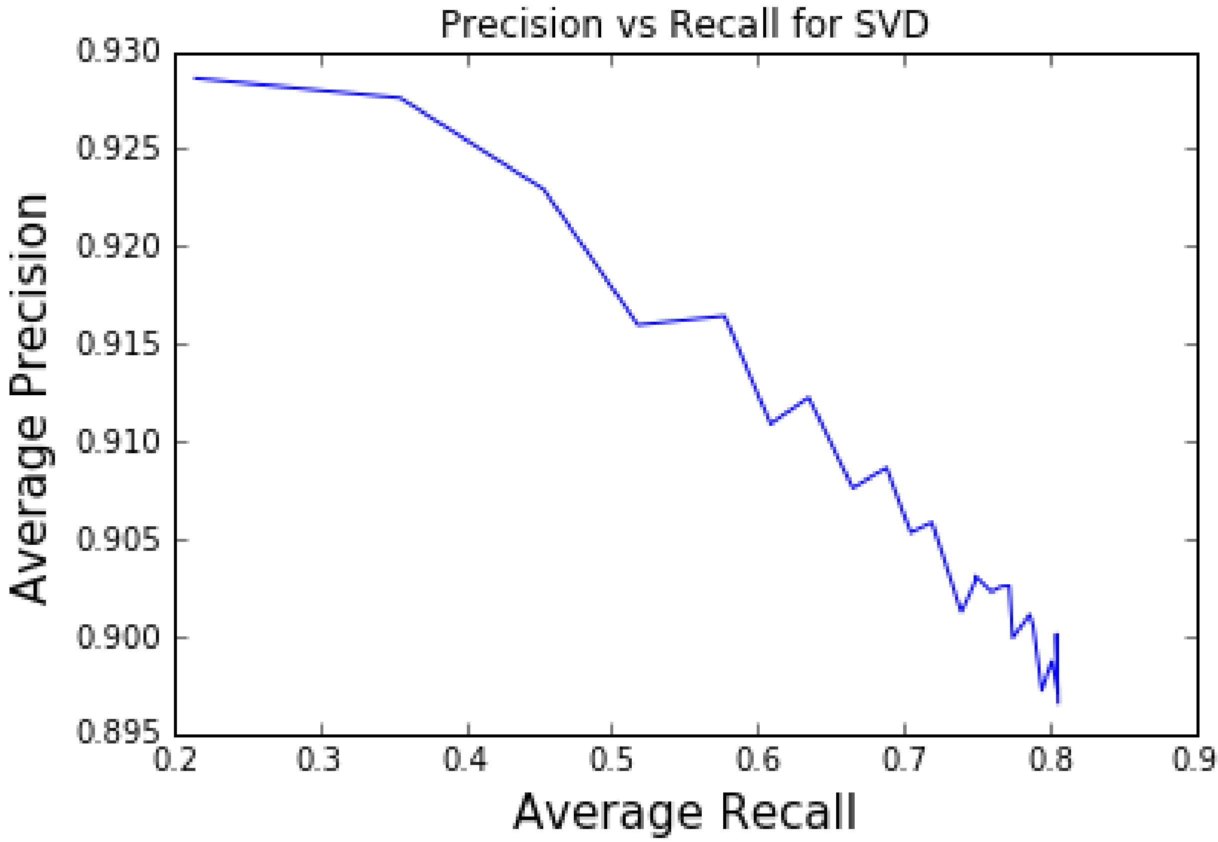

Figure 47: Average Precision Vs Average Recall for MF with bias

Interpretation The first graph shows the change in recall as the recommendation size t increases. It shows that as the recommendation size increases, the recall increases and the increase is exponential. The second graph tells us that precision reduces as the item set size increases but the decrease in precision is small. Compared to the other two algorithms, the precision curve for MF with bias is noisier. A reason for this is that there are a number of hyperparameters for us to tune in MF with bias. The curves were far more noisier and we were able to obtain a less noisier version by varying the hyperparameters and achieve this result. The third graph is the precision vs recall curve which shows the trade-off between precision and recall and helps us to determine the ideal item set size that gives us the best achievable trade-off between precision and recall.

4.39 Question 39

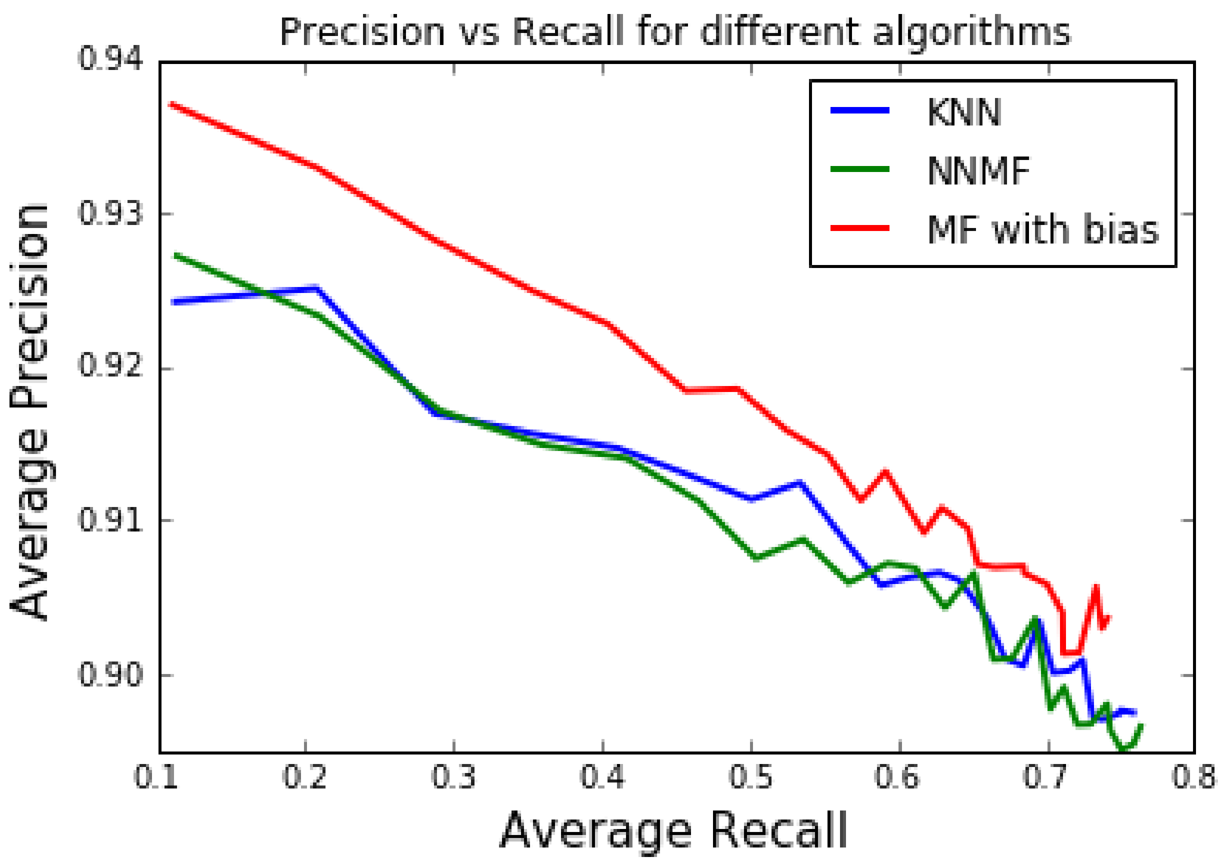

Figure 48: Precision-Recall curves for different algorithms

Interpretation The plot shows the precision recall curves for different algorithms. It helps us to compare the effectiveness of the recommendations of the different algorithms. As seen from the graph, the curve for MF with bias is clearly above the other two curves which produce similar results. This tells us that MF with bias will help us to achieve a better recommendation compared to KNN and NMF. Since KNN and NMF produce curves which are similar, they will perform equally in terms of recommendation results. For a given average recall, an algorithm using MF with bias will have a higher precision compared to the other algorithms. The other two algorithms are closer to the baseline result. This shows us that MF with bias will be a better algorithm for recommending movies compared to the other two.

5 Conclusion

Through this project, we have implemented several algorithms for creating a recommendation system. The two major tasks of predicting user ratings for unseen movies and creating top recommendations for users were done. For both these tasks, different collaborative filtering techniques were implemented and their differences were compared. Precision-recall curves and ROC curves were used to determine the performance.