EE-219 Project 3

COLLABORATIVE FILTERING

Devanshi Patel	504945601	devanshipatel@cs.ucla.edu
Ekta Malkan	504945210	emalkan@cs.ucla.edu
Pratiksha Kap	704944610	pratikshakap@cs.ucla.edu
Sneha Shankar	404946026	snehashankar@cs.ucla.edu

INTRODUCTION

Recommender Systems are used in modern day applications like Amazon, Netflix, etc. to identify patterns or similarities between a user and the products . They use this data to recommend similar products to the user. This helps the companies to make more profits and helps the users by trying out recommended products that they wouldn't have found easily by searching.

There are two types of Recommender Systems algorithms- Content based and Collaborative Filtering. Content-Based algorithms rely on the features and attributes of the product for recommendation, whereas collaborative filtering relies on identifying user-user or item-item similarity and recommending those items to the user based on likes of other users who are similar to the current user.

In this project, we study collaborative filtering methods namely neighborhood based and model-based collaborative filtering.

Dataset

We have used the MovieLens Dataset for this project titled "ml-latest-small". It contains 100004 ratings across 9125 movies by 671 users.

The "ratings" file in this dataset contains information about users, movie, ratings and the timestamp. The "movie" file in the dataset contains information about movie id, title and genres. However, in this project we will be using just the movie ratings and not the movie genre information.

Question 1

Compute the sparsity of the movie rating dataset, where sparsity is defined by equation below :

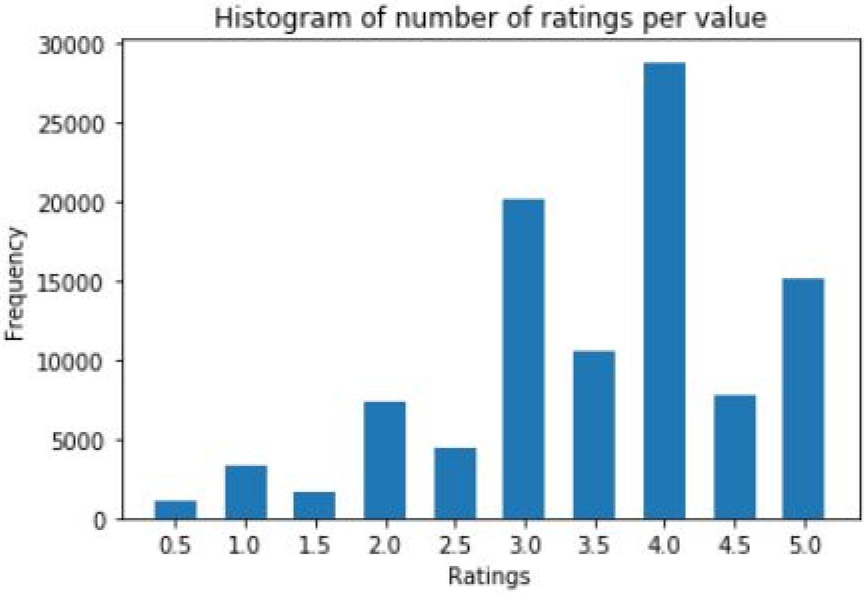
Sparsity = 1- (Total number of available ratings / Total number of possible ratings)

Answer: 1- 0.0163 = 0.9837

The sparsity value determines the proportion of missing rating values for users and movie combinations. A high sparsity value of **0.9837** indicates that the ratings matrix has lesser number of values populated. This is expected as the number of movies is large and every user might have watched just a limited number of movies.

Question 2

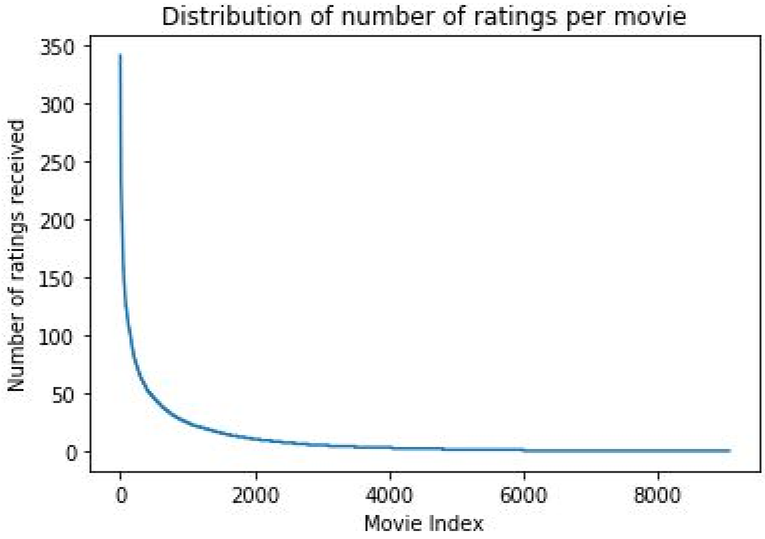
Plot a histogram showing the frequency of the rating values. To be specific, bin the rating values into intervals of width 0.5 and use the binned rating values as the horizontal axis. Count the number of entries in the ratings matrix R with rating values in the binned intervals and use this count as the vertical axis. Briefly comment on the shape of the histogram.



Observations:
Most movies have been rated 4.0 by the users and the second largest is rating 3.0. The number of movies with ratings less than 3.0 are very few.

Question 3

Plot the distribution of ratings among movies. To be specific, the X-axis should be the movie index ordered by decreasing frequency and the Y -axis should be the number of ratings the movie has received.

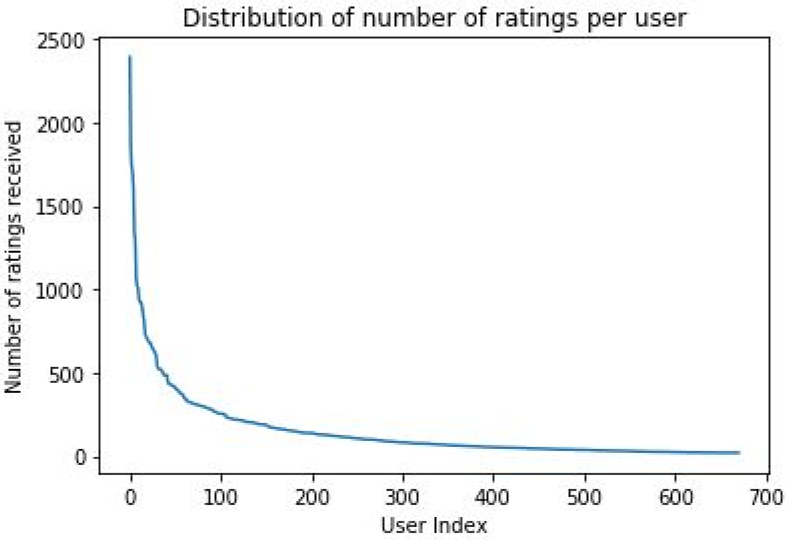


Observations:

The above distribution indicates that there are very few movies that have a lot of ratings. These might be the famous and legendary movies. For majority of the ordinary movies, the number of ratings is very less which is depicted by the exponential nature of the plot above. The recently added movies may have no ratings at all and this is the reason for the sparsity of data and its skewed distribution.

Question 4

Plot the distribution of ratings among users. To be specific, the X-axis should be the user index ordered by decreasing frequency and the Y -axis should be the number of movies the user have rated.



Observations:

The above distribution indicates that there are few users who have rated a lot of movies. These users are mainly the movie buffs or movie critics. For majority of the common users, the number of ratings is very less and hence the exponential shape of the plot. This indicates that the distribution of ratings is skewed and the matrix is sparse. The fewer ratings may be due to the reason that the users newly added to the system may have no ratings at all.

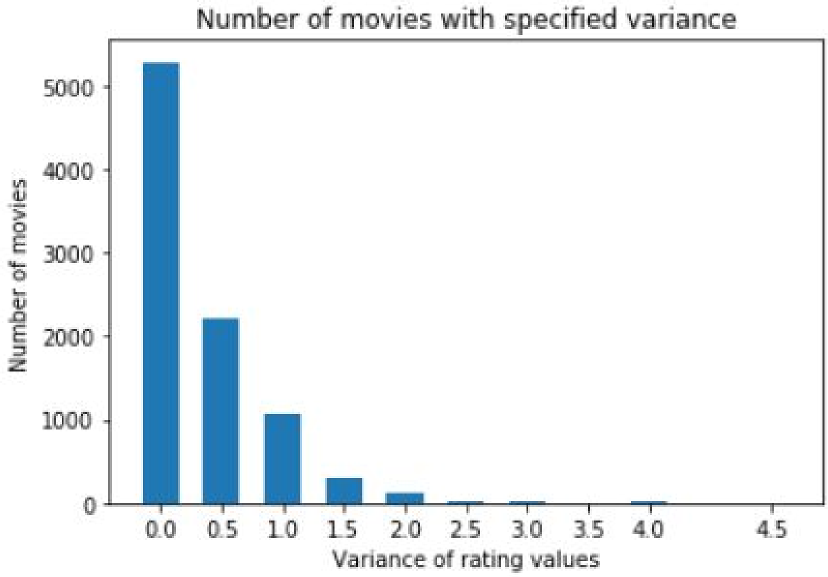
Question 5

Explain the salient features of the distribution found in question 3 and their implications for the recommendation process.

Answer: The distribution in question 3 indicates that only a fraction of (popular) items are rated more frequently but most other items are rated less frequently. This kind of distribution is termed as long tail distribution. The traditional collaborative filtering methods are not very effective in dealing with this type of sparse and skewed distribution. It is difficult to provide robust rating predictions. This has an effect on diversity of recommendations and the user might get bored because of being recommended same set of popular items.

Question 6

Compute the variance of the rating values received by each movie. Then, bin the variance values into intervals of width 0.5 and use the binned variance values as the horizontal axis. Count the number of movies with variance values in the binned intervals and use this count as the vertical axis. Briefly comment on the shape of the histogram.



Observations:
The above histogram indicates that there are very few movies whose variance in ratings is high. A high value of variance exists for movies that are controversial in nature. For a majority of movies, the variance falls in the 0.0 bucket which means that the variance is less than 0.5. The shape of the plot indicates that the variance in the values of ratings decreases exponentially.

Neighborhood-based collaborative Filtering

KNN based Collaborative filtering is a user-based filter model which relies on the fact that similar users portray similar rating tendencies, and similar items end up getting similar ratings.

The goal of this task is to predict the rating that a user will give to a movie based on the ratings of other similar users on the same movie. This predicted rating is then evaluated against the ground truth rating given by the user and the deviation(error) is calculated. MAE and RMSE are the two most common metrics used for this calculation.

MAE : Mean Absolute Error

Mean Absolute Error measures the average deviation (error) in the predicted rating vs. the true rating. Let u(c, s) be the true ratings, and u^p(c, s) be the ratings predicted by a recommender system. Let W = {(c, s)} be a set of user-item pairs for which the recommender system made predictions. Then, the mean absolute error, is defined as follows:

$$|\bar{E}| = \frac{\sum_{(c,s)\in W} |u^{p}(c,s) - u(c,s)|}{|W|}$$

RMSE : Root Mean Squared Error

It represents a sample standard deviation of the differences between the estimated values and the actual values. It gives a measure of accuracy of the model. The lower the RMSE, the better is the accuracy of the model.

$$|\sqrt{\bar{E}^2}| = \sqrt{|\bar{E}^2|} = \sqrt{\frac{\sum_{(c,s)\in W} (u^{p}(c,s) - u(c,s))^2}{|W|}}$$

Question 7

Write down the formula for μ_u in terms of I_u and r_{uk}.

Answer:

$$\mu_u = \frac{\sum_{k \epsilon I_u} r_{uk}}{|I_u|}$$

Question 8

In plain words, explain the meaning of $I_u \cap I_v$. Can $I_u \cap I_v = \phi$.

Answer: I_u is the set of indices for which user u has provided ratings and I_v is the set of Indices for which user v has provided ratings. Hence, the meaning of $I_u \cap I_v$ is the set of Indices for which both the users have provided ratings.

Yes, it is highly possible that for a pair of users u and v, $I_u \cap I_v$ is an empty set ϕ. This is because the ratings matrix available with us is sparse (Sparsity=98%). Hence for a large pair of users, movies observed by them would be in totally different sets and hence the common set of movies would be empty.

Question 9

Can you explain the reason behind mean-centering the raw ratings ($r_{vj} - \mu_v$) in the prediction function?

Answer: Suppose we have many users in our dataset, some of them may rate all the movies equally high and some may rate most movies poorly.

The formula for predicting user u’s rating for a movie when rating of user is available (without mean centering) will be :

$$r_{uj=}\mu_u + \frac{\sum_{v \epsilon P_u} Pearson\ (u,v) r_{vj}}{\sum_{v \epsilon P_u} |Pearson\ (u,v)|}$$

In this case, when we are calculating rating for a movie for user u, if there exists a user v who always rates all movies highly, the rating of the movie would increase by (Pearson coefficient between users u and v) * the rating of user v (i.e. 5), which would result in a higher rating for the movie than what should have been.

Similarly, if a user v always gives movies very low rating, then adding his rating to the equation would result in lower than expected predicted rating for the movie.

To nullify the effect of such outliers in the dataset, each rating given by a user is subtracted from the mean rating of that user. The mean rating for a user is summation of his ratings for all movies divided by the number of movies he has rated.

Now, after mean-centering the new formula becomes:

$$r_{uj=}\mu_u + \frac{\sum_{v \epsilon P_u} Pearson\,(u,v)(r_{vj} - \mu_v)}{\sum_{v \epsilon P_u} |Pearson\,(u,v)|}$$

If the user u's rating is now calculated with respect to user v who gives a high rating to all the movies, the mean-centering would result in *(r_{vj} - μ_v) i.e.* 5-5 =0 value. This value when added to the formula would now result in a pure predicted estimate of user u's rating for movie j.

Hence, mean-centering the raw ratings in the prediction function is useful to remove the impact of outlier data.

We now find K-Nearest Neighbors (set Pu) of user u, who have the highest Pearson correlation coefficient with user u.

Question 10

Design a k-NN collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate it's performance using 10-fold cross validation. Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis).

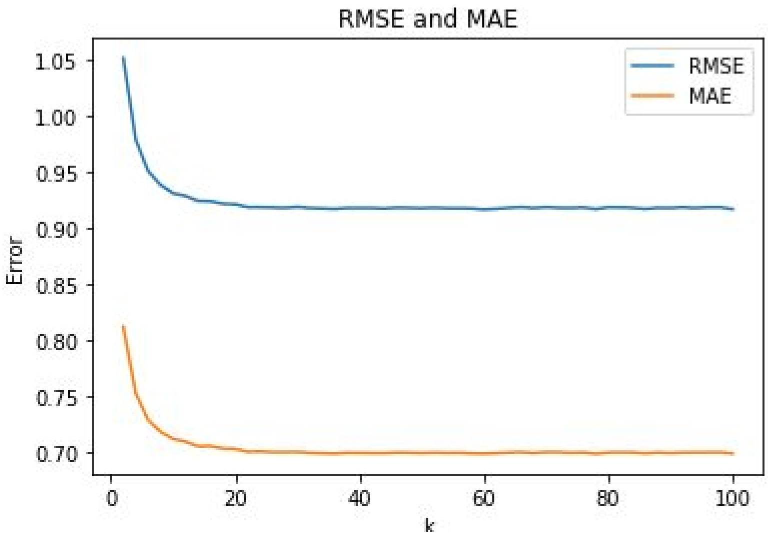
For this question, we implemented KNN collaborative filter using pythons' Surprise Library on the MovieLens Dataset.

Surprise is a python scikit for building and analyzing recommender systems.

We analyze our KNN collaborative filter using 10-Fold Cross validation, which splits the data into 10 slices by using 9 slices for training and one for testing. It then repeats this process, every time taking a new slice for testing and the rest for training the data.

We also iterated for getting the optimal number of neighbor's k such that the Root Mean Squared Error and Mean Averaged Error is minimized. Before jumping to the tasks, let us understand RMSE and MAE in more detail.

We have plotted the RMSE and MAE against k and analyzed the graphs as shown below:



Observations:

- For lesser number of neighbors i.e. k, both the RMSE and MAE is high. (RMSE=1.04 and MAE=0.80)
- As k increases, the error decreases up to a certain point, after which the error stabilizes. (at k=22)
- Any increase in k after this point results in a constant error. This is the optimal number of neighbors for our KNN Collaborative filter. (RMSE stays constant at 0.91 and MAE at 0.69)

Question 11

Use the plot from question 10, to find a 'minimum k'. Note: The term 'minimum k' in this context means that increasing k above the minimum value would not result in a significant decrease in average RMSE or average MAE. If you get the plot correct, then 'minimum k' would correspond to the k value for which average RMSE and average MAE converges to a steady-state value. Please report the steady state values of average RMSE and average MAE.

Answer: Minimum **k= 22**

As we can see from the graphs below, at k=22, the RMSE and MAE plots show steady values.

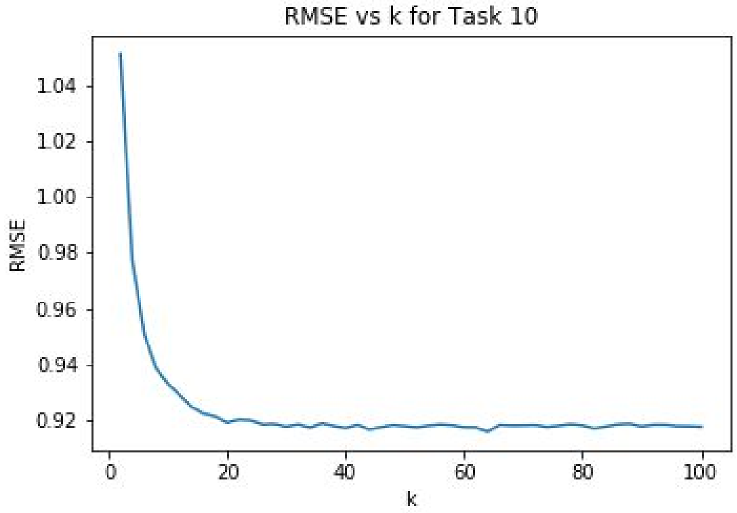
As observed from the code output, the values of mean RMSE and mean MAE were high initially.

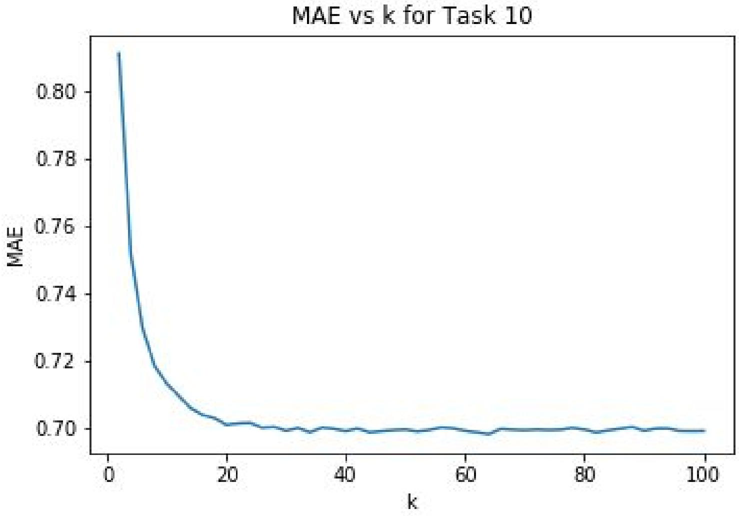
For k =2, mean RMSE was 1.05 and mean MAE was 0.81. Then as k increased to 2,4, and so on, mean RMSE decreased to 0.9759, 0.9709 and mean MAE dipped to 0.7507,0.7296 and so on. It is clear from the plot above, this steep decrease continues till k=16, when the line begins to curve. The values of RMSE and MAE are still decreasing but the change rate is lesser than earlier.

Finally, at k=22, the values begin to steady as shown in the graphs.
The Steady State values of mean RMSE and mean MAE are:

Min k	**22**
Steady state mean RMSE	**0.9178**
Steady state mean MAE	**0.7009**

This can be verified from the below plots for mean RMSE against k and mean MAE and k as shown below:





Thus, we have found the minimum k in this task that results in steady state values for average RMSE and average MAE.

So far we have used the entire testset for collaborative filtering. We now evaluate the performance of KNN for a trimmed testset by popular movies, unpopular movies, and movies with high variance in ratings (i.e. controversial movies).

We find minimum average RMSE, min k, steady state RMSE and steady state MAE for each of these tasks:

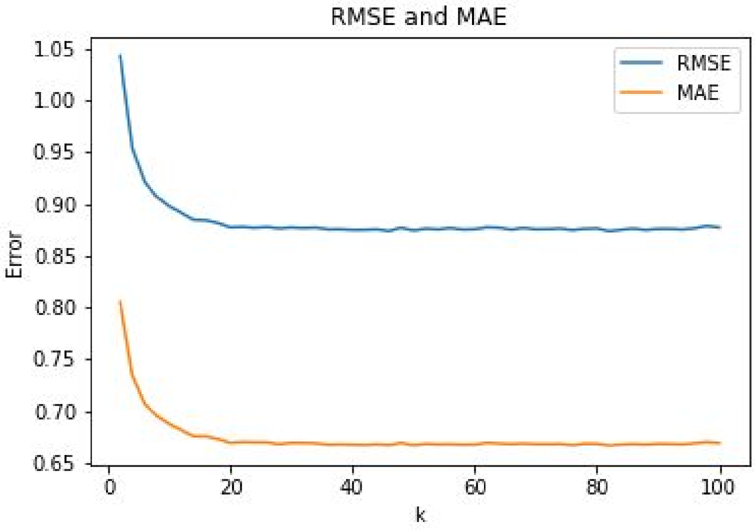
Question 12

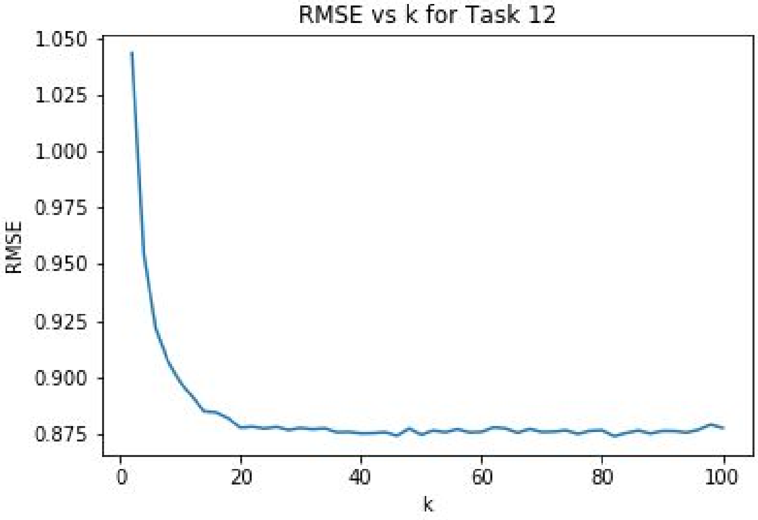
Design a k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate it's performance using 10-fold cross validation. Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

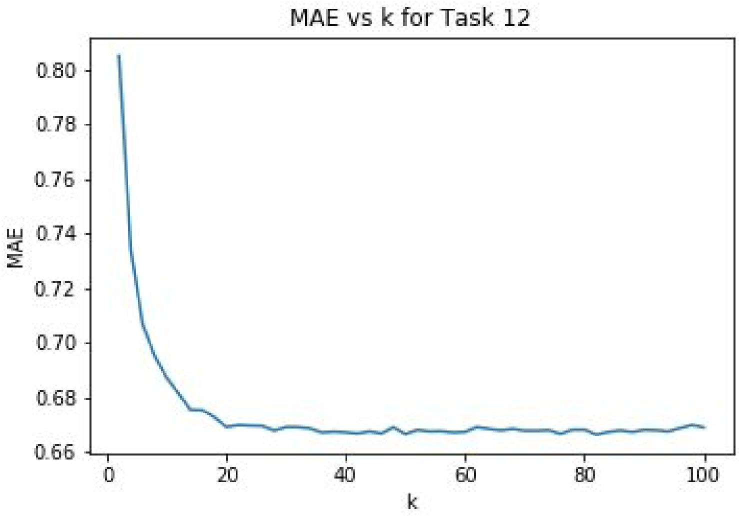
Answer:

In this task, for each fold of 10-Fold cross validation, we trim our test set for retaining popular movies only. Popular movies are movies which have received more than 2 ratings.

The average RMSE and average MAE obtained for different k values is as follows:







Minimum average RMSE: 0.8726

As we can see from the graphs above, the graphs for popular trimmed testset is almost similar to the untrimmed testset in the previous task. Digging deeper, let us look at the min k values and steady state mean RMSE and mean MAE values:

Min k	**22**
Steady state mean RMSE	**0.87**
Steady state mean MAE	**0.66**

As we can see from the above table, the minimum value of k at which the RMSE and MAE errors stabilize is similar to the untrimmed testset. However the steady state error values are less by 4-5% than that for untrimmed testset (0.91 vs 0.87 and 0.70 vs 0.66).

Why is this so? The reason may be that most of the movies in our dataset have received a rating of more than 2 (See histogram plot in question 2 for reference). Hence the popular trimmed testset is very similar to the untrimmed testset, and hence the plots and min k values are similar.

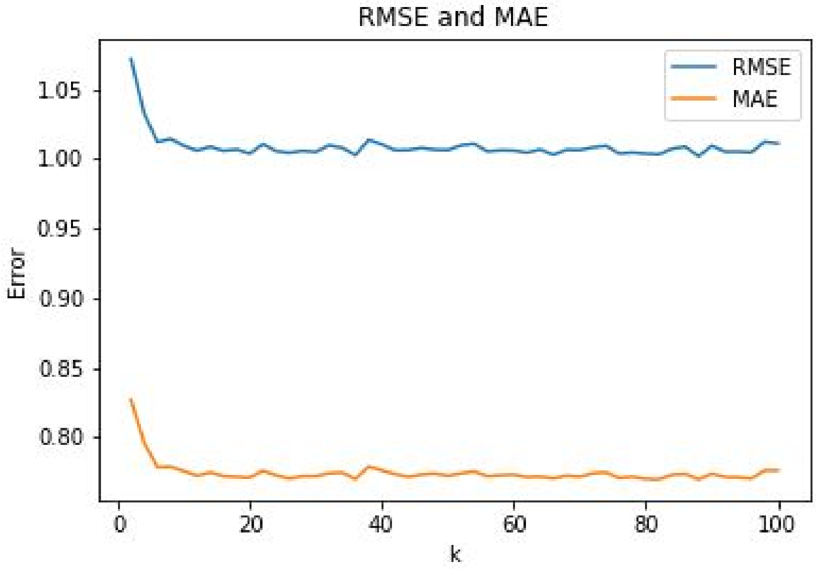
What would be the benefit of trimming our testset for popular movies only?

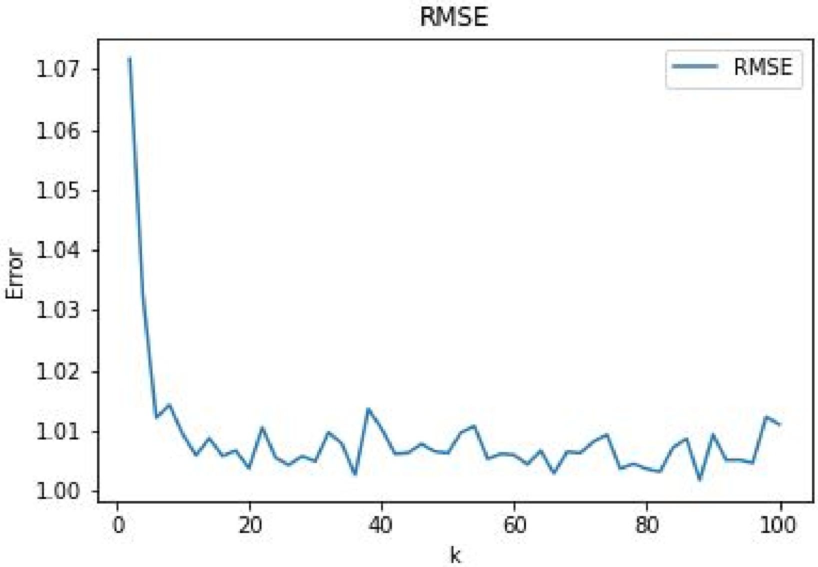
By trimming with popular movies, **we have reduced our Root Mean Squared error and Mean absolute error by 4-5%**. This results in better prediction of ratings for movies for which the rating is unknown based on other similar movies. One intuition here is that popular movies are observed by more people, and most of the time the content is equally well-appreciated.

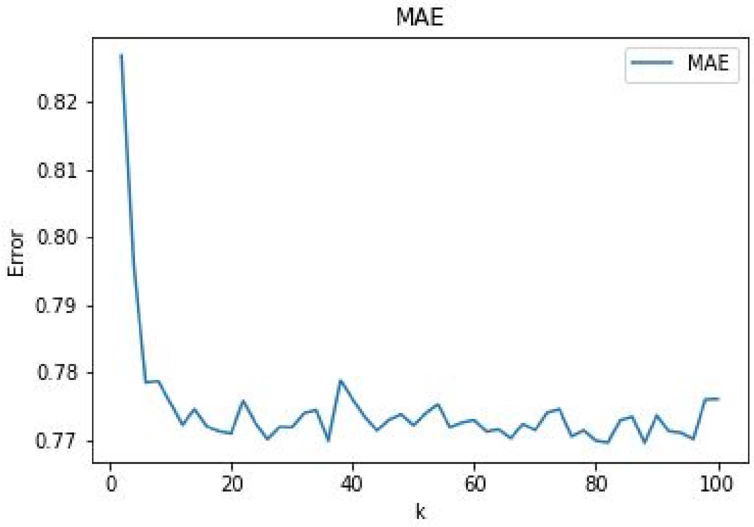
Question 13

Design a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate it's performance using 10-fold cross validation. Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

In this task, we trim our test set for unpopular movies. Unpopular movies are those movies that have received less than or equal to 2 ratings. The below plots were obtained for average RMSE and average MAE vs k, after performing 10 fold cross validation.







Minimum Average RMSE : 1.0005

We observe that the curves are not as smooth as obtained in the previous tasks. Why is this so? If we observe the histogram in Question 2, we realize that the number of movies with 2 or less ratings is far less as compared to the number of movies with higher than 2 ratings. Hence, as the data is very sparse, the curves obtained are not so smooth. On observing the steady state values :

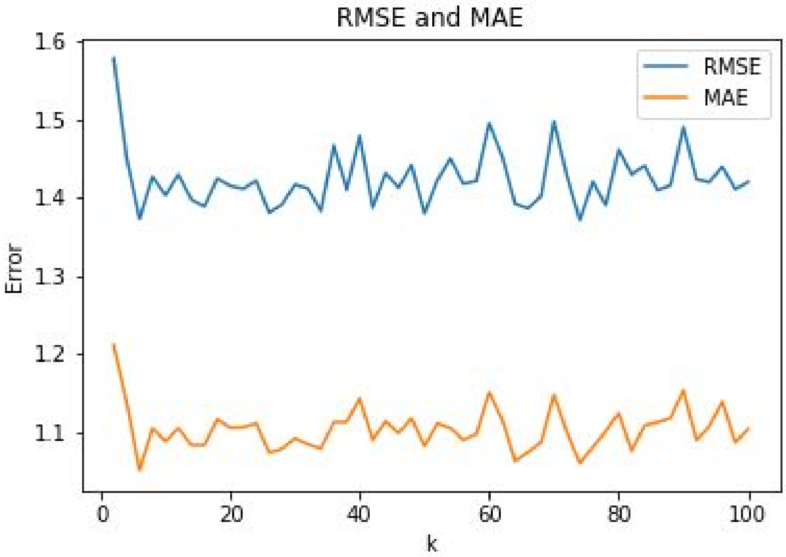
Min k	**16**
Steady state mean RMSE	**1.0057**
Steady state mean MAE	**0.7734**

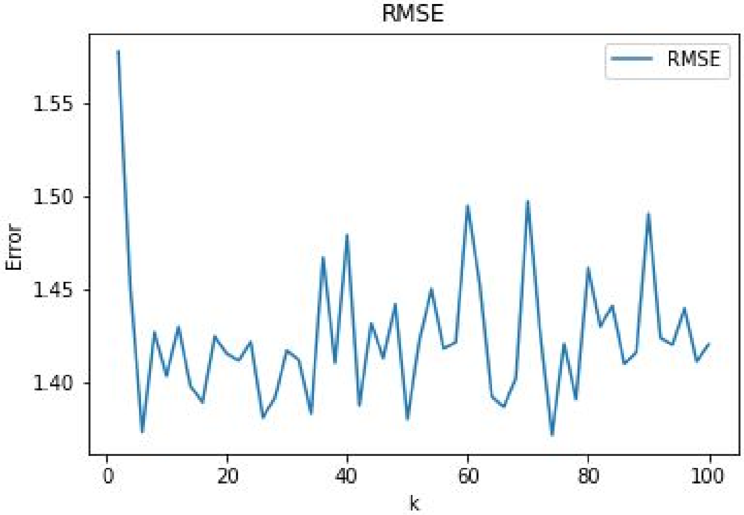
What would be the benefit of trimming our testset for unpopular movies only? Should we do it?
Clearly, we should not analyze our recommendation model based on unpopular testset only. One reason is extremely sparse data, which may result in incorrect predictions for unknown movies. Second reason is that the steady state RMSE and MAE both have increased for unpopular movies (0.91 vs 1.0057 and 0.70 vs 0.77) and the min k value is 16. This shows that analyzing our model against unpopular movies testset will not be beneficial. This also means that our knn model cannot be used efficiently for predicting ratings for unpopular movies.

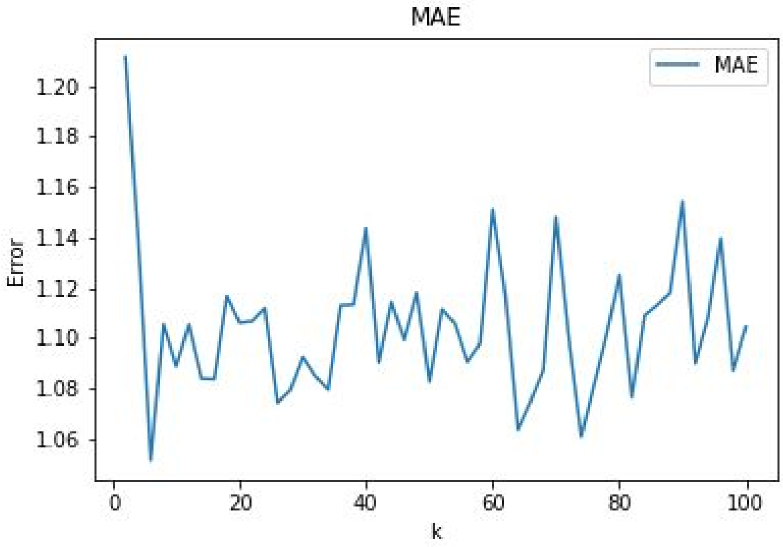
Question 14

Design a k-NN collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation. Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

In this task, we trim our testset for movies with high variance. For this, we retain the set of movies which have received at least 5 ratings and have a variance of the rating values of the movies is at least 2. This implies the set of highly controversial movies, movies that impact a few audiences more than others, etc. The below plots were obtained for average RMSE and average MAE vs k, after performing 10 fold cross validation.


RMSE and MAE
RMSE
MAE
Error
k
1.6
1.5
1.4
1.3
1.2
1.1
0
20
40
60
80
100


RMSE
RMSE
Error
k
1.55
1.50
1.45
1.40
0
20
40
60
80
100



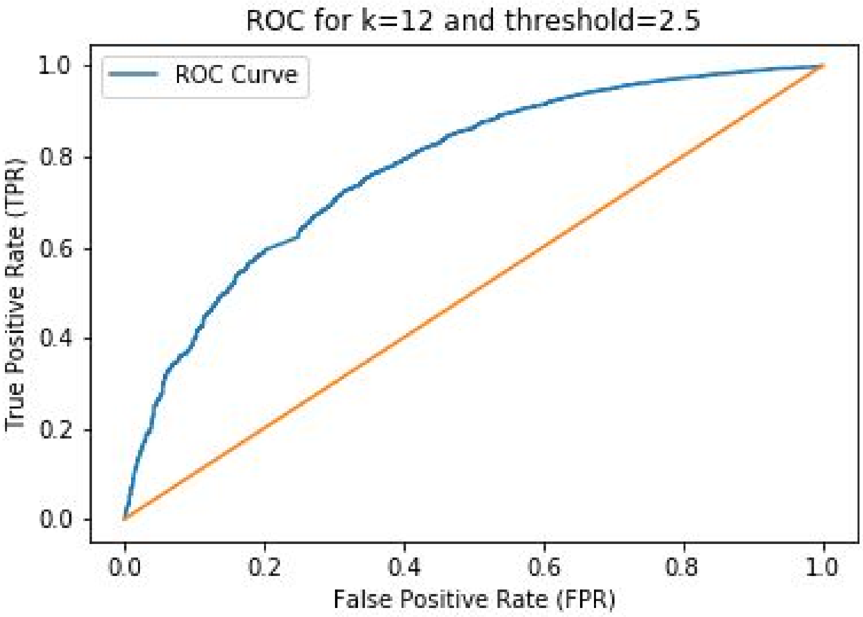
Minimum Average RMSE : 1.3541

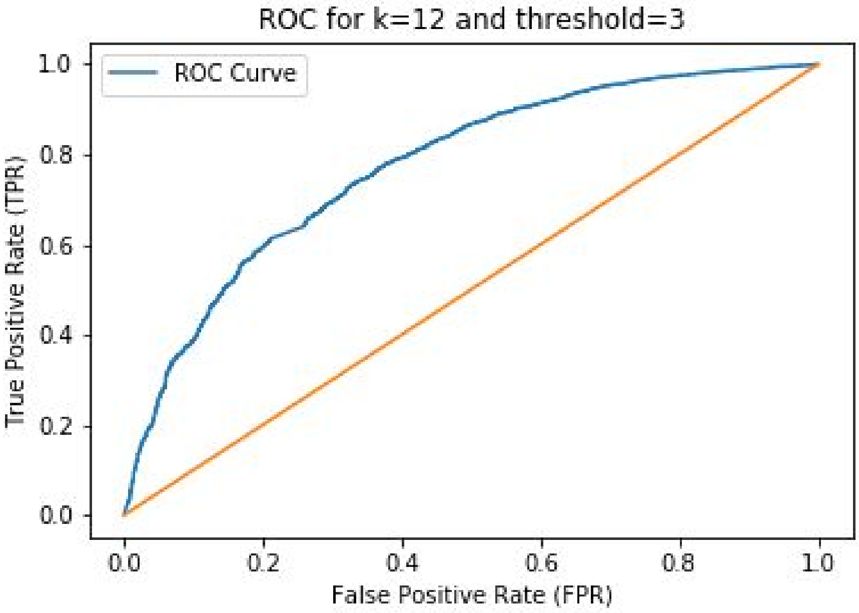
We observe here that all the three plots for high variance movies are not smooth. Usually, this implies noisy data. However, in our analysis, it simply means that testing our knn-collaborative filter against movies with high variance will not provide very accurate results.
Why is this so? If we observe our plots in Question 6, we realize that the number of movies with a variance of 2 or more is extremely less. On checking the count, we found that only a meagre 199 movies with a variance greater than 2, of which only 55 movies satisfied the criteria of having number of ratings greater than 5. Hence, out of 9125 movies, we train our dataset for about 8213 movies, and test it against very few movies in our testset (not all 55 would be contained in a slice of the testset).

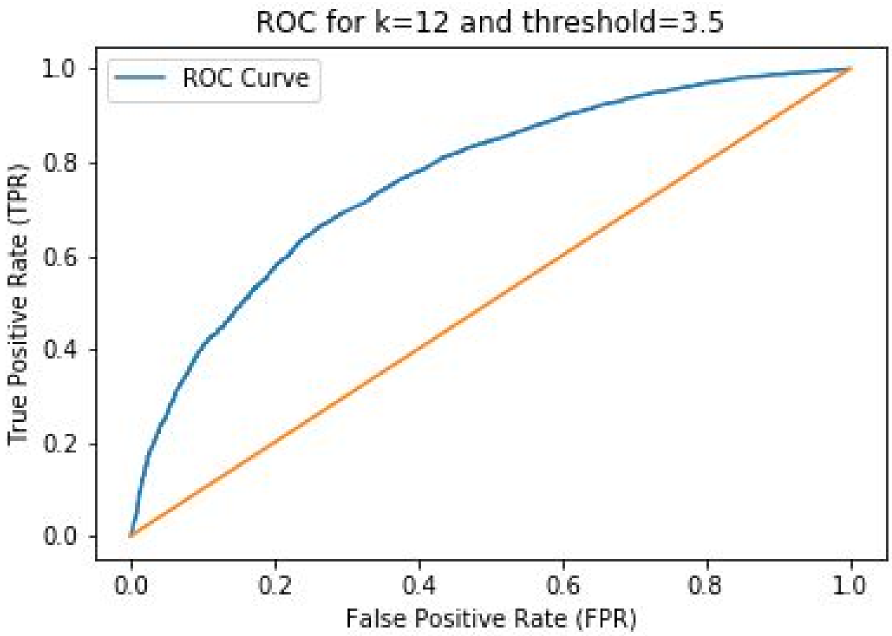
Hence, the graphs obtained above show a very high error rate for both RMSE and MAE. On checking further for steady state values, we found that RMSE values were in range 1.35 to 1.42 and MAE values were in the range of 1.09 to 1.11. The values were randomly increasing or decreasing and no steady state was achieved.
Should we use knn for testing high variance movies? We can use knn for testing high-variance movies and predicting a possible rating for the same , however the accuracy is not guaranteed. High variance movies are usually liked and disliked by different audiences. Hence, even though we calculate a possible rating for such movie using knn, the probability of error is high, and so the probability of our predicted rating matching the ground truth rating for that user.

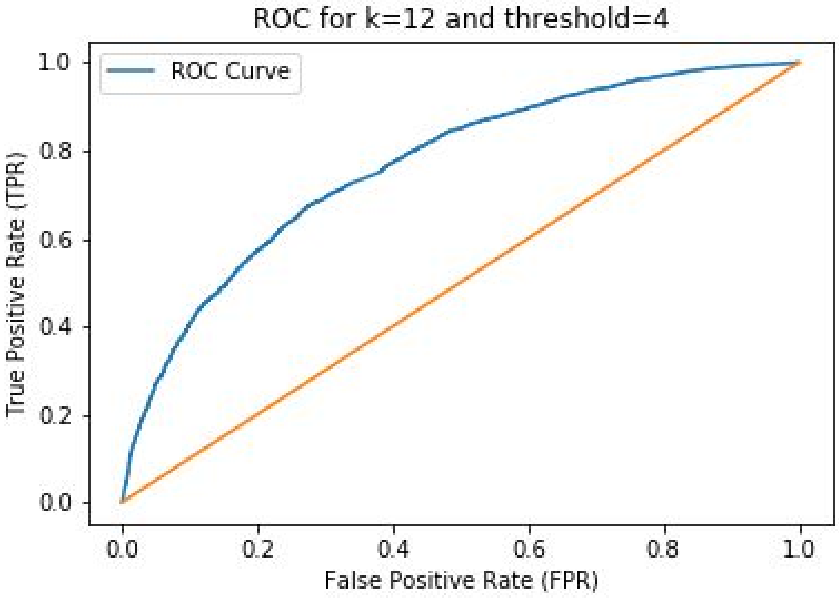
Question 15

Plot the ROC curves for the k-NN collaborative alter designed in question 10 for threshold values [2:5; 3; 3:5; 4]. For the ROC plotting use the k found in question 11. For each of the plots, also report the area under the curve (AUC) value.


ROC for k=12 and threshold=2.5
ROC Curve
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


ROC for k=12 and threshold=3
ROC Curve
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0





Area Under the Curve

k = 22				
t	**2.5**	**3**	**3.5**	**4**
Area under curve	0.7724	0.7745	0.7768	0.7787

Observations:

The ROCs obtained are in a good shape.

The AUC values are closer to 1, which is characteristic of a good model.

MODEL BASED COLLABORATIVE FILTERING

In model-based collaborative Filtering, models are developed using ML algorithms to predict users' rating of unrated items.

Question 16:

Is the optimization problem given by equation 5 convex? Consider the optimization problem given by equation 5. For U fixed, formulate it as a least-squares problem.

The equation in the above question is:

$$\underset{U,V}{\text{minimize}} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2$$

A *non-convex optimization problem* is any problem where the objective or any of the constraints are non-convex. Such a problem may have multiple feasible regions and multiple locally optimal points within each region. It can take time exponential in the number of variables and constraints to determine that a non-convex problem is infeasible, that the objective function is unbounded, or that an optimal solution is the "global optimum" across all feasible regions [1].

The optimization problem given by equation above is **non convex**. The fact that both U's and V's values are unknown variables makes this cost function non-convex.

Now, if we keep U fixed and solve for V : the above equation gets reduced to a simple convex problem of least squares as follows :

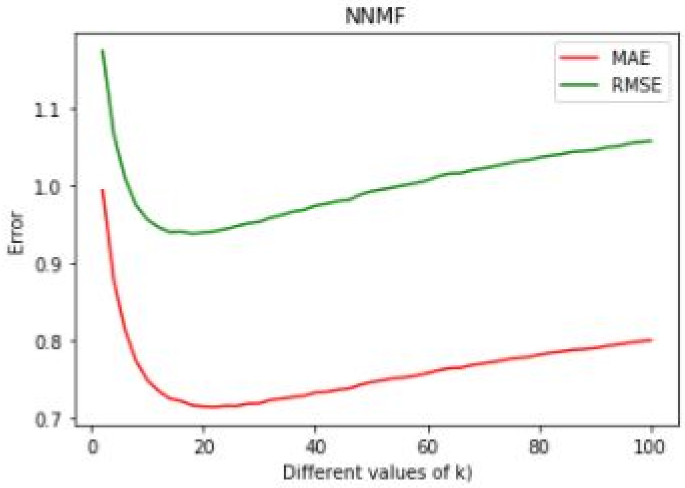
$$\underset{\mathbf{V}}{\mathbf{minimize}} \sum_{\mathbf{i=1}}^{\mathbf{m}}\sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{W_{ij}} \left(\mathbf{r_{ij}} - \mathbf{U} * \mathbf{v_J^T}\right)^{\mathbf{2}}_{\mathbf{ij}}$$

where v ∈ V

Question 17:

Design a NNMF-based collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate it's performance using 10-fold cross-validation. Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. Plot the average RMSE (Y-axis) against k (X-axis) and the average MAE (Y-axis) against k (X-axis).

For solving this question, use the default value for the regularization parameter.



Question 18:

Use the plot from question 17, to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE. Is the optimal number of latent factors same as the number of movie genres?

Optimal number of latent factors is **20.**

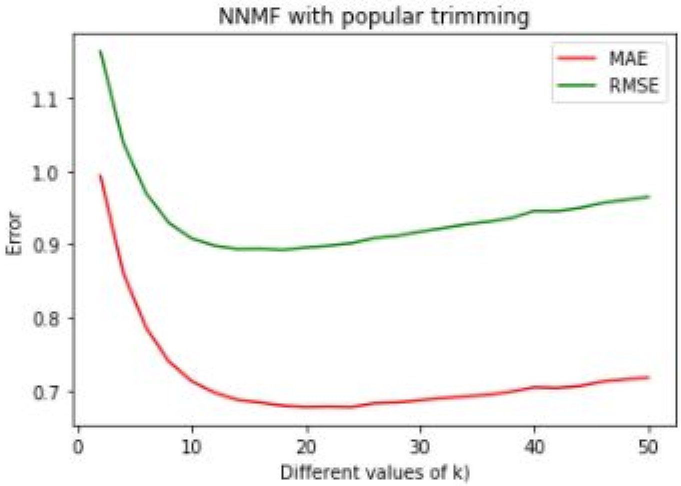
Minimum average RMSE = **0.9381**

Minimum average MAE= **0.7138**

Yes, optimal number of latent factors is equal to the number of movie genres.

Question 19:

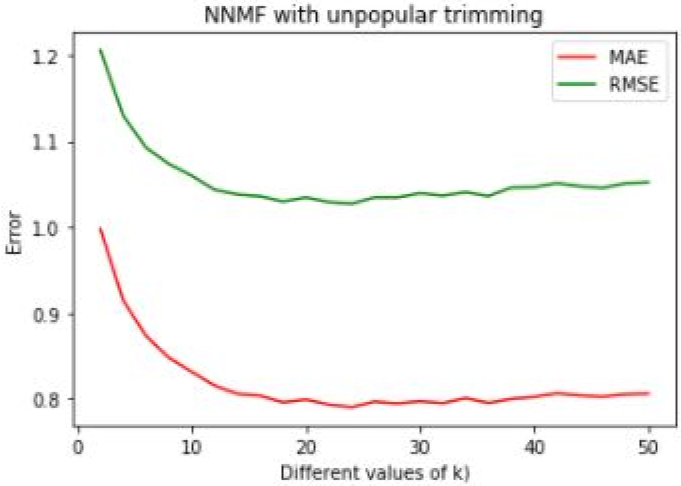
Design a NNMF collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate it's performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis).
Also, report the minimum average RMSE



Minimum Average RMSE is **0.8926**

Question 20:

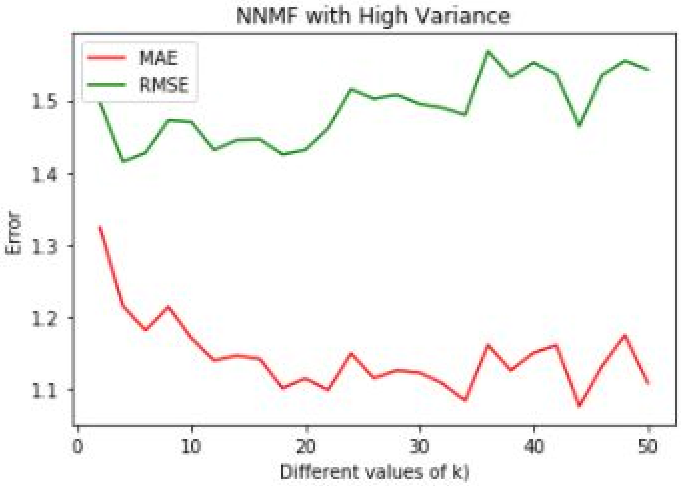
Design a NNMF collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate it's performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis).
Also, report the minimum average RMSE



Minimum Average RMSE is **1.02722**

Question 21:

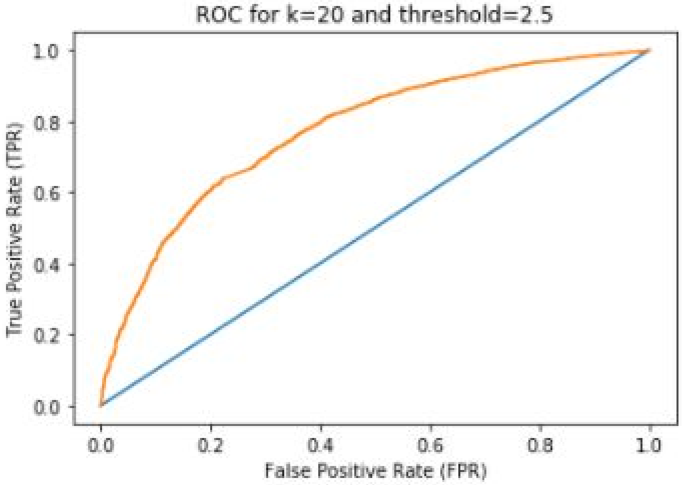
Design a NNMF collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate it's performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE

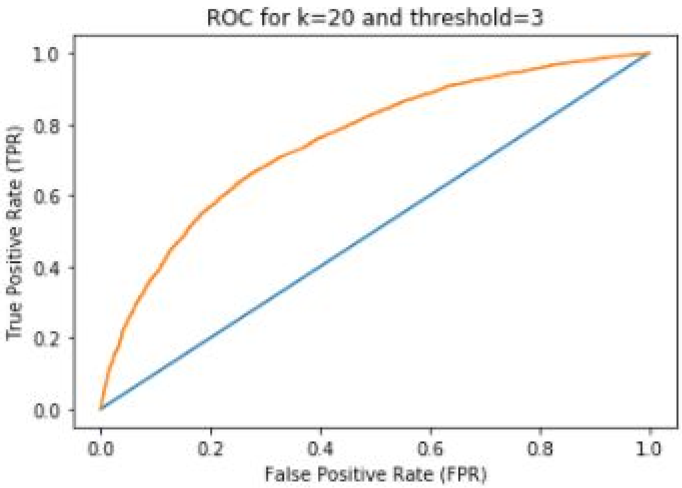


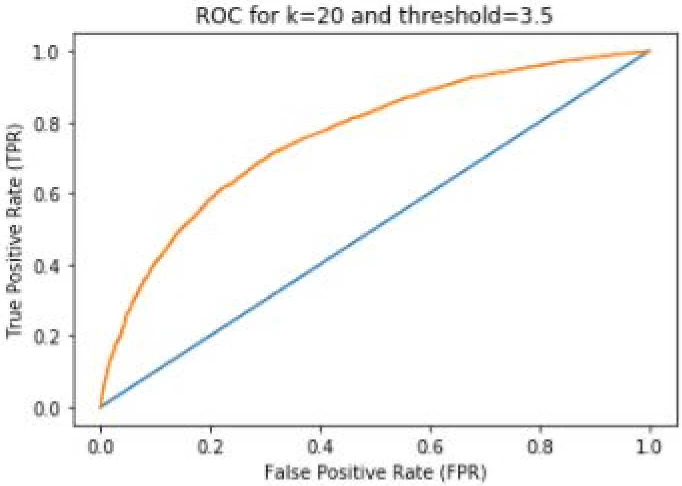
Minimum Average RMSE is **1.4151**

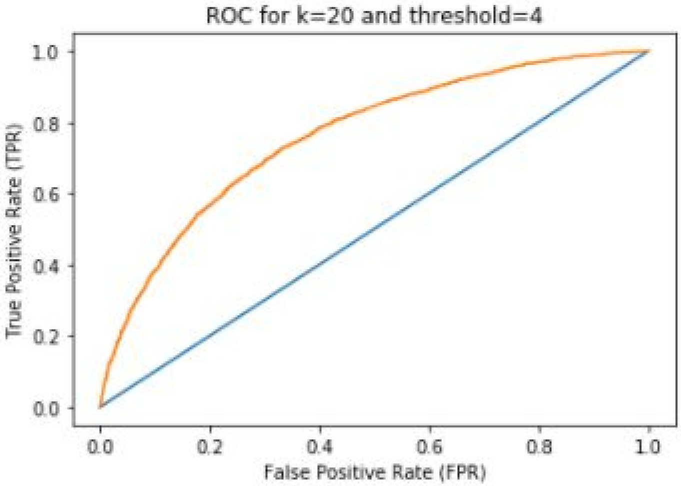
Question 22:

Plot the ROC curves for the NNMF-based collaborative filter designed in question 17 for threshold values [2.5; 3; 3.5; 4]. For the ROC plotting use the optimal number of latent factors found in question 18. For each of the plots, also report the area under the curve (AUC) value.









For NNMF k = 20				
t	**2.5**	**3**	**3.5**	**4**
Area under curve	0.7726	0.7542	0.7616	0.7614

We can observe that all the ROCs are in good shape.

Question 23:

Perform Non-negative matrix factorization on the ratings matrix R to obtain the factor matrices U and V , where U represents the user-latent factors interaction and V represents the movie-latent factors interaction (use k = 20). For each column of V , sort the movies in descending order and report the genres of the top 10 movies. Do the top 10 movies belong to a particular or a small collection of genre? Is there a connection between the latent factors and the movie genres?

The following table shows the genre of the top 10 movies for a couple of columns. We can observe that the top 10 movies mostly fall under the genre **"COMEDY" or "DRAMA".**

Drama	Comedy|Drama	Comedy|Mystery
Comedy|Drama	Drama|Thriller	Drama
Action|Thriller	Drama|Thriller	Horror
Comedy|Western	Adventure|Drama	Adventure|Drama
Drama	Musical	Comedy|Drama|Romance
Drama|Romance	Drama	Comedy
Comedy	Drama	Comedy
Drama|Romance	Action|Adventure|Drama|Sci-Fi	Comedy|Romance
Comedy|Romance	Comedy|Western	Comedy|Drama
Crime|Drama	Comedy|Romance	Action|Adventure|Drama

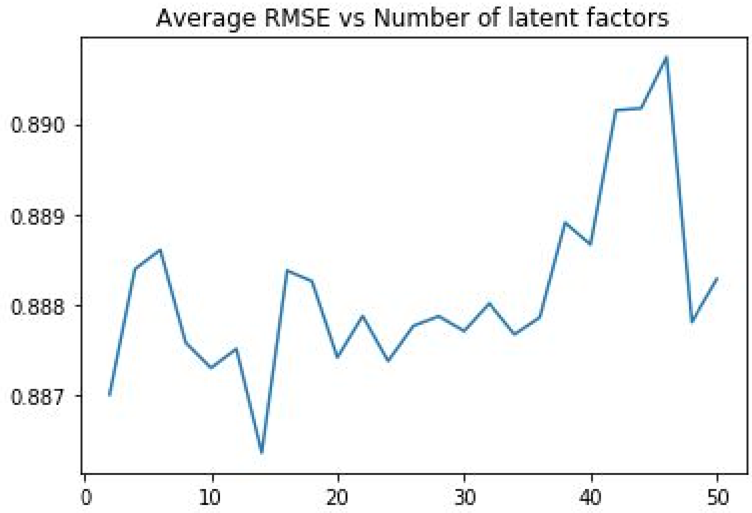
Every movie is tied to one or more categories or genres in some sense, and each user based on his own likes or dislikes prefers one genre over the other. Ratings that a user assigns to a movie is a function of similarity between the user's liked categories and the disliked categories to which the movies belong to. Latent factors help us to identify those broad categories to which the users liking fall. These categories may be a single genre, or multiple genres. There are other factors as well, such as a user's liking for particular movie stars, languages, etc which the movie genres do not contain. It is hard to identify the effect of each of those factors on the user's rating individually. Latent factors help us identify all those categories, thus helping us predict the ratings better.

Matrix Factorization with Bias

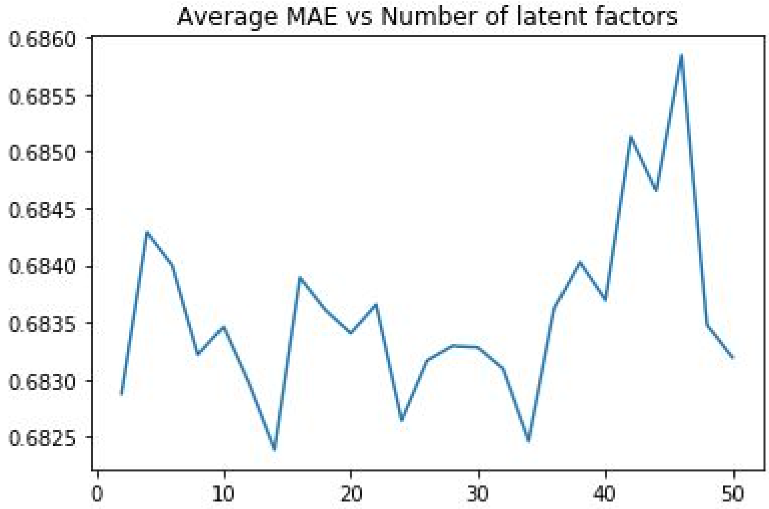
In MF with bias, a bias term is added to the cost function for each user and item. We evaluate the performance of this algorithm using 10-fold cross validation on the MovieLens dataset. We compute the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) on each of the 10 folds as well as the entire dataset. We also plot roc curves for different scenarios to compare the performance of the algorithm.

Question 24

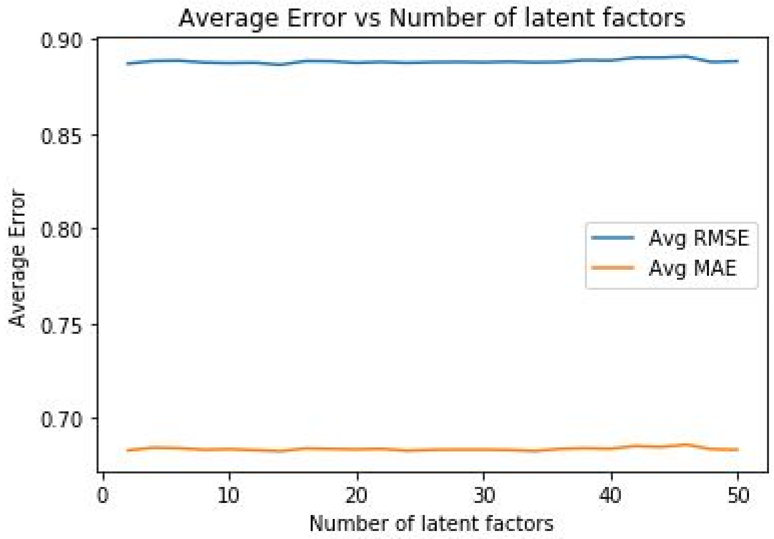
Design a MF with bias collaborative filter and evaluate its performance using 10-fold cross-validation. Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. Plot the average RMSE (Y-axis) against k (X-axis) and the average MAE (Y-axis) against k (X-axis). For solving this question, use the default value for the regularization parameter.



As seen from the above plot, the **average RMSE** for the MF with bias collaborative filter ranges from **0.8860 to 0.8910** as we change the number of latent factors from 2 to 50. The value of average RMSE is **least** at **k=14** which is **0.8864**. The **maximum** value is **0.8907** obtained for **k=46**.



As seen from the above plot, the **average MAE** for the MF with bias collaborative filter ranges from **0.6820 to 0.6880** as we change the number of latent factors from 2 to 50. There is a lot of variation in the values of mean error and we observe dips at **k=14** and **k=34**. The **maximum** value is **0.6858** obtained for **k=46**.



The above plot indicates that there is a **difference** of **0.2** between the values of average RMSE and average MAE.

Question 25

Use the plot from question 24, to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE.

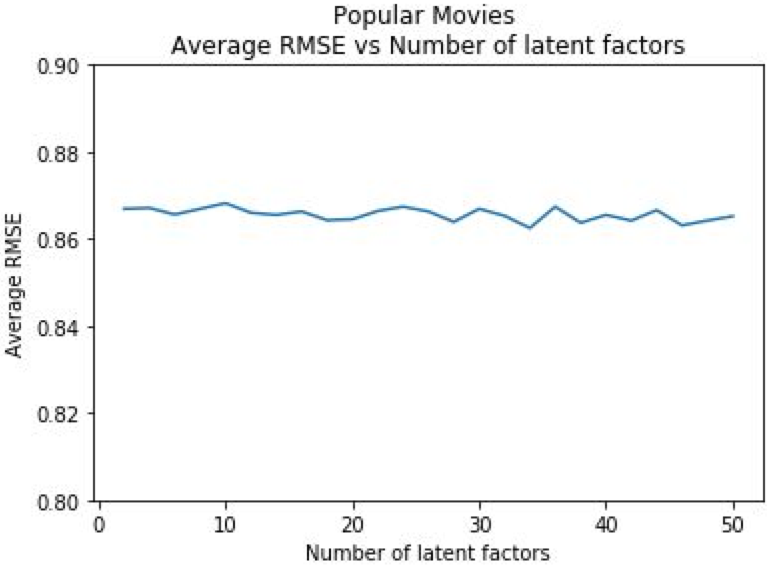
- **The optimal number of latent factors is 14.**

Minimum Mean RMSE	Minimum Mean MAE
0.6824	0.8864

Question 26

Design a MF with bias collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

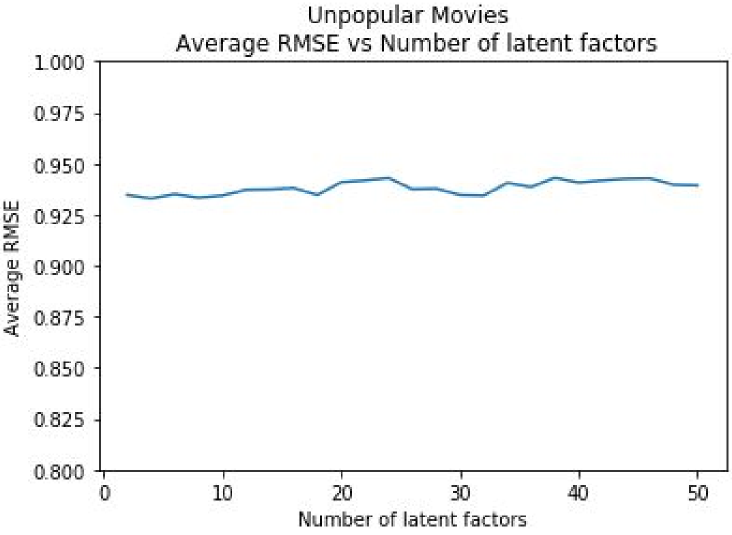
- Minimum average RMSE: **0.8625**
- **Observations:**
  For all values of k ranging from 2 to 50, the value of average RMSE for popular movie trimming is somewhere around 0.86 which is quite good of a performance. This is justified by the fact that there are enough ratings available for training and prediction of popular movies.



Question 27

Design a MF with bias collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

- **Minimum average RMSE: 0.9328**



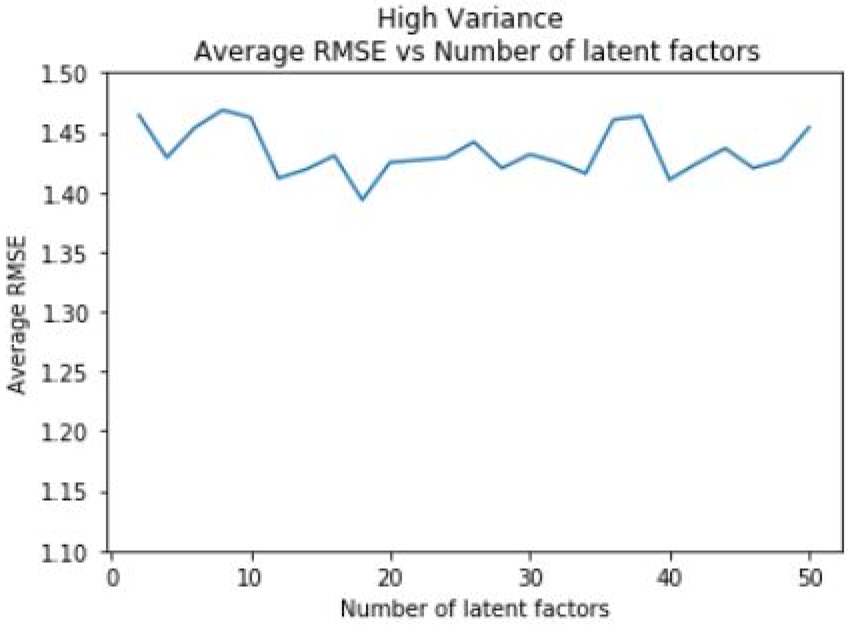
- **Observations:**
  For all values of k ranging from 2 to 50, the value of average RMSE for unpopular movie trimming is somewhere around 0.93 which is greater than the average RMSE for popular movies

trimmed dataset. This is due to the fact that there are fewer ratings available from the ratings matrix for unpopular movies and this hampers the performance of the dataset.

Question 28

Design a MF with bias collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

- **Minimum average RMSE: 1.3934**



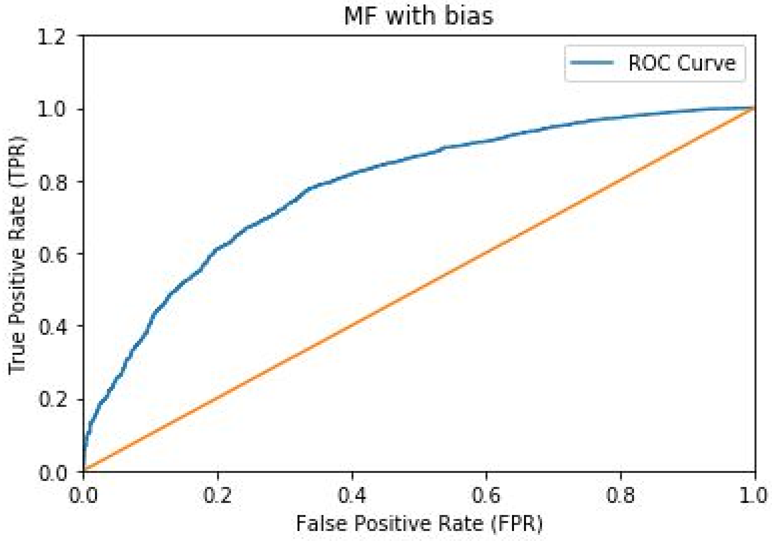
- **Observations:**
  For all values of k ranging from 2 to 50, the value of average RMSE for high variance movie trimming varies a lot from 1.35 to 1.50. As expected, the value of average RMSE obtained is very high compared to that for popular and unpopular movie trimming. This is due to the fact that training and prediction is difficult for set of ratings that have high variance.

Question 29

Plot the ROC curves for the MF with bias collaborative filter designed in question 24 for threshold values [2.5; 3; 3.5; 4]. For the ROC plotting use the optimal number of latent factors found in question 25. For each of the plots, also report the area under the curve (AUC) value.

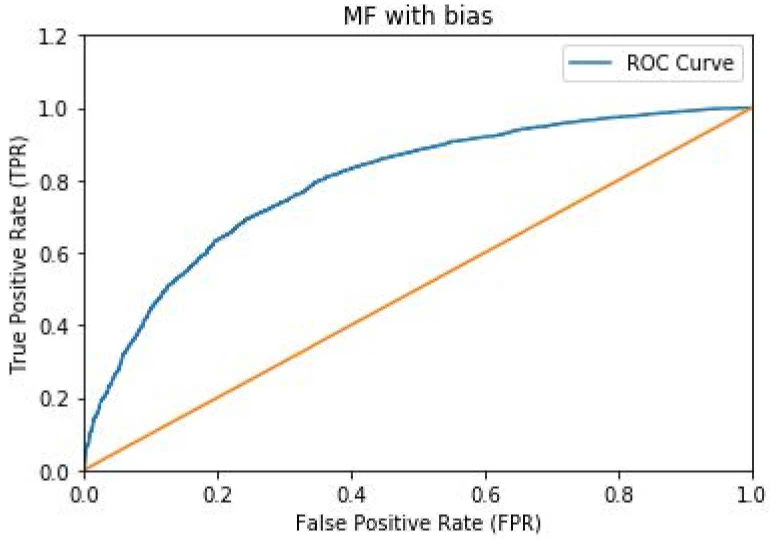
1. Threshold = 2.5

Area under ROC = 0.7806



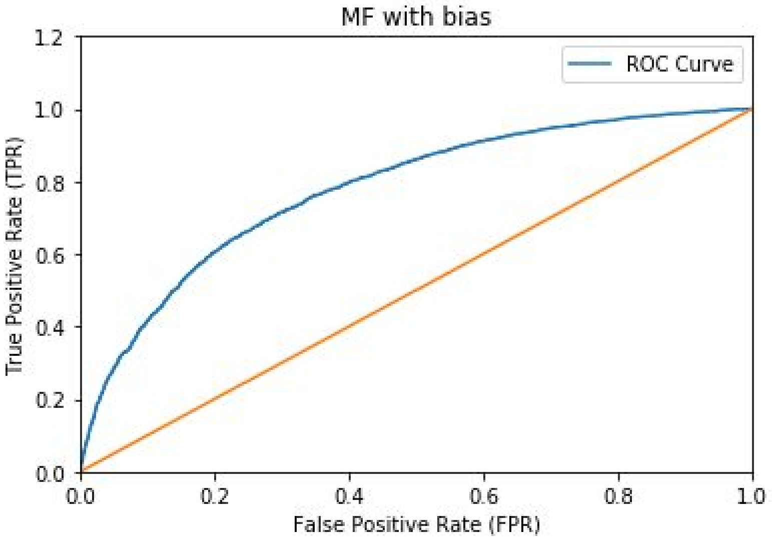
2. Threshold = 3

Area under ROC = 0.7920



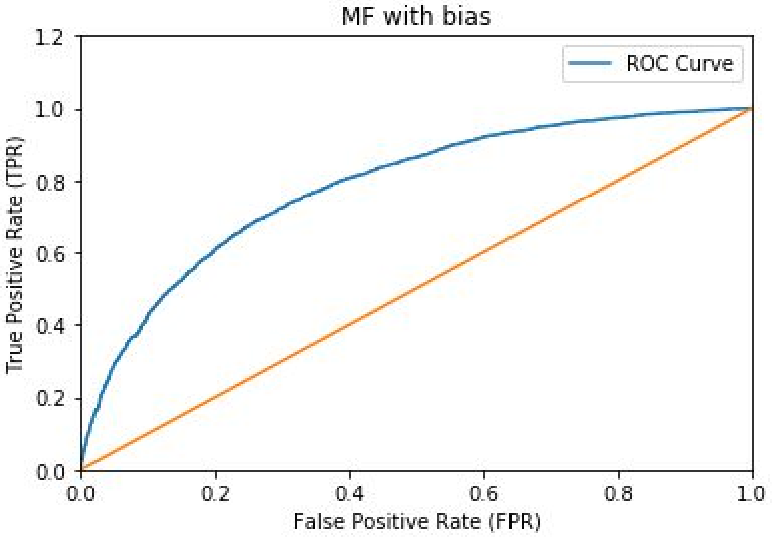
3. Threshold = 3.5

Area under ROC = 0.7751



4. Threshold = 4

Area under ROC = 0.7809



Observations:

From the above plots, we can say that as we increase the threshold, the area under curve increases until it reaches maximum and after that it decreases. The area under the curve is maximum for threshold=3 for MF with bias collaborative filter. If we consider that the users who rated the movies with 3.0 or more, liked the movie then, we are able to provide them with better recommendations as compared to the other values of threshold.

Naïve Collaborative Filtering

In this section, we have designed a Naïve Collaborative Filter. This filter is then used to predict the ratings of the movies in the MovieLens dataset. Here, we develop a naïve prediction formula which basically returns the mean rating of the user as the predicted rating for an item. Hence the name ‘**Naive’.**

a. Prediction Formula

As mentioned above, the predicted rating for an item of a user i is the mean of the ratings of that particular user i. We can denote this by the following formula:

$$\hat{r}_{ij} = \mu_i$$

In the above formula, μ_i is the mean rating of the user i.

b. Design and Test via Cross Validation

We have used the above prediction formula to design the Naïve Collaborative filter in which we have predicted the ratings of the movies in the MovieLens dataset. It’s performance is evaluated using 10-fold cross validation on the MovieLens dataset. We have computed the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) on each of the 10 folds as well as the entire dataset. The RMSE and MAE values of the entire dataset were calculated by averaging the respective values across all the 10 folds.

Procedure:

- Split the dataset into 10 folds (train set and test set)
- Predict the ratings of the movies in the testset only using the prediction function
- Compute the RMSE and MAE
- Repeat the above two steps for all the 10 folds and compute the mean RMSE and MAE

Question 30

Design a naive collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate it's performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

Answer:

Mean RMSE across 10 folds	Mean MAE across 10 folds
0.9128	0.74462

Naïve Collaborative Filter Performance on Trimmed TestSet

In this part, we have applied the Naïve collaborative filter on the trimmed testset. Trimming was done on the main dataset first and then 10 folds were applied to it. Trimming techniques used here are:

- Only the popular dataset
- Only the unpopular dataset

- Only the high variance dataset

Question 31

Design a naive collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate it's performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

Mean RMSE across 10 folds	Mean MAE across 10 folds
0.9066	0.7430

Question 32

Design a naive collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate it's performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

Mean RMSE across 10 folds	Mean MAE across 10 folds
1.0214	0.7724

Question 33

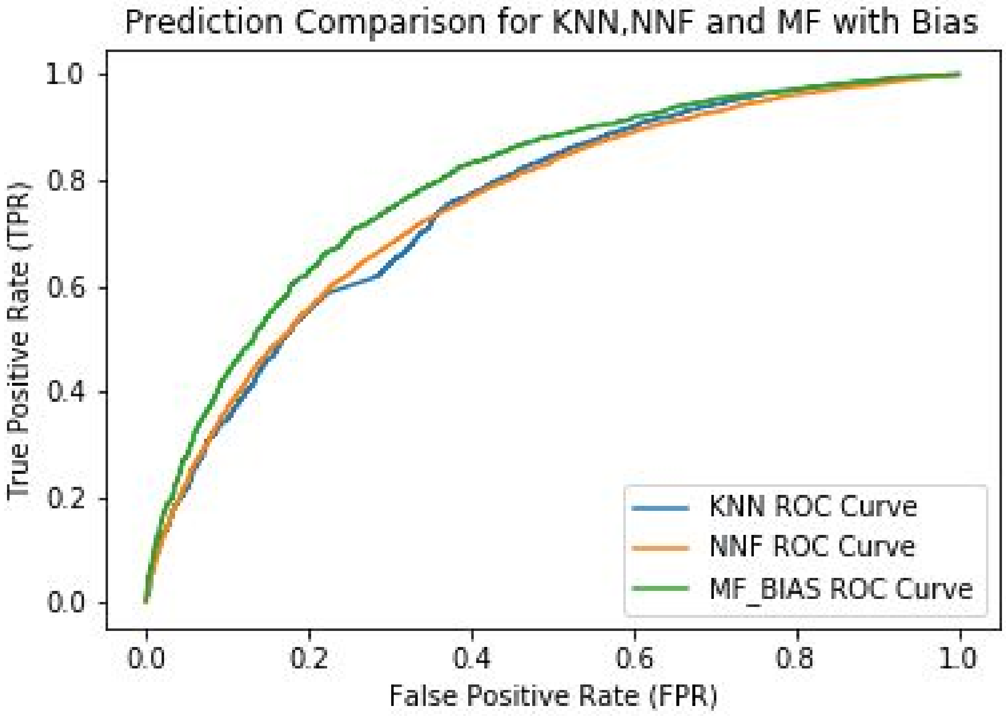
Design a naive collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate it's performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

Mean RMSE across 10 folds	Mean MAE across 10 folds
2.3548	1.2619

Observations: This collaborative filter predicts ratings specific to a user. As seen from the above tables, we can comment that the RMSE for the high variance dataset is more as compared to the popular dataset and the unpopular dataset and hence the prediction will be poor in high variance dataset as compared to the other two.

Question 34

Plot the ROC curves (threshold = 3) for the k-NN, NNMF, and MF with bias based collaborative filters in the same figure. Use the figure to compare the performance of the filters in predicting the ratings of the movies.



As we can observe from the above graph, KNN and NNMF give almost similar area under ROC curves. Thus, the performance of KNN and NNMF filters is quite comparable. The best performance is obtained by using MF with bias collaborative filter which is indicated by the maximum area under the curve.

Question 35

Precision and Recall are defined by the mathematical expressions given by equations 12 and 13 respectively. Please explain the meaning of precision and recall in your own words.

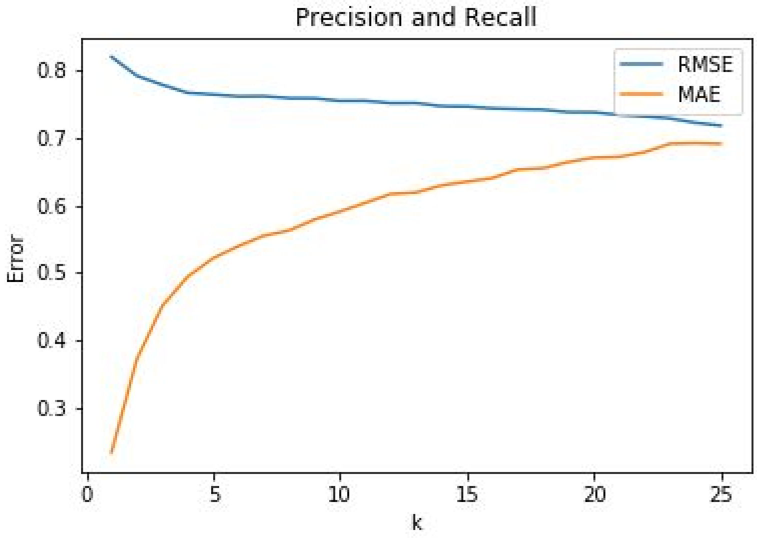
Answer: Precision and recall are the measures used to evaluate the relevance of the rankings presented to users by recommendation systems.

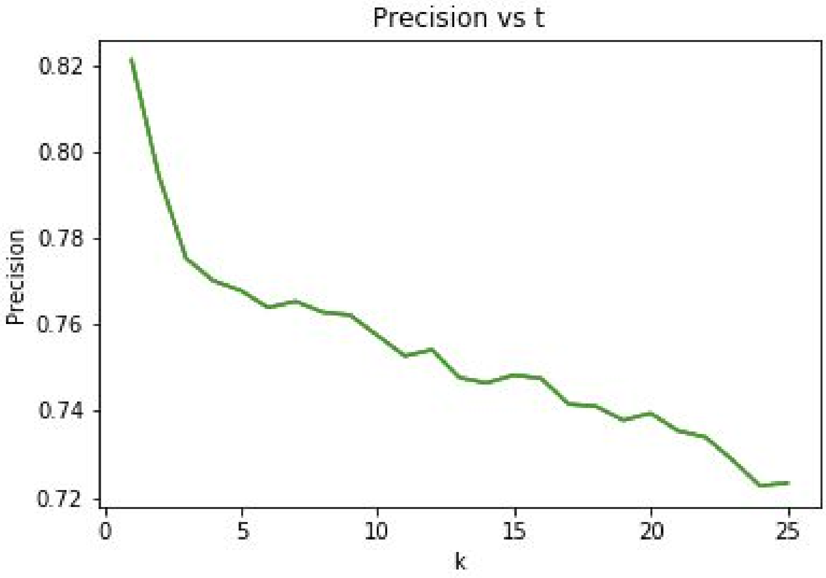
Precision is the proportion of recommended items in the top-k items that are relevant.

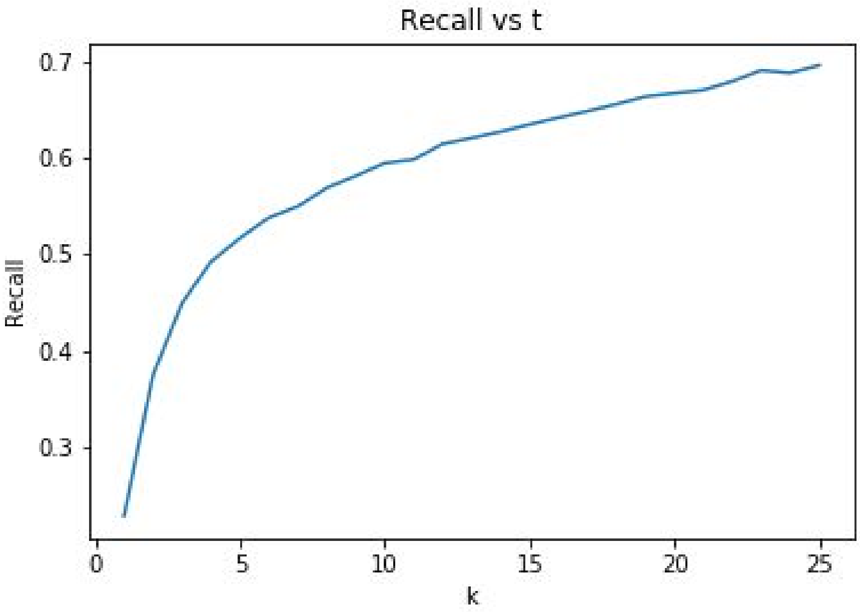
Recall is the proportion of relevant items found in the top-k recommendations.

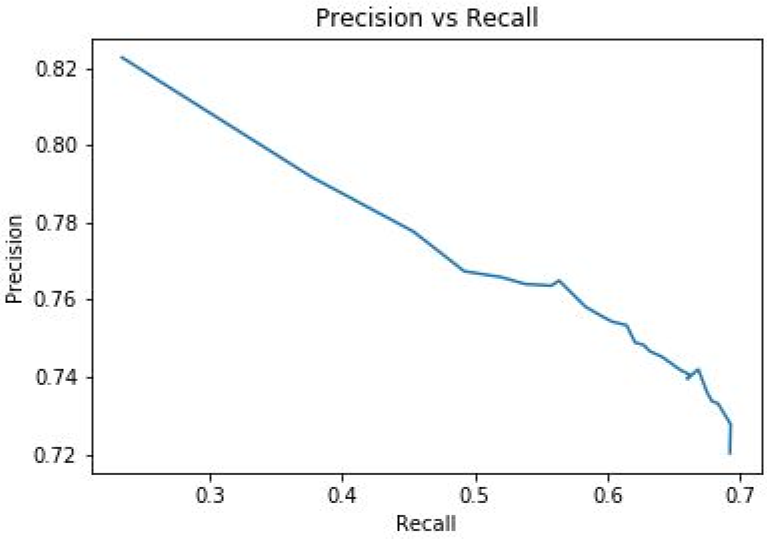
Question 36

Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using k-NN collaborative alter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use the k found in question 11 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.


Precision and Recall
RMSE
MAE
Error
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
k


Precision vs t
Precision
0.82
0.80
0.78
0.76
0.74
0.72
0
5
10
15
20
25
k

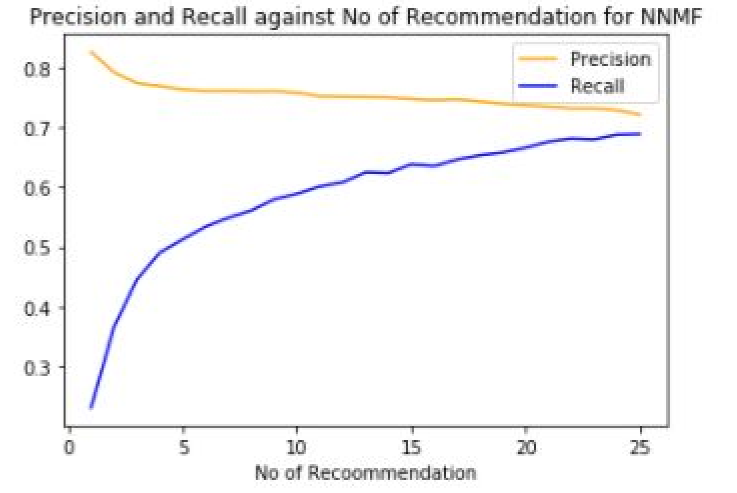


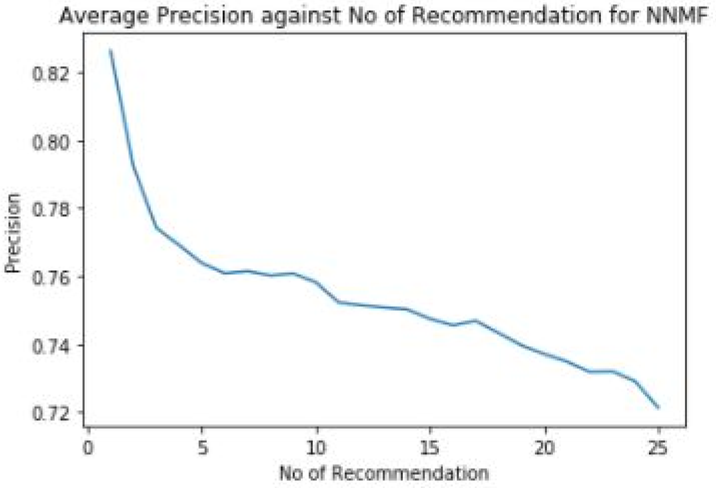


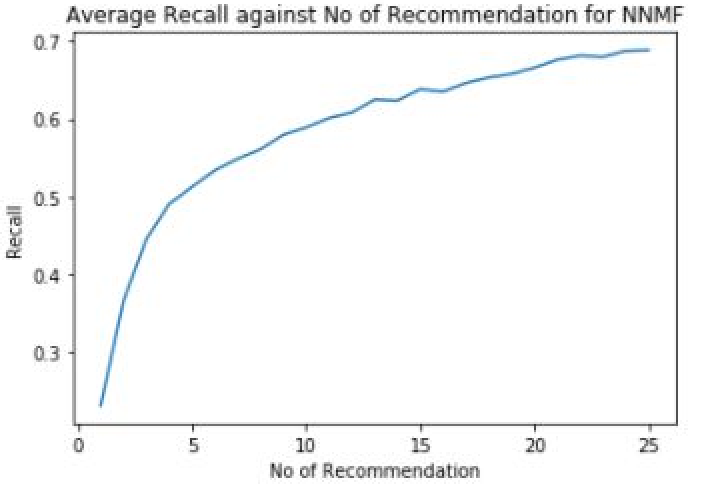
Observations: We can see that as threshold (t) increases, precision decreases and recall increases.

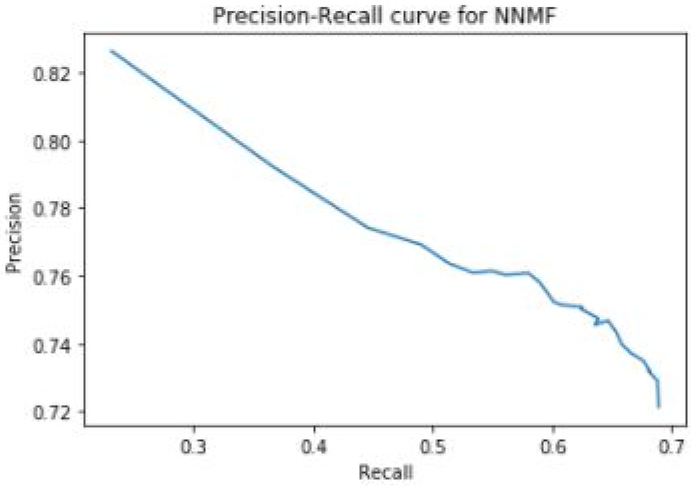
Question 37:

Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using NNMF-based collaborative filter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use optimal number of latent factors found in question 18 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.


Precision and Recall against No of Recommendation for NNMF
Precision
Recall
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
No of Recoommendation


Average Precision against No of Recommendation for NNMF
Precision
0.82
0.80
0.78
0.76
0.74
0.72
0
5
10
15
20
25
No of Recommendation

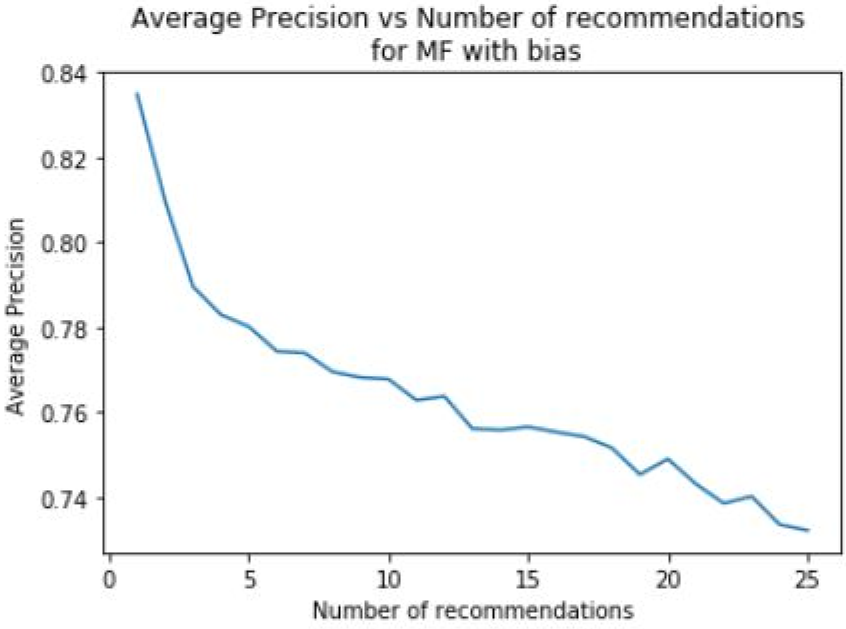

Average Recall against No of Recommendation for NNMF
Recall
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
No of Recommendation

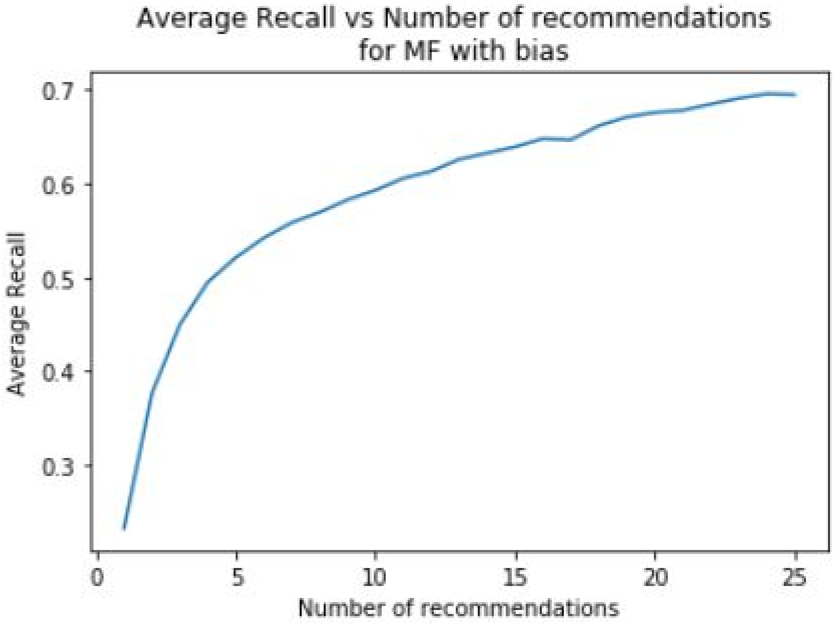


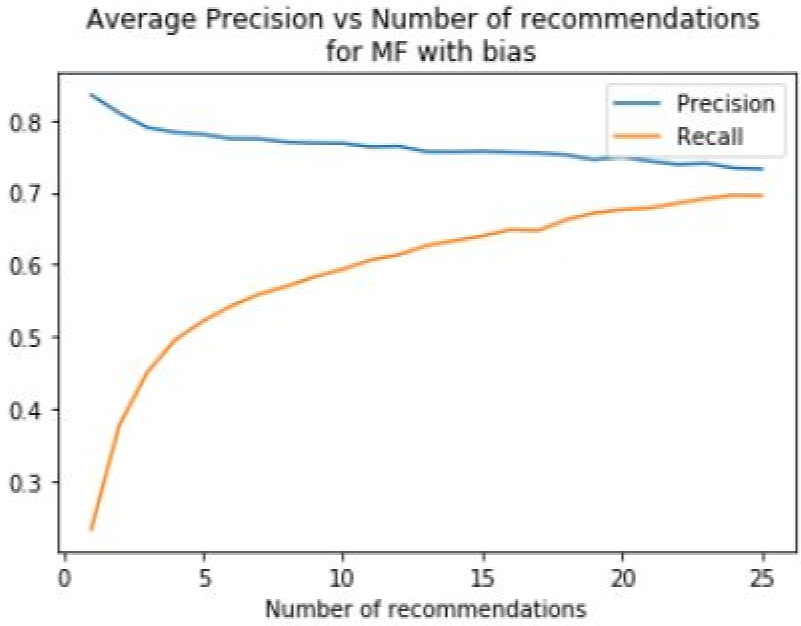
From the above graphs, we can observe that as the number of recommendations (t) increases, recall increases but precision decreases.

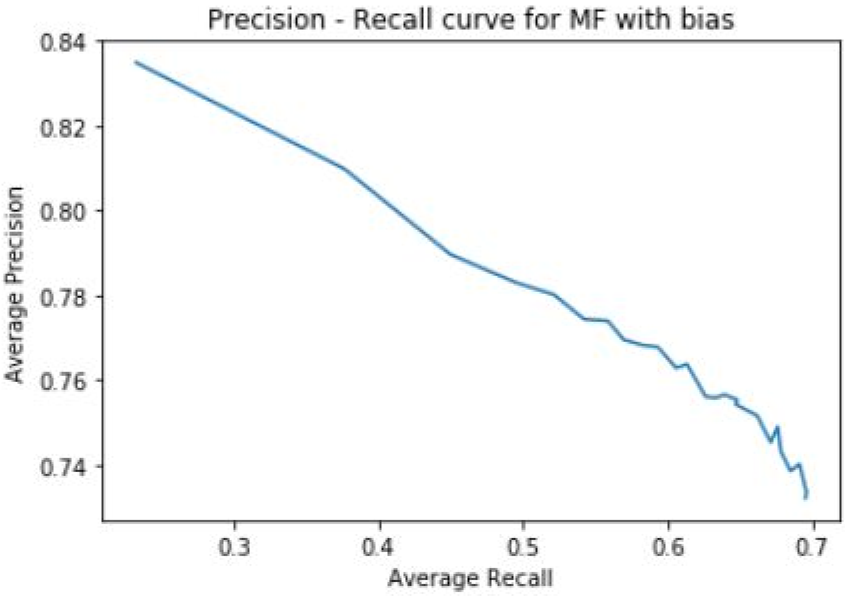
Question 38:

Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using MF with bias-based collaborative filter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use optimal number of latent factors found in question 25 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.




Average Recall vs Number of recommendations
for MF with bias
Average Recall
0.3
0.4
0.5
0.6
0.7
0
5
10
15
20
25
Number of recommendations


Average Precision vs Number of recommendations
for MF with bias
Precision
Recall
0.3
0.4
0.5
0.6
0.7
0.8
0
5
10
15
20
25
Number of recommendations

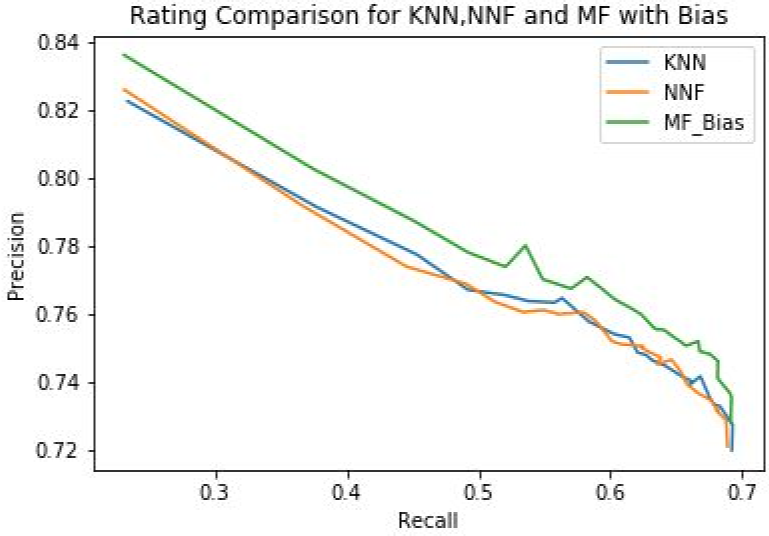


Observations:

- As we increase the number of recommended items for MF with bias, the precision of the recommendation system reduces. Thus, we get the best value of precision of we recommend less number of movies to the user.
- Recall increases rapidly initially with the increase in the number of recommendations and then the rate of increase reduces.
- From the precision-recall curve we can say that as number of recommendations increases, the precision gradually reduces and recall rapidly increases.
- Thus, from our observations, the ranking obtained using MF with bias collaborative filter has better performance on precision as compared to recall for lesser number of recommendations.

Question 39

Plot the precision-recall curve obtained in questions 36,37, and 38 in the same figure. Use this figure to compare the relevance of the recommendation list generated using k-NN, NNMF, and MF with bias predictions.



From the above graph we see that the rating performances for all three collaborative filters is almost similar in terms of precision and recall. However, the MF with bias has slightly better performance than the above two. For these algorithms, the precision increases as recall decreases and vice versa as expected. This means that MF with bias models clearly results in better accuracy and should be utilized for rating predictions in Collaborative Filtering.

CONCLUSION

In this project we have compared the predictions of four different collaborative filters for recommender systems. In addition to predicting the ratings of the complete MovieLens dataset, we have also drawn predictions on popular, unpopular and high variance datasets. The RMSE values for high variance dataset was more than the other two sets in all the collaborative filters. According to the RMSE values and the precision v/s recall curve, we observe that the MF with bias filter gives the best performance of the four filters and should be used in collaborative filters for recommender systems.

REFERENCES

1. https://www.solver.com/convex-optimization
2. http://surprise.readthedocs.io/en/stable/knn_inspired.html
3. http://surprise.readthedocs.io/en/stable/model_selection.html#surprise.model_selection.validation.cross_validate
4. http://surprise.readthedocs.io/en/stable/similarities.html
5. http://surprise.readthedocs.io/en/stable/getting_started.html#use-cross-validation-iterators
6. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html
7. http://surprise.readthedocs.io/en/stable/matrix_factorization.html