Large Scale Data Mining: Models and Algorithms
Project #3

Due on February 20st, 2020 at 11: 59 pm

Professor Roychowdhury, Vwani

Wang, Yin, He, Kang

Question 1

One motivation for the recommend system to implement the collaborative filtering rather than other machine learning algorithms is the sparsity of the dataset. Hence, a reasonable guess about the sparsity score of the MovieLens dataset before calculating should be as small as possible. According to the expression stated, the sparsity of our dataset is

$$Sparsity = \frac{N_{ratings}}{N_{movies} \times N_{users}} \approx 1.700 \times 10^{-3} \tag{1.1}$$

where N_{movies} is the number of movies rated by users in the data set. The value of Sparsity does live up to our expectation.

Question 2

Before plotting the histogram of the frequency of rating scores, we should notice that there are only 10 possible values for the rating, each of which is an integral multiple of 0.5. Hence, we should be very carefully when using the *np.histogram* to count the account of rating values in the binned intervals. We add a bias to each binned interval such that each interval contains only one possible rating values. The histogram illustrating the frequency of the rating values is shown as Figure 1

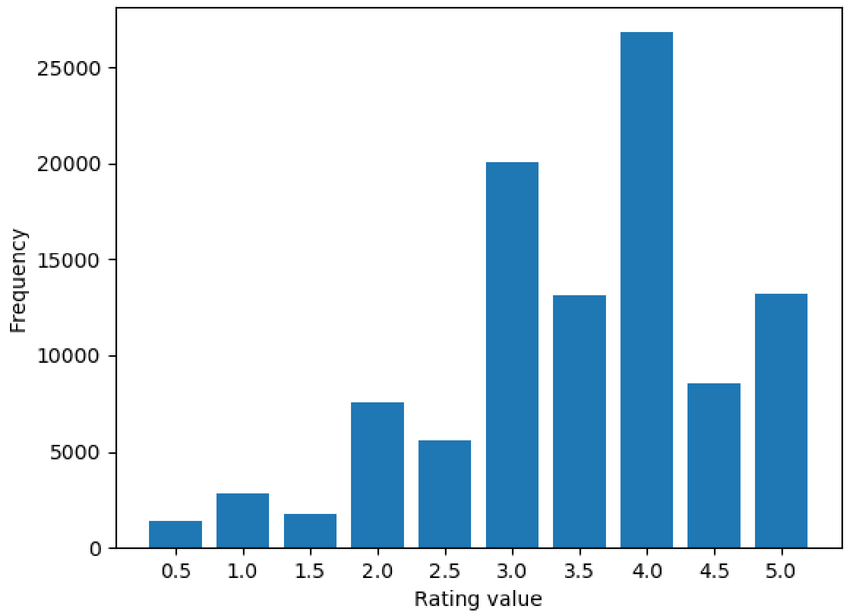

Figure 1: The histogram of the frequency of the rating values.

Question 3

To explicitly show the distribution of the number of ratings received among movies, we are going to plot a curve whose value decreasing with x rather than a histogram. Since there are hundreds of movies do not receive any rating from the users in this dataset, we will directly eliminate these movies to exhibit the

Question 3 continued on next page. . .

character of the distribution clearly. The distribution of the account of rating received among movies after abandoning movies without ratings can refer to Figure 2

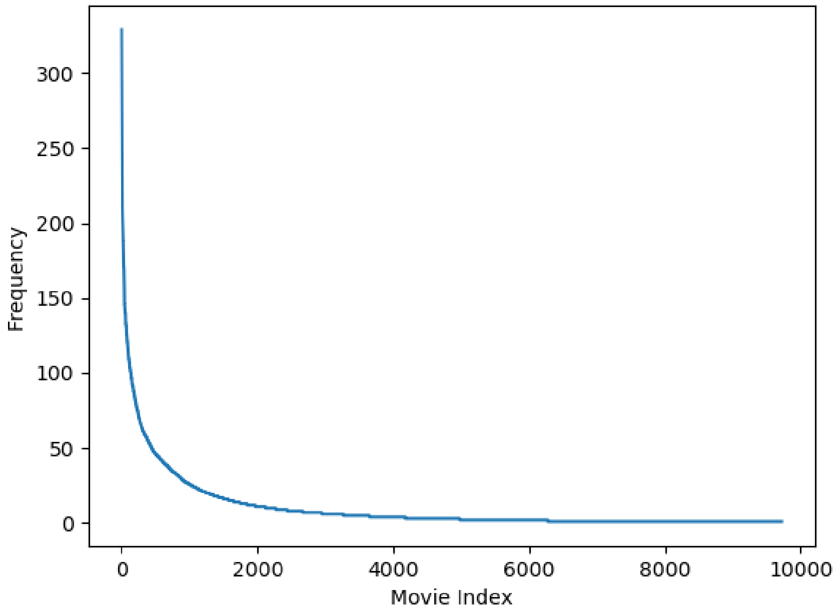

Figure 2: The distribution of the account of rating received among movies.

Question 4

Similarly, we are going to plot the distribution of rating among users with the same configuration in Question 3. Then the corresponding figure is shown as Figure 3

Question 5

According to the Figure 2, we can see that the majority of the movies in the data set have few comments. Thousands of the movies just have one rating from totally different peoples. Hence, it is impractical for us to use inter-item correlations to predict the rating one might give to a specific movie. The samples received by the movie is so small that it may not represent the true latent correlation behind movies. This implies us to use the inter-user correlation to predict the movie rating. The distribution of ratings among users can be seen as Figure 3. Each user at least comments on 20 different movies, making it reasonable to work out the correlation between users.

Question 6

The variance of ratings can be derived easily after knowing the distribution of the comments received by a specific movie. However, the question now is that the ratings are organized according to the users rather than the movies. We have to separate the comments according to the corresponding movies they are given to. There are two possible solution. The vanilla method is to iterate the whole dataset to generate the

Question 6 continued on next page. . .

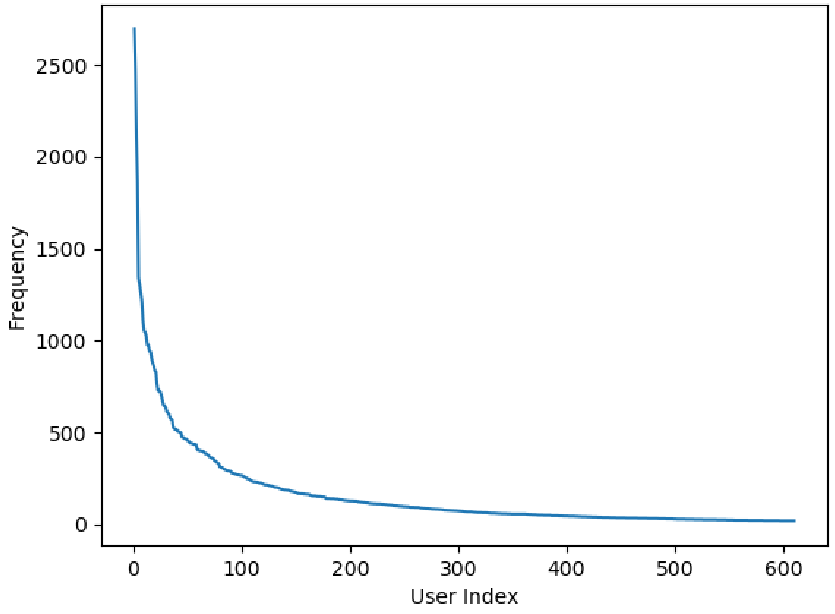

Figure 3: The distribution of the number of ratings among users.

ratings tuple for each movies respectively and calculate the variance according to the corresponding tuple. Another way is dynamic, which will update the variance of the corresponding movie when encountering a new rating. Notice that when a new value is added, the variance of the variable will change as follows:

$$\begin{aligned} V' &= \frac{1}{N+1}\sum_{i=1}^{N+1}(x_i-\mu')^2 = \frac{1}{N+1}\sum_{i=1}^{N+1}(x_i-\mu+\mu-\mu')^2 \\ &= \frac{1}{N+1}\{\sum_{i=1}^{N}[(x_i-\mu)^2+(\mu-\mu')^2+2(x_i-\mu)(\mu-\mu')]+(x_{N+1}-\mu')^2\} \\ &= \frac{1}{N+1}\{NV+N(\mu-\mu')^2+(x_{N+1}-\mu')^2\} \end{aligned} \tag{6.1}$$

Hence, we just need to store the variance, the mean and the account of the ratings each movie receives and update them according to the new rating confronted. According to the equation (6.1), we can finally get a matrix of the features of the movies. Similarly, we will delete the variance of the movies without any ratings received. Using *plt.hist* function, we can get the histogram of the variances of movies shown as Figure 4. The distribution of the variance is similar to the standard normal distribution, which will decrease dramatically if deviating its mean.

Question 7

The formula of μ_u in terms of I_u and r_{uk} can be written as:

$$\mu_u = \frac{\sum_{k\in I_u} r_{uk}}{|I_u|} \tag{7.1}$$

where the $|I_u|$ is the cardinal number of set I_u, namely, the account of the elements in the set I_u.

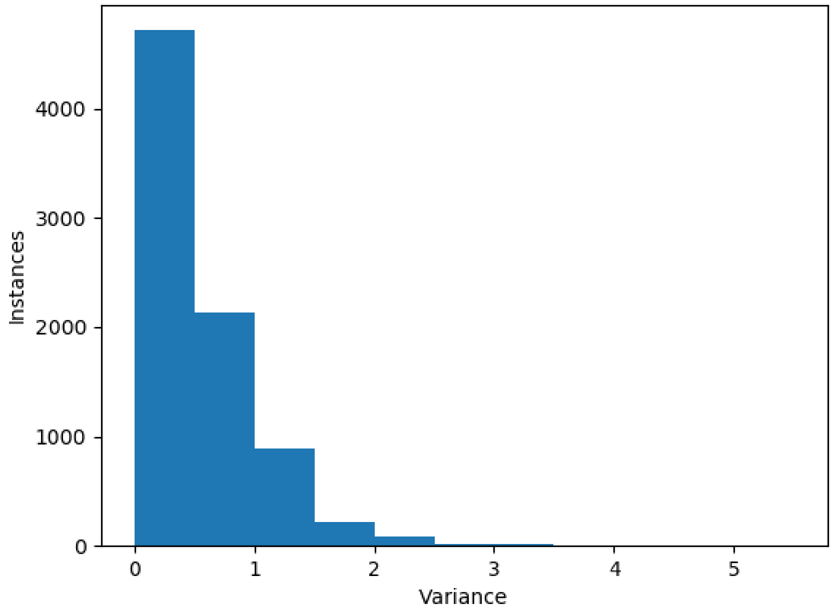

Figure 4: The histogram of the variances of the movies.

Question 8

According to the definition of the I_u and I_v, the term I_u represents the indices of the movies commented by a person u. Hence, the formula $I_u \cap I_v$ means the indices for which ratings have been specified by both user u and v. Since the rating matrix is sparse, it is possible for $I_u \cap I_v = \emptyset$.

Question 9

The reason for utilizing mean-centering to predict the rating of user u for item j can be understood by considering one extreme condition. For example, if there is one person who is critical about movies and gives every movie he has watched a low rating. Even though he might give a rating for a specific movie a relatively high score, which is still low comparing to other ratings this movie has received, the prediction will be impacted to be a low score if using raw rating. Hence, if considering the critical degree of different users, which can be represented as the mean of the ratings one gives, we can get more accurate result on the preference of similar users, thus a more precise prediction based on user correlation.

Question 10

According to the problem, we will going to use the *cross_validate* function in the surprise package and set the number of split folds as 10. The implementation of sweeping is straightforward and can be done by iteration. The result of our code is illustrated as Figure 5 and Figure 6. For clarity, we use the dot-line chart here so that we can find the specific k to minimize the RMSE and MAE in the following question.

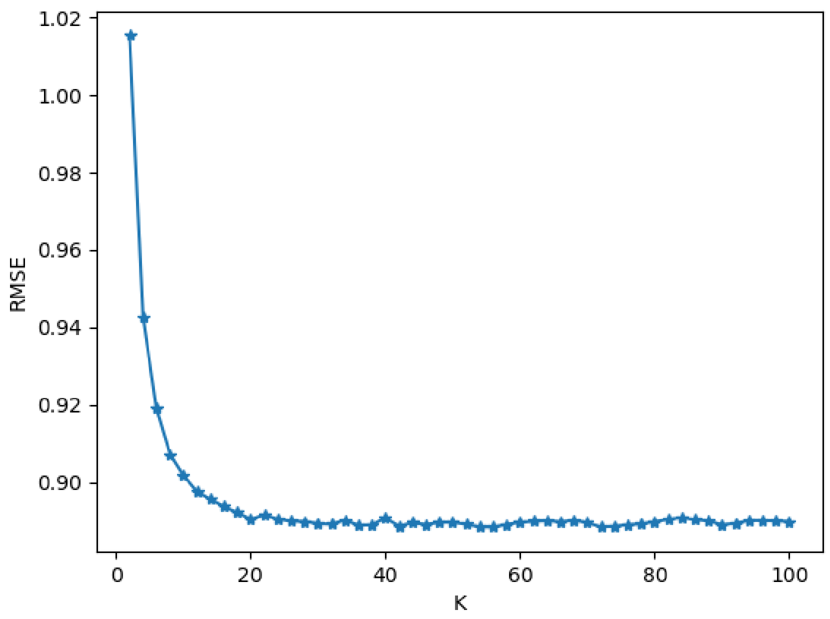

Figure 5: The average RMSE against k closest users for k-NN collaborative filtering.

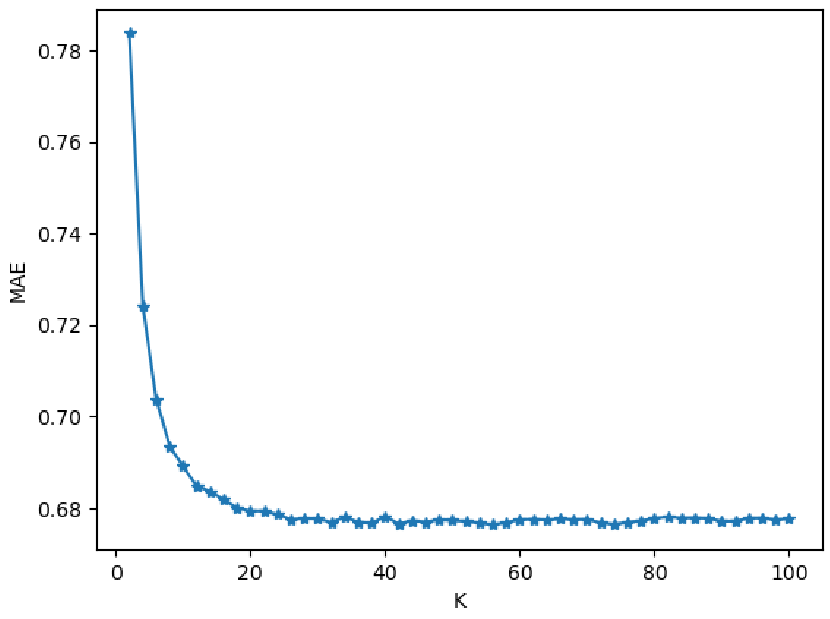

Figure 6: The average MAE against k closest users for k-NN collaborative filtering.

Question 11

According to the definition, the minimum k is the threshold beyond which the increment of k will not result in a significant decrease in the metric. With the help of the figure we obtain in Question 10, we can determine the minimum k for both RMSE and MAE. Both metrics will achieve a steady-state value when k equals 20, hence, 20 may be a good selection for the minimum k. Using the result when $k = 20$, the final value of average RMSE and average MAE are 0.8915 and 0.6796 respectively.

Question 12

The three trimming methods for the test set all rely on the variance or the number of the ratings one movie receives. Thankfully, we have constructed a matrix to store such features of the movies ratings. It is very easy to implement these trimming methods with the help of this matrix. The value of average RMSE varying with different numbers of neighbors are shown as Figure 7. Because we do not need to specify the minimum k in this question, we just plot the trendancy directly. The minimum RMSE for popular trimmed test is 0.8719. Notice that the tendency for popular trimmed test set is similar to the untrimmed test set, meaning

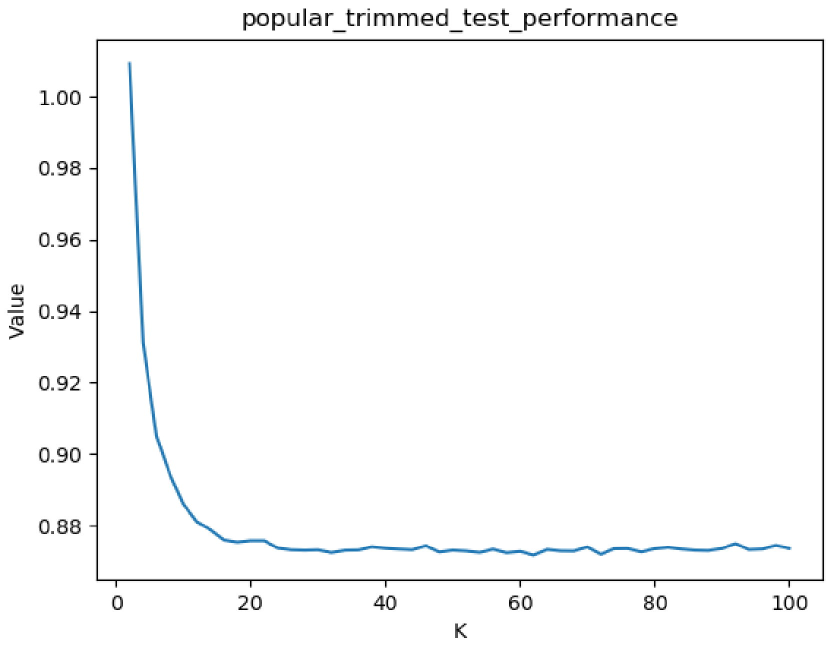

Figure 7: The average RMSE against k for k-NN collaborative filtering based on popular trimmed test set.

that the reduction RMSE of the whole dataset mainly result from the better prediction for popular movies.

Question 13

Similarly, we can get the average RMSE against different numbers of the neighbors as Figure 8. According to the graph, we can get the minimum value of average RMSE is 1.1063. Moreover, the irregular trend of the RMSE against k means that the prediction of unpopular movies does not depend on the choice of k. In

Question 13 continued on next page. . .

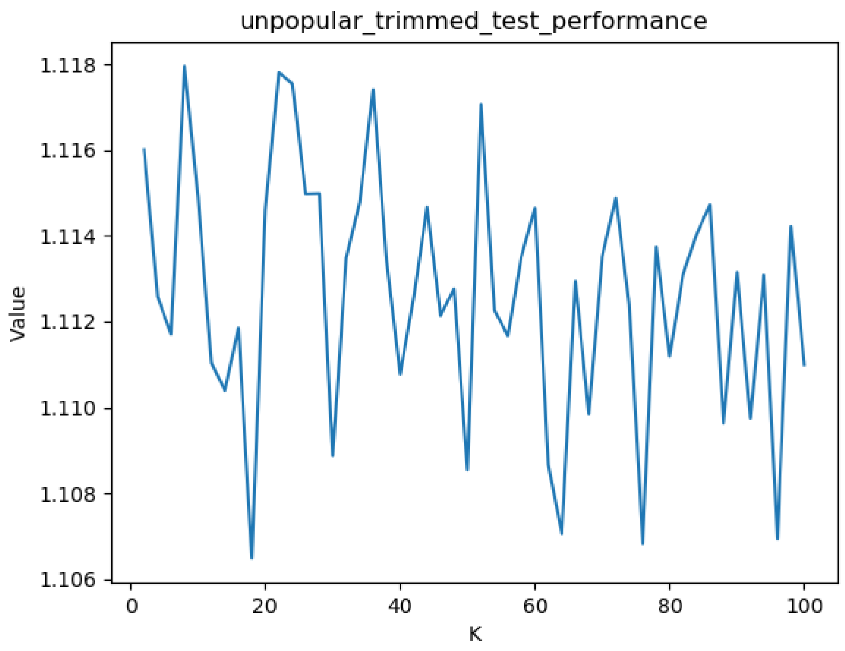

Figure 8: The average RMSE against k for k-NN method based on unpopular trimmed test set.

other words, the prediction of unpopular movies is always inaccurate. This may result from the small part of the users we can use to predict the ratings.

Question 14

The corresponding result of RMSE varying with the choice of k is shown as Figure 9. The tendency of RMSE with k is similar to the result based on unpopular trimmed test set. However, the cause resulting in such tendency of unpopular trimmed test set may not be the reason for the one of high variance trimmed test set. The imprecise prediction might be caused by the total different views about the high variance movies from the similar users. Hence, the prediction algorithm is hard to determine the appropriate value for the rating.

Question 15

As mentioned in Question 11, we are going to choose minimum k as 20. Hence, we will use at most 20 nearest users' ratings to predict the rating one would like to give. The ROC curves for the k-NN collaborative filter with different thresholds are shown as Figure 10, 11, 12, 13. The corresponding AUC for these ROC curve are shown at the lower right corner of the figures.

According to the figures, there are slight difference between the AUC and ROC curve for the threshold values. It means that selecting the threshold for like and dislike elaborately will not change the performance of our method. The way to improve our prediction is either to change the collaborative filter or select the k appropriately.

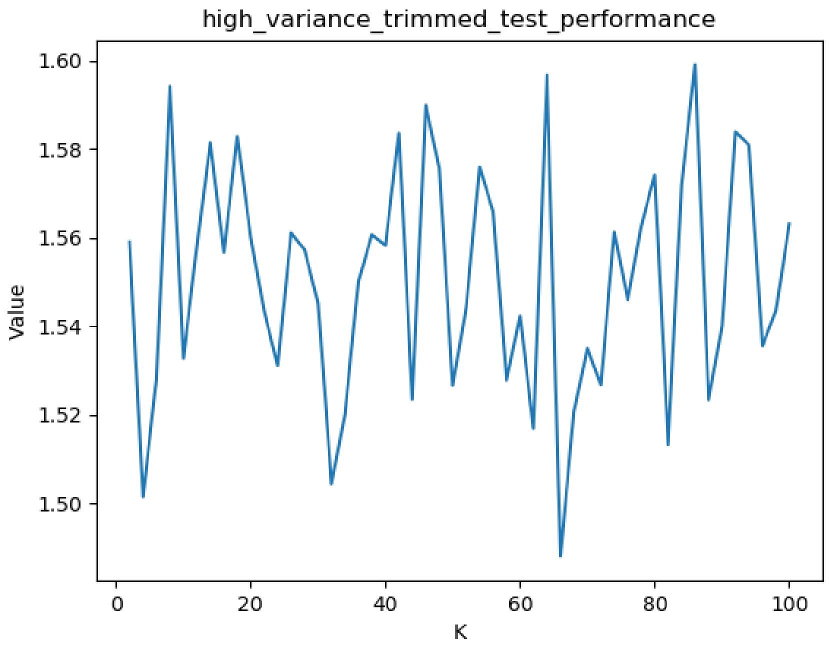

Figure 9: The average RMSE against k for k-NN method based on high variance trimmed test set.

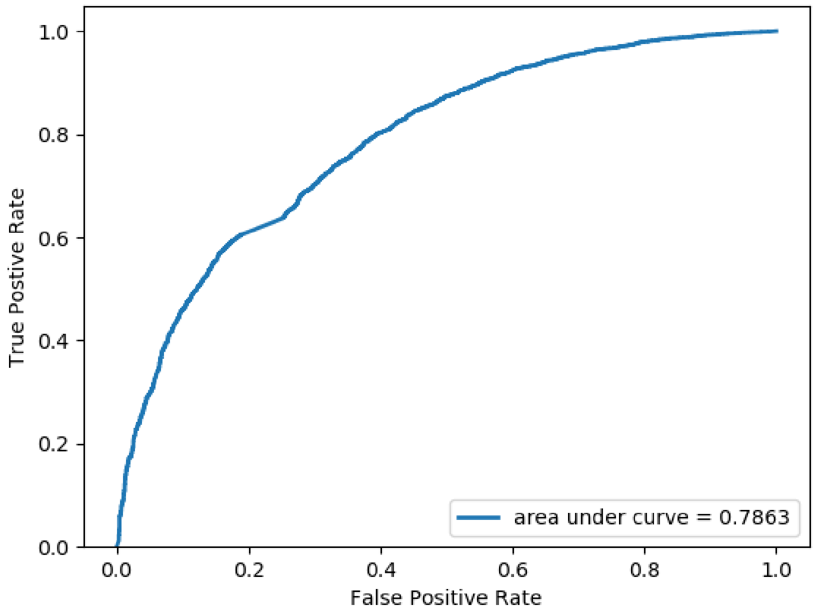

Figure 10: The ROC curve for the k-NN method for threshold values as 2.5

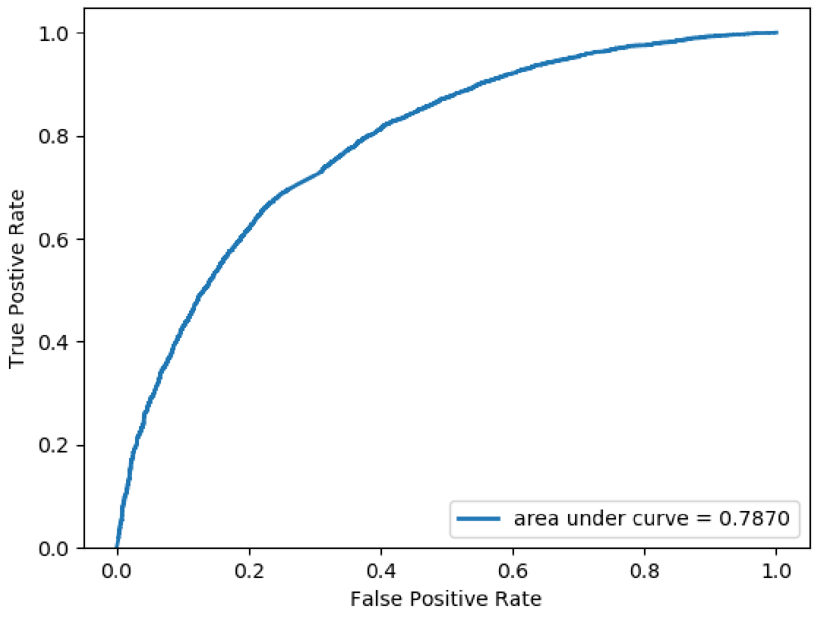

Figure 11: The ROC curve for the k-NN method for threshold values as 3

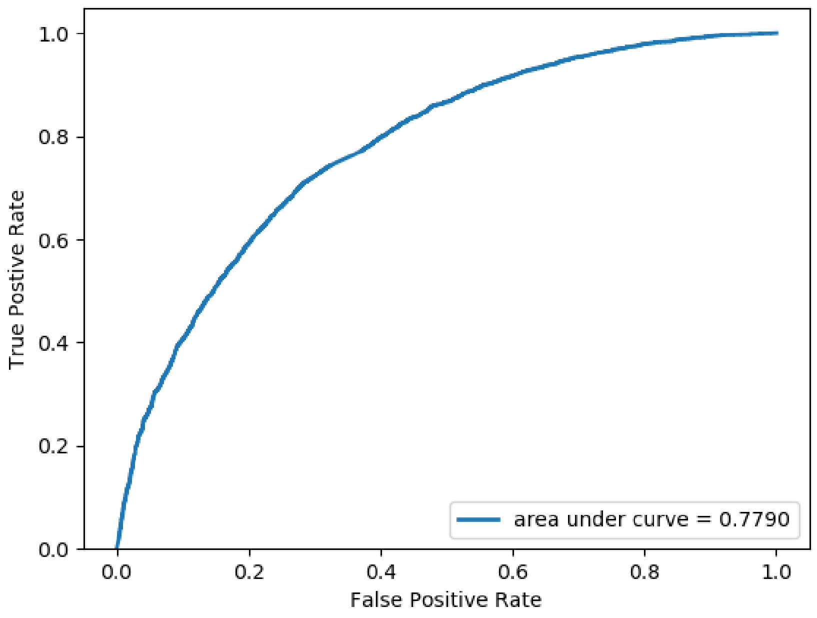

Figure 12: The ROC curve for the k-NN method for threshold values as 3.5

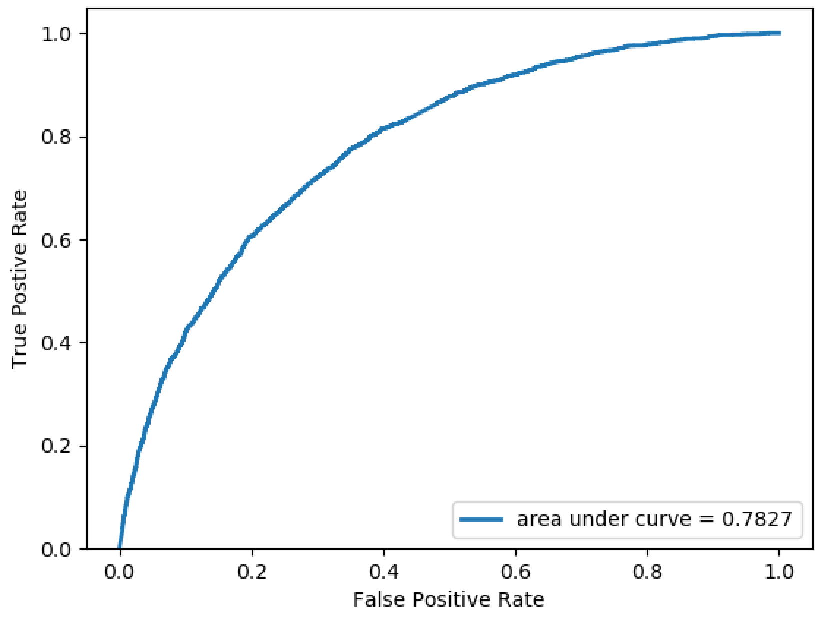

Figure 13: The ROC curve for the k-NN method for threshold values as 4

Question 16

Unfortunately, the optimization problem is NOT a convex problem. A function which has more than one global minima is highly likely to be non-convex. Suppose there is a pair of (U, V) that attains the minimum of our objective, then for any invertible and symmetric matrix B, the pair (UB, VB^{-1}) also achieves the minimum of the same objective. If the objective function is convex, then it must be constant on the line segment connecting these two points, where it is not.

It also can be shown that the Hessian Matrix of this equation is not positive definite. If we set U fixed, then the optimization problem becomes:

$$\min_{V} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}[y - \hat{y}]^2 \tag{16.1}$$

where $y = r_{ij}$ and $\hat{y} = UV^T$. That is the loss function of a least squares problem.

Question 17

Designing a NNMF-based collaborative filter is simple. We only need to use the NMF function in the $surprise$ package. For evaluating the filter, just like Question 10, cross validation of 10 folds is used to find the best parameters. From 2 to 50 in step sizes of 2, we can calculate the average RMSE and average MAE value for each k and the tendency of the results is shown as Figure 14, where the blue line is the curve of RMSE against k and the yellow one is the curve of MAE.

Question 18

According to the graph in Question 17, the average RMSE and average MAE decrease rapidly when k

Question 18 continued on next page. . .

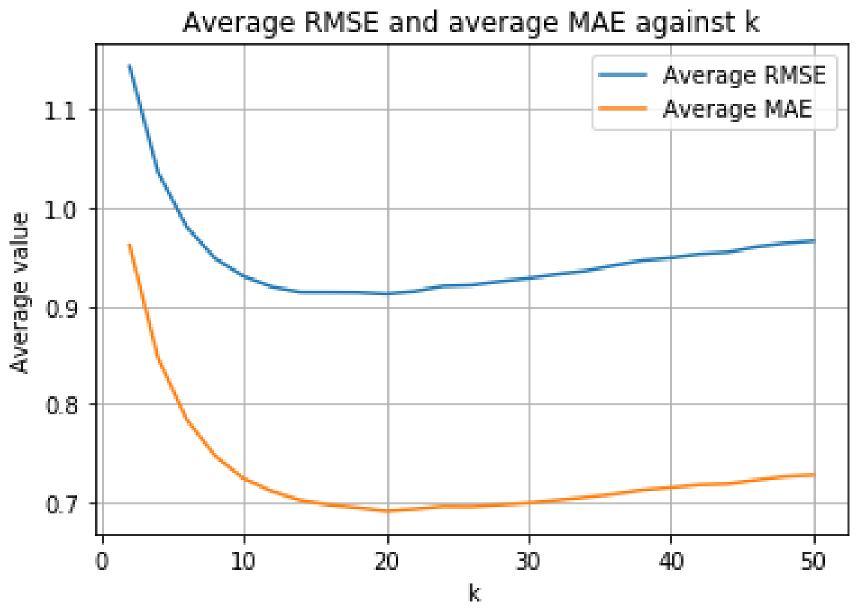

Figure 14: The average RMSE and MAE against k latent factors for NNMF.

increases at first. But then while k continues increasing, the two metrics do not decrease any more but slowly rise. So, the minimum average RMSE and MAE are 0.9123 and 0.6915 respectively, which achieve when $k = 20$. If including IMAX, there are 19 different genres in the dataset, which is almost same as the optimal k.

Question 19

The value of average RMSE against different numbers of latent factors based on popular trimmed test set is shown as Figure 15. Notice that the tendency for popular trimmed test set is similar to the untrimmed test set, which again prove that the popular movies are much easier for prediction.

Question 20

Figure 16 shows the tendency of RMSE vary with the number of latent factors for the unpopular movies set. Generally speaking, there is a tendency that the RMSE decreases with k. This may result from that the more latent factors are introduced, the more accurate for the algorithm to predict the irregular ratings.

Question 21

The corresponding result of RMSE varying with the choice of k is shown as Figure 17. Unlike the result based on the popular and unpopular trimmed test set, the tendency of RMSE with k in is unclear.

Question 22

We plot the ROC curves using the latent factor k=20 as found in question 18, with threshold values 2.5, 3, 3.5, and 4. We plotted each curve on the same graph for comparison. AUCs are also shown on the

Question 22 continued on next page...

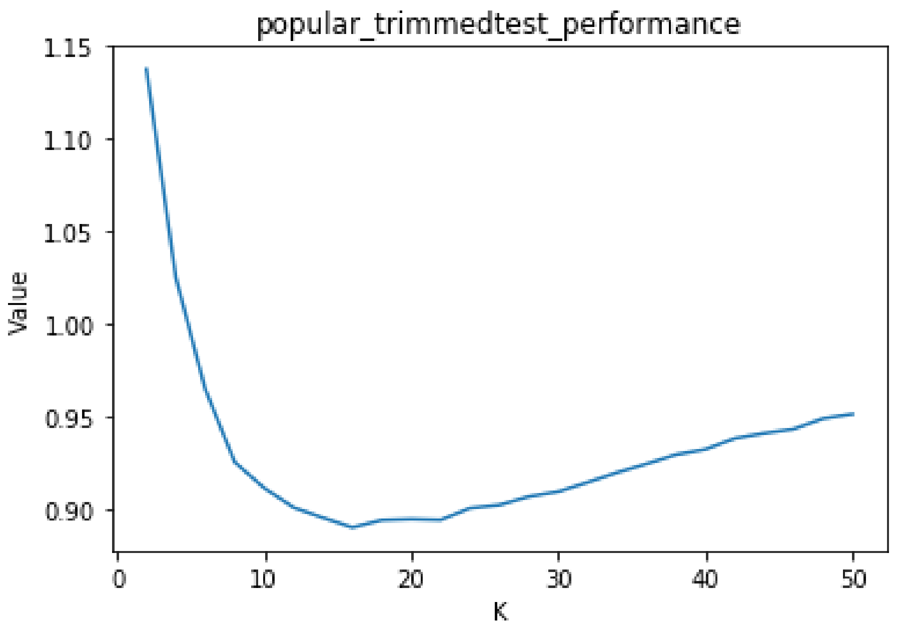

Figure 15: The average RMSE against k latent factors for popular trimmed test set based on NNMF method.

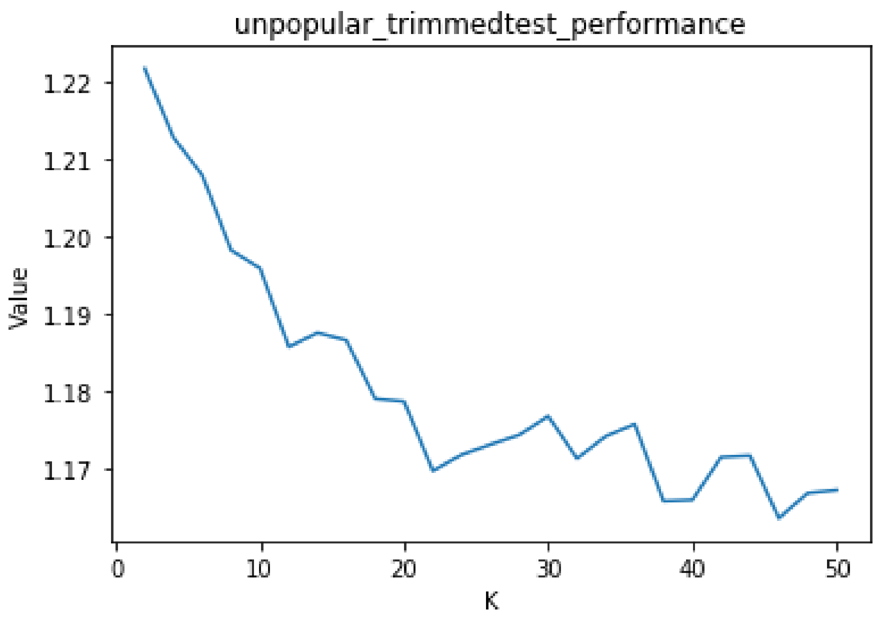

Figure 16: The average RMSE against k latent factors for unpopular trimmed test set based on NNMF method.

Question 22 continued on next page. . .

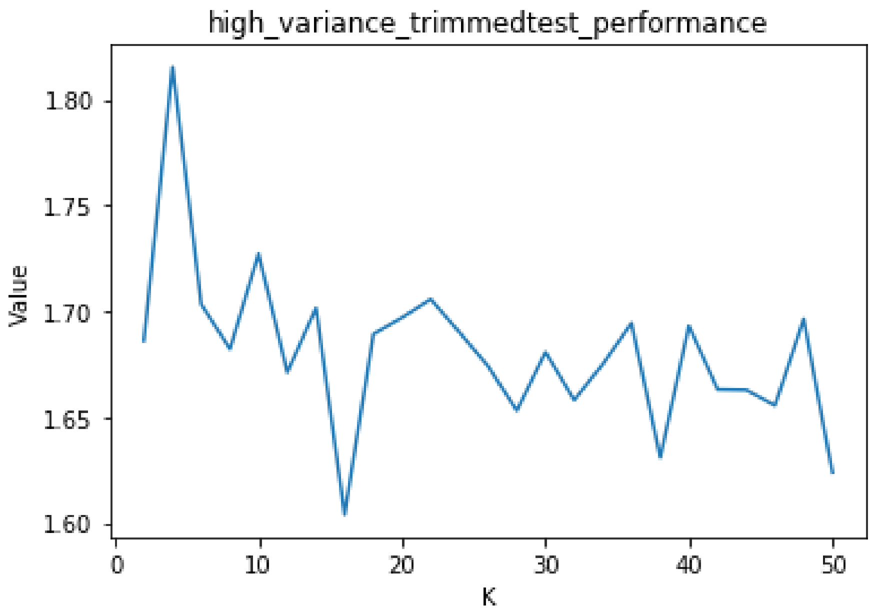

Figure 17: The average RMSE against k latent factors for unpopular trimmed test set based on NNMF method.

bottom-right of the graph: with the threshold being 2.5, 3, 3.5 and 4, the areas under the curve are 0.7649, 0.7714, 0.7573 and 0.7542. The corresponding figure is shown as Figure 18.

Question 23

The top 10 movie genres and their corresponding values for latent factors 0 $\sim$ 4 is shown as Table 1, Table 2 and Table 3. The reason we are going to separately show these results is that the width of our pdf file is limited and is to exhibit two latent factors results parallel at most. From the tables demonstrated here, We can see that for latent factor 0, 4 of them include drama; for latent factor 1, 4 out of 10 include Comedy. So it?s likely that they correspond to drama and comedy genres respectively. The same thing happens to all columns. Each column/latent factor has movie genres with a high similarity.

Order	Latent Factor 0		Latent Factor 1	
	Movie Genres	Corresponding Value	Movie Genres	Corresponding Value
1_{st}	Comedy|Romance	1.8412	Drama	2.6002
2_{nd}	Drama	1.7950	Horror|Thriller	2.5262
3_{rd}	Drama|Thriller	1.7008	Children|Comedy	2.0855
4_{th}	Drama|Mystery	1.6124	Comedy|Drama|Romance	2.0732
5_{th}	Mystery|Sci-Fi|Thriller	1.5691	Drama	1.9909
6_{th}	Children|Drama	1.4767	Children|Comedy	1.8206
7_{th}	Comedy|Musical|Romance	1.4723	Drama	1.7607
8_{th}	Crime|Thriller	1.4504	Crime|Drama	1.7386
9_{th}	Action|Mystery|Thriller	1.4352	Comedy|Crime	1.6478
10_{th}	Comedy	1.4083	Crime|Drama	1.6176

Table 1: Top Movie Genres and their corresponding values for Latent factor 0$\sim$1.

Question 23 continued on next page. . .

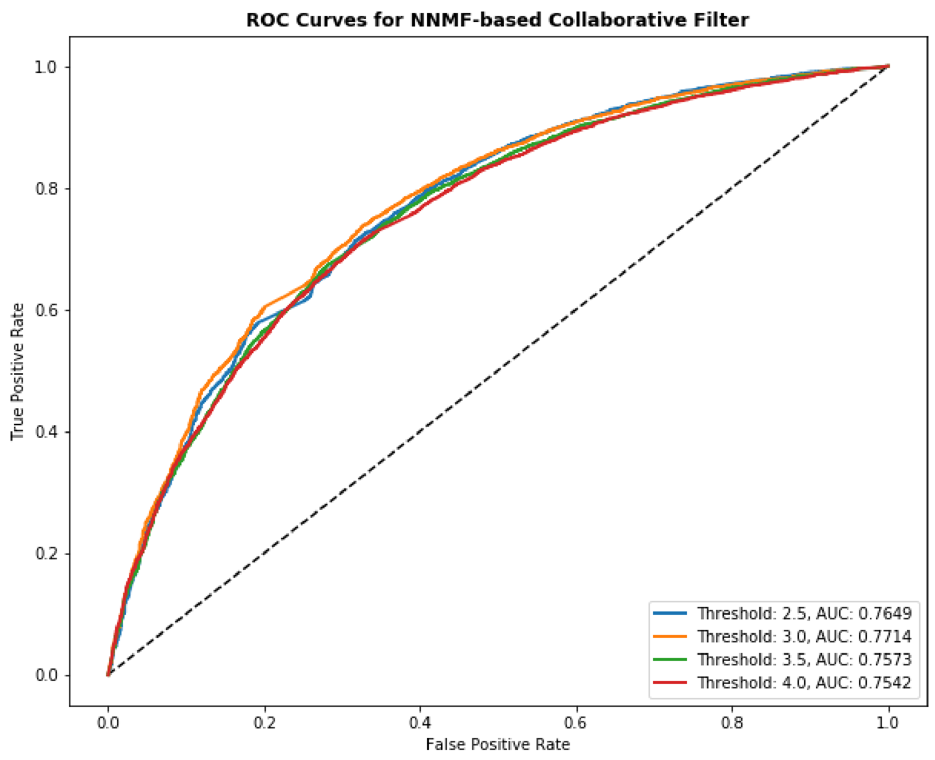

Figure 18: The ROC curves for NNMF-based collaborative filter with different thresholds.

Question 24

We use 10-fold cross-validation to evaluate the performance of MF with bias method and the quality of the rating predictions. We calculate the average RMSE and average MAE for each k with values 2, 4, 6, ?, 50 across 10 folds. Based on the Figure 19 and Figure 20 shown below, we can find the best k to use in this case.

Question 25

The Optimal number of latent factors is 48, because at k=48, it has the minimum average MAE. Although it's not the minimal for RMSE, it is very close to being the best. So considering both RMSE and MAE, we choose the optimal number of latent factors to be 48.

Question 26

A similar procedure is performed on popular movie trimmed test set as in Question 25. For each k = 2, 4, 6, ..., 50, average RMSE is computed across 10 folds and reported below. Based on the graph, we can see that the minimum average RMSE after popular movie trimming is 0.858. The corresponding result is shown as Figure 21.

Question 27

A similar procedure is performed on unpopular movie trimmed test set. For each k = 2, 4, 6, ..., 50, average

Question 27 continued on next page. . .

Order	Latent Factor 2		Latent Factor 3	
	Movie Genres	Corresponding Value	Movie Genres	Corresponding Value
1_{st}	Drama	1.9559	Horror|Thriller|IMAX	2.0007
2_{nd}	Crime|Drama	1.9369	Comedy	1.9209
3_{rd}	Film-Noir|Mystery|Thriller	1.7263	Action|Crime|Drama	1.7032
4_{th}	Drama|Romance	1.7021	Drama	1.6480
5_{th}	Drama|Musical|Romance	1.6778	Crime|Fantasy|Horror	1.5948
6_{th}	Comedy|Horror|Thriller	1.6266	Action|Sci-Fi	1.5636
7_{th}	Drama|Fantasy|Romance	1.5260	Comedy|Romance	1.5381
8_{th}	Comedy|Romance	1.5095	Drama	1.5088
9_{th}	Animation|Documentary	1.4823	Action|Adventure|Comedy	1.4628
10_{th}	Drama|War	1.4682	Comedy	1.5589

Table 2: Top Movie Genres and their corresponding values for Latent factor 2∼3.

Order	Latent Factor 4	
	Movie Genres	Corresponding Value
1_{st}	Action|Comedy|Sci-Fi|IMAX	1.9922
2_{nd}	Comedy|Drama|Romance	1.9902
3_{rd}	Action|Adventure|Drama|Romance	1.8771
4_{th}	Comedy|Romance	1.8318
5_{th}	Comedy|Romance	1.7916
6_{th}	Drama|Film-Noir	1.7045
7_{th}	Comedy|Drama|Romance	1.7037
8_{th}	Comedy|Drama|Horror	1.6624
9_{th}	Drama|Horror	1.6243
10_{th}	Action|Thriller	1.6016

Table 3: Top Movie Genres and their corresponding values for Latent factor 4.

Question 27 continued on next page. . .

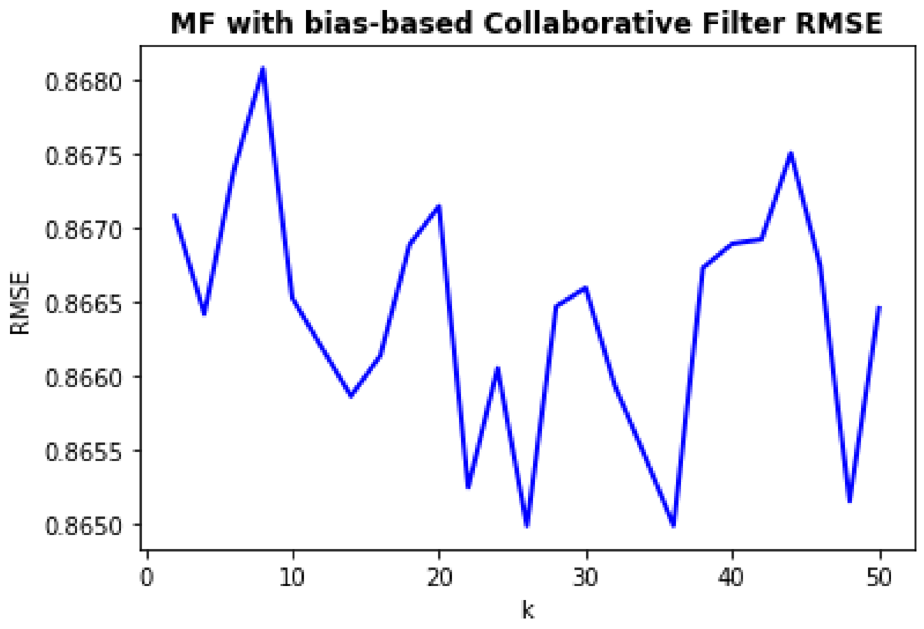

Figure 19: The average RMSE aganist different numbers of latent factors based on MF collaborative filtering.

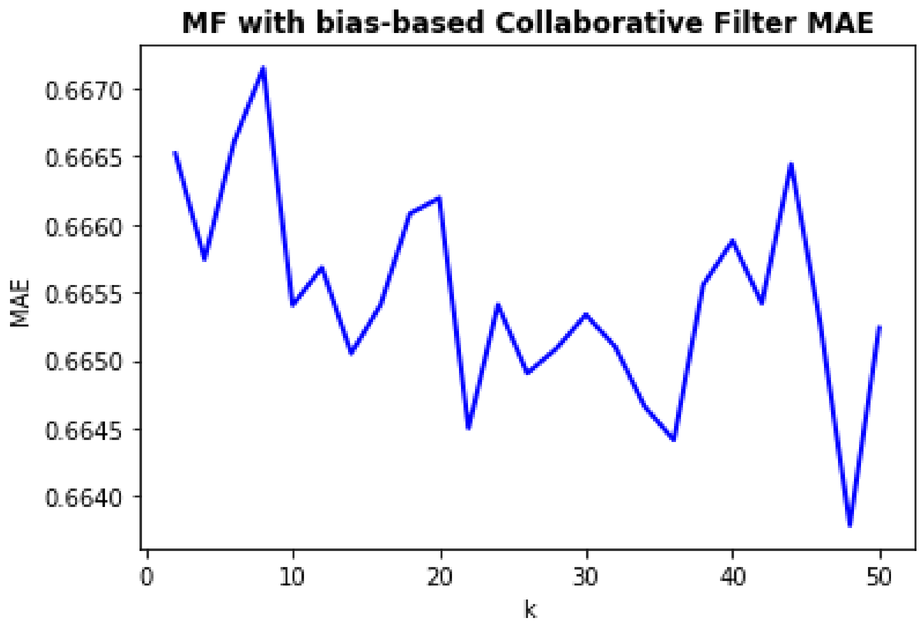

Figure 20: The average MAE aganist different numbers of latent factors based on MF collaborative filtering.

Question 27 continued on next page. . .

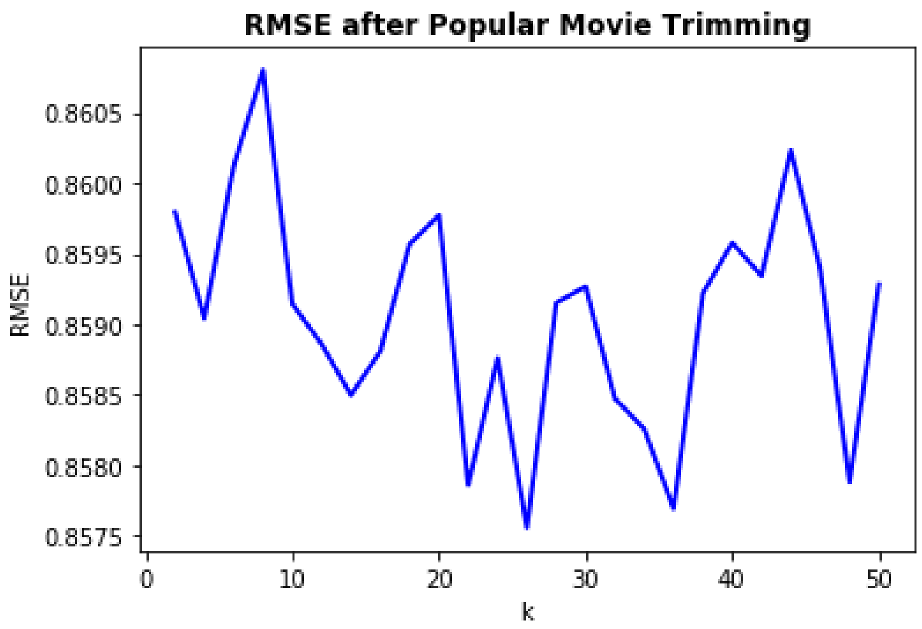

Figure 21: The average RMSE aganist different numbers of latent factors based on MF collaborative filtering for popular movies.

RMSE is computed across 10 folds and reported below. Based on the graph, we can see that the minimum average RMSE after popular movie trimming is 0.972, which is higher than the popular movie trimmed test set. The result can refer to Figure 22.

Question 28

The result is illustrated as Figure 23. Based on the graph, we can see that the minimum average RMSE after popular movie trimming is 1.437, which is even higher than both popular and unpopular movie trimmed test sets.

Question 29

We plot 4 ROC curves with threshold values 2.5, 3, 3.5, and 4 for the collaborative filter in question 24 on the same graph for comparison. With the threshold being 2.5, 3, 3.5 and 4, the areas under the curve are 0.7880, 0.7925, 0.7767 and 0.7735, which are shown on the bottom-right of the graph. On a general view, the differences for each threshold are not large, so you may not see much differences if you plot them on 4 different graphs. The corresponding result is shown as figure 24.

Question 30

Starting from this problem, we are going to design a naïve collaborative filter to predict the ratings of the movies in the MovieLens dataset. Specifically, the prediction function is defined as follows: the predicted rating of user i for item j, denoted by $\hat{r}_ij$, is given by:

$$\hat{r}_ij = \mu_i \tag{30.1}$$

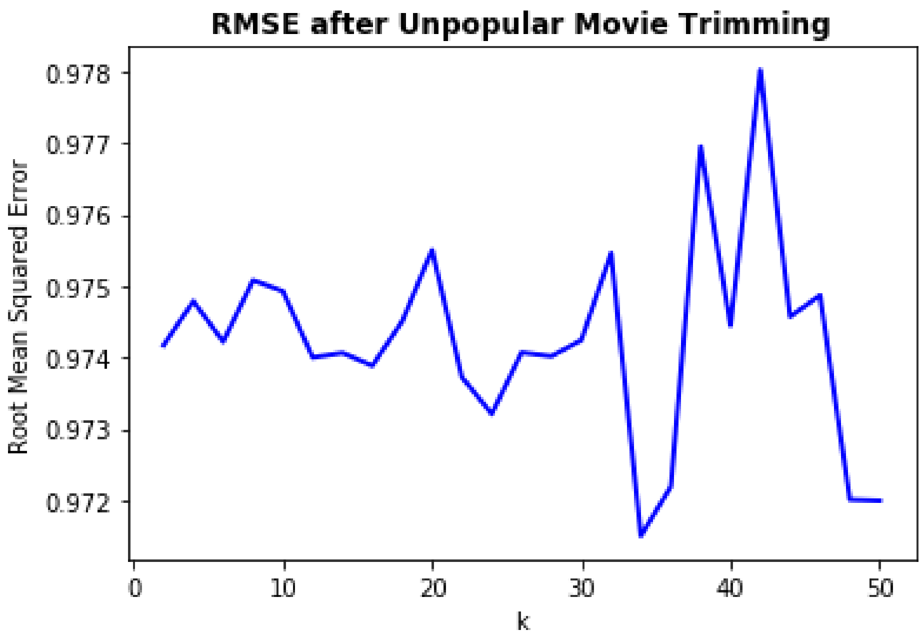

Figure 22: The average RMSE aganist different numbers of latent factors based on MF collaborative filtering for unpopular movies.

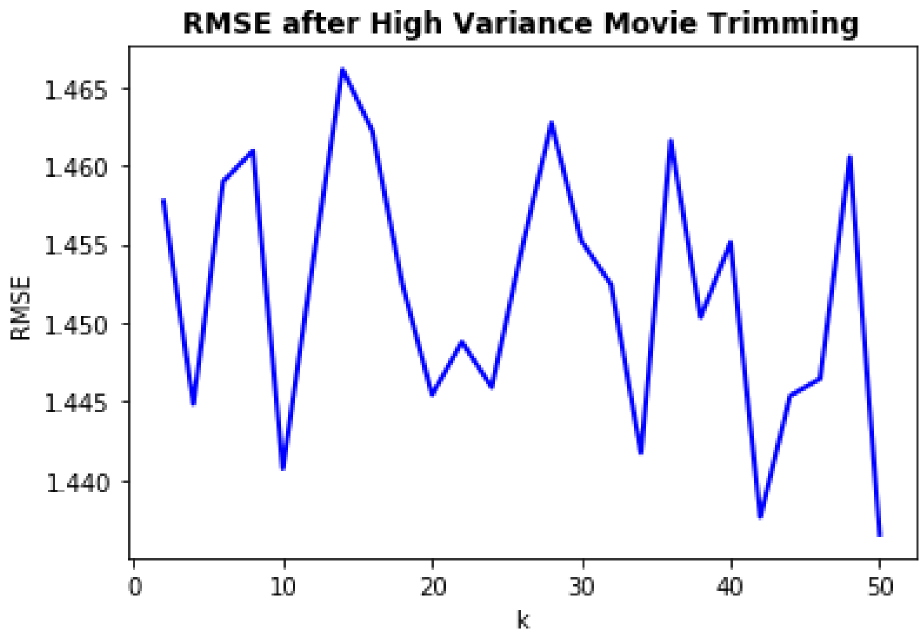

Figure 23: The average RMSE aganist different numbers of latent factors based on MF collaborative filtering for high variance movies.

Question 30 continued on next page. . .

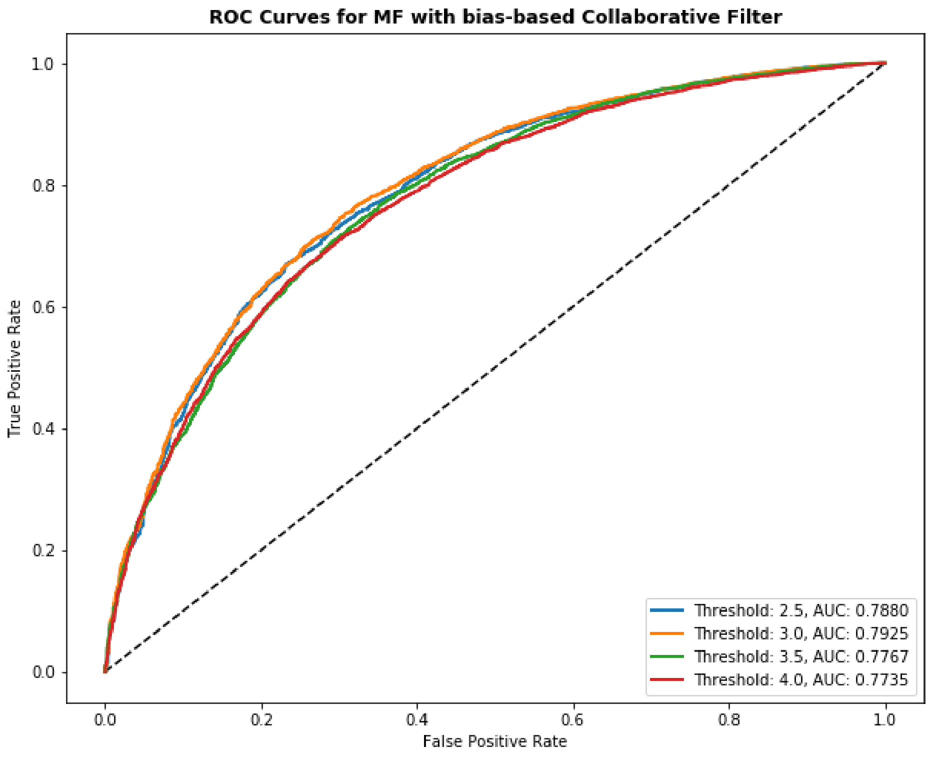

Figure 24: The ROC curves for MF with bias collaborative filter considering various thresholds.

where μ_i is the mean rating of user i.

For this particular question, we are going to use this naïve collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross-validation, and compute the average RMSE value. The average RMSE computed turns out to be: 0.9409.

Question 31 $\sim$ 33

With the naïve collaborative filter designed in the previous question, in these three questions, we will test the performance of this filter by predicting the ratings of the movies in the trimmed test set, which is discussed in the previous questions, and we will use the same trimming operations as before. Again, we will evaluate the performance using 10-fold cross-validation and compute the average RMSE for each of the trimmed test set. The results are shown as Table 4:

Trimmed Test Set	Average RMSE value
Popular Movies	0.9390
Unpopular Movies	0.9752
High Variance Movies	1.4529

Table 4: Average RMSE value for different trimmed test set.

From this table, we can see that the average RMSE for the Popular Movie set is slightly lower than the original set, and the average RMSE for the Unpopular Movie set is slight higher than the original set, however the average RMSE for the High Variance Movie set is much higher than the original set, which is as expected.

Question 34

In this question, we will plot the ROC curves with threshold = 3 for the kNN, NNMF and MF with bias based collaborative filters in the same figure, in order to compare the performance of these collaborative filters. The plot can refer to Figure 25.

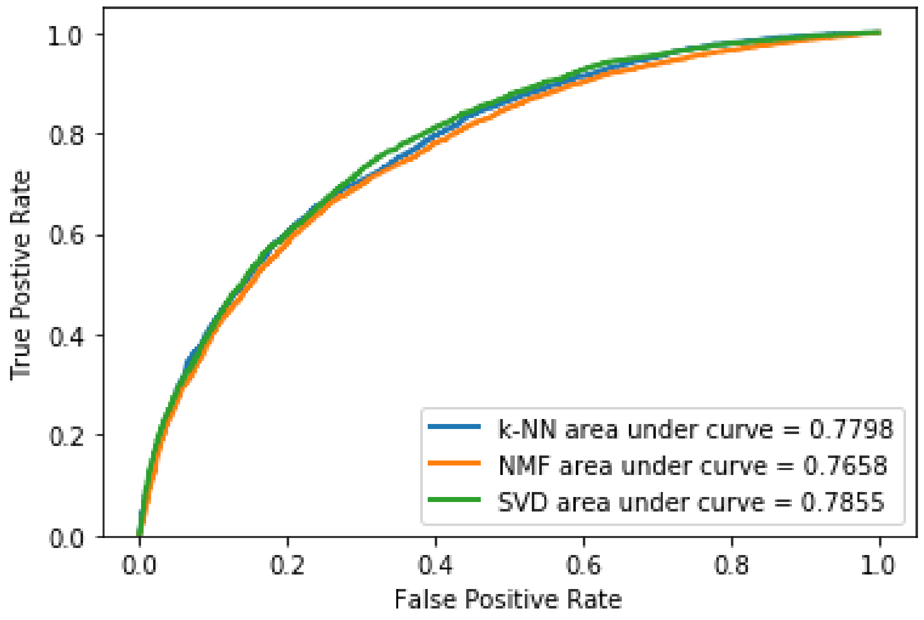

Figure 25: ROC curves for different collaborative filtering methods with threshold = 3.

We can see that when considering the area under the curve, SVD method has the largest area, thus the best performance, then k-NN method has the second largest area, and lastly NMF method has the smallest area, thus poor performance compare to SVD or k-NN method.

Question 35

Starting from this question, we will explore ways to solving the ranking problem. To begin with, a recommendation system can be formulated into two ways:

- Predict the rating value for a user-item combination (Prediction)
- Recommend the top k items for a particular user (Ranking)

In this part, we will mainly be focusing on the ranking version of this problem. In particular, our approach will be 'Solve the prediction problem and then rank the predictions'. The detailed procedure is discussed in the project handout, thus we will not discuss it again in this section. However, one thing needs to be discussed is that using precision-recall curve to evaluate ranking. The equations to compute Precision and Recall are given as:

$$Prediction(t) = \frac{|S(t) \cap G|}{|S(t)|} \tag{35.1}$$

Question 35 continued on next page…

$$Recall(t) = \frac{|S(t) \cap G|}{|G|} \tag{35.2}$$

where $S(t)$ is the set of items of size t recommended to the user, and G is the set of items liked by the user (or we can say they are ground-truth positives). To explain the meanings of precision and recall in words, consider that Precision answers the question that 'Given all positives you have predicted, how many of them are really positive?' and Recall answers the question that 'Among all positives, how many of them are corrected predicted by you?'. A model can be 'high precision and low recall', 'low precision and high recall', or 'high precision and high recall', and each case is desirable within some specific cases, however, we will not discuss it in detail in this section.

Question 36

In this problem, we are going to plot average precision (y-axis) against t (x-axis) for the ranking obtained using kNN collaborative filter predictions. Also, plot the average recall (y-axis) against t (x-axis) and average precision (y-axis) against average recall (x-axis). We will use the k found in question 11 and sweep t from 1 to 25 in step sizes of 1. Note that t is the size of the recommended list. The plots about the performance of kNN method are shown as Figure26, 27 and 28 respectively:

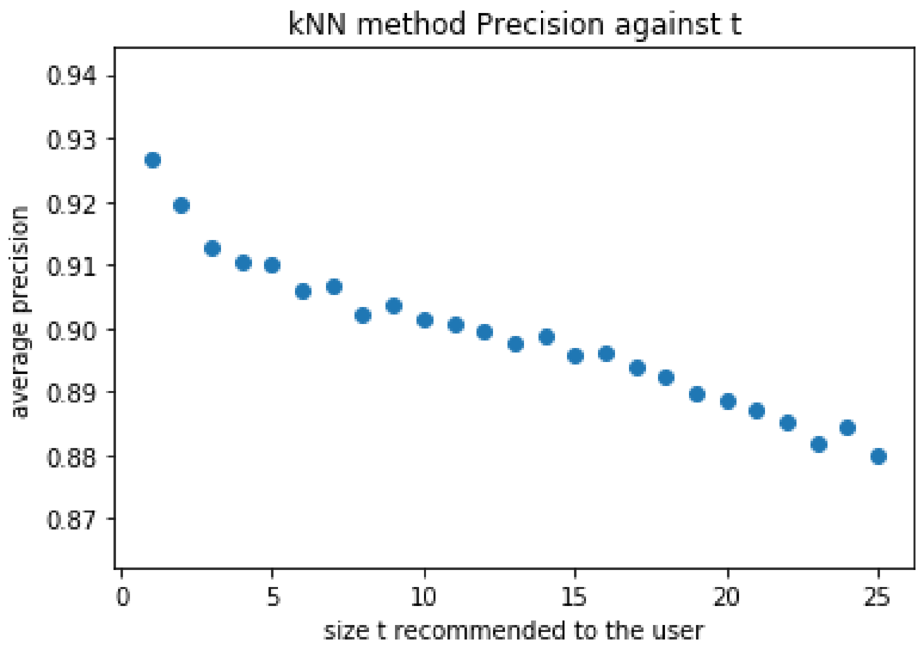

Figure 26: Precision Rate for k-NN method with different size t recommended to the user.

We can see from these plots that in the case of precision, the more movies recommended to the users, the less average precision we will get, and in the case of recall, the more movies recommended to the users, the more average recall we will get, and lastly, average precision goes down as average recall goes up, as expected.

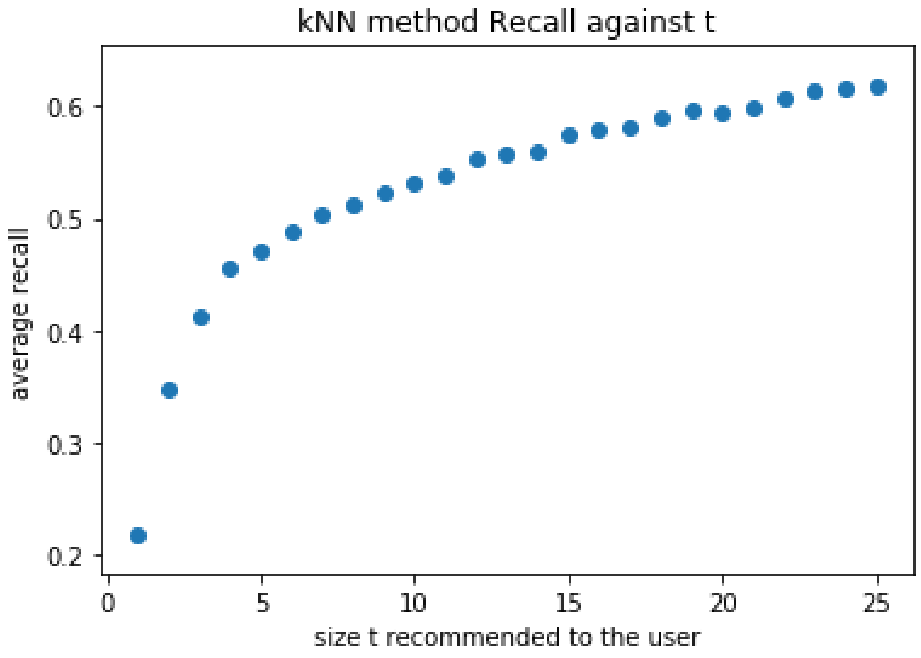

Figure 27: Recall Rate for k-NN method with different size t recommended to the user.

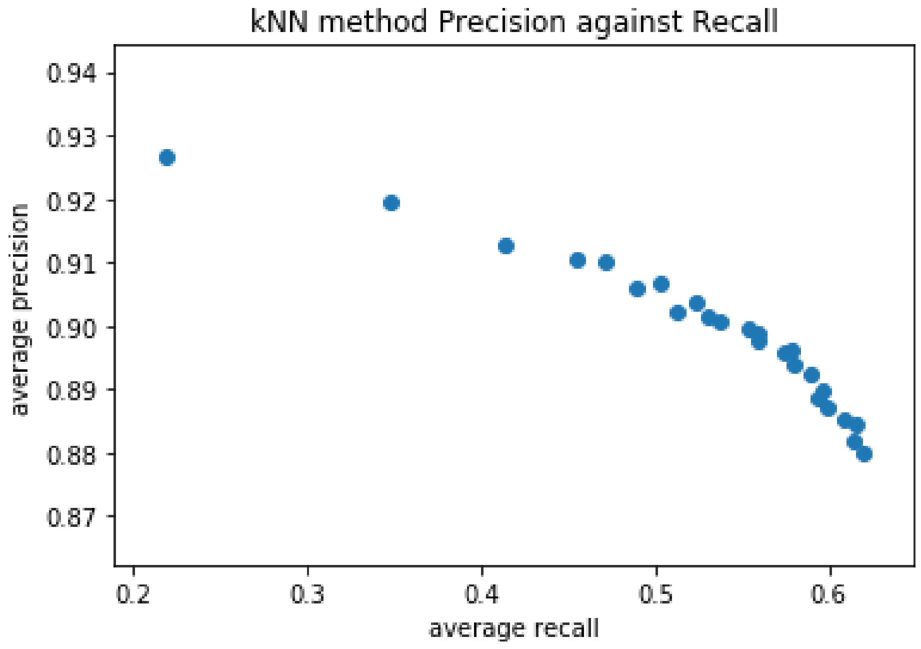

Figure 28: Precision against Recall for k-NN method with different size t recommended to the user.

Question 37

In this problem, we are going to 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). We will use the optimal number of latent factors found in question 18 and sweep t from 1 to 25 in step sizes of 1. The plots can refer to Figure 29, 30 and 31.

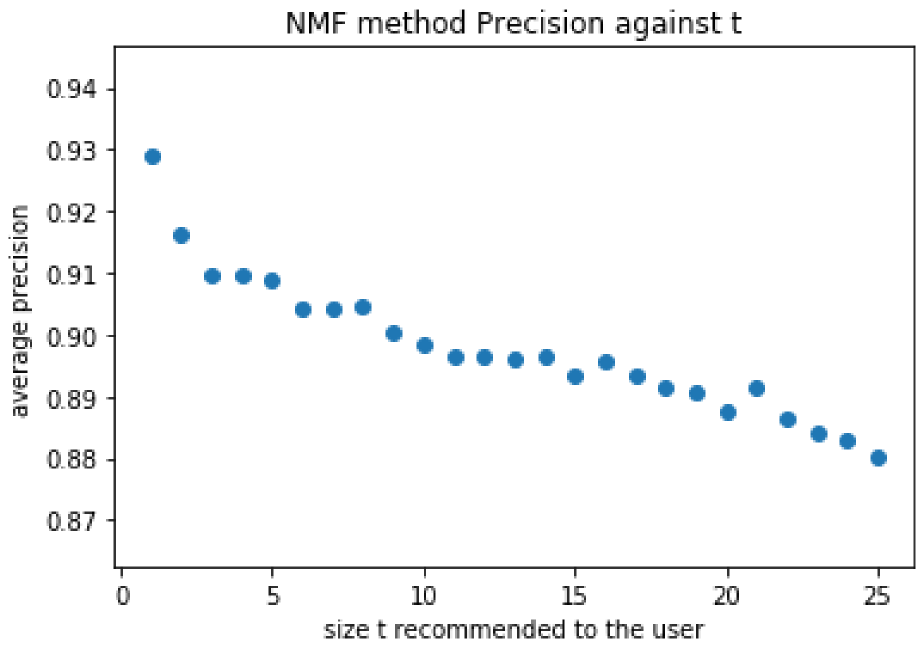

Figure 29: Precision Rate for NNMF method with t items recommended to the user.

We can see from these plots that, the same as the previous method, in the case of precision, the more movies recommended to the users, the less average precision we will get, and in the case of recall, the more movies recommended to the users, the more average recall we will get, and lastly, average precision goes down as average recall goes up, as expected.

Question 38

Similarly to the previous two questions, we are going to use the optimal number of latent factors found in question 25 and sweep t from 1 to 25 in step sizes of 1 to attain the corresponding plots about the performance of SVD methods with different numbers of item recommendation. The result can refer to Figure 32, 33 and 34.

We can see from these plots that, the same as the previous methods, in the case of precision, the more movies recommended to the users, the less average precision we will get, and in the case of recall, the more movies recommended to the users, the more average recall we will get, and lastly, average precision goes down as average recall goes up, as expected.

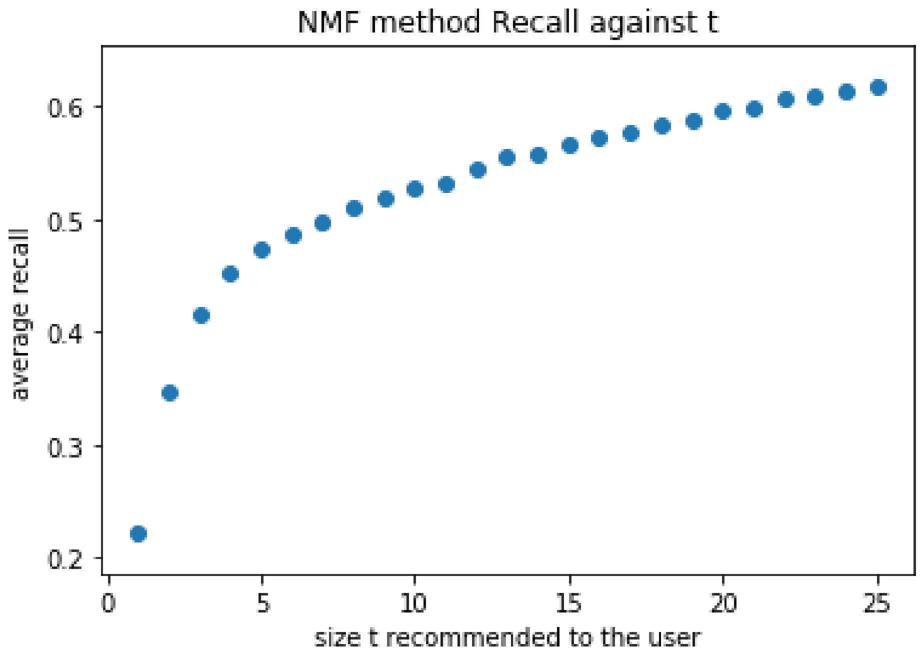

Figure 30: Recall Rate for NNMF method with t items recommended to the user.

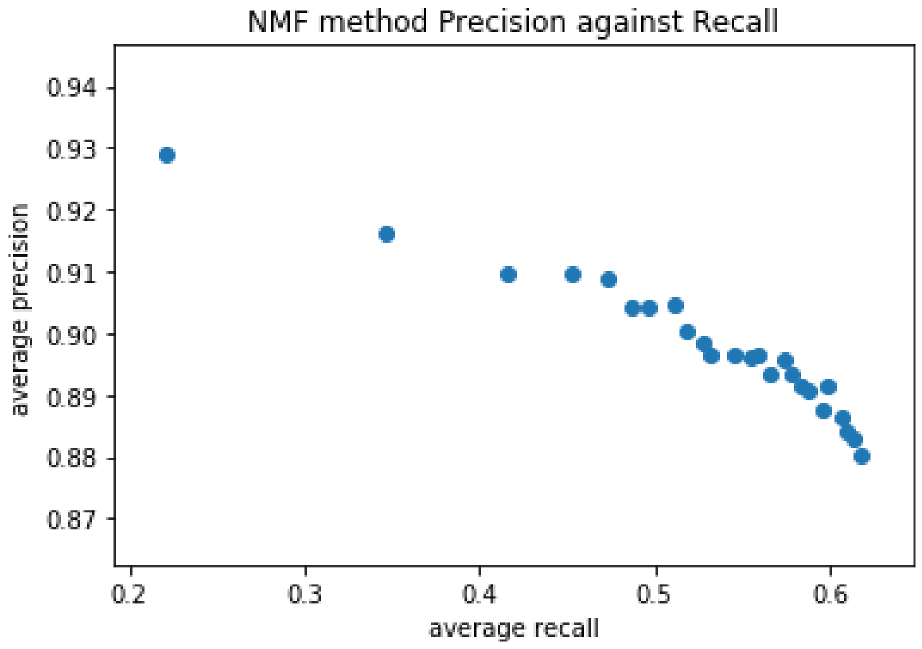

Figure 31: Precision against Recall for NNFM method with t items recommended to the user.

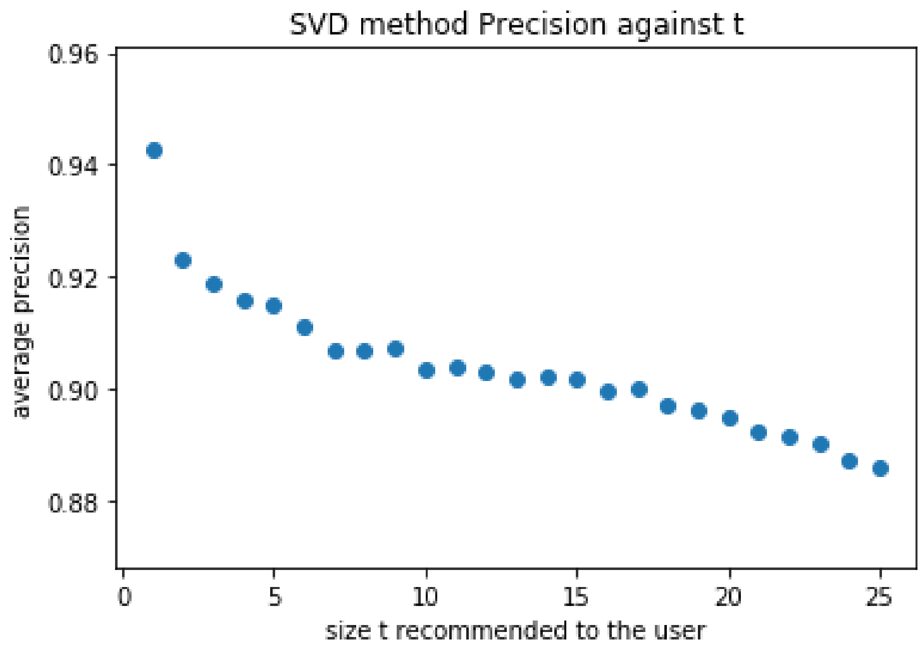

Figure 32: Precision Rate for SVD method against t items recommended to the user.

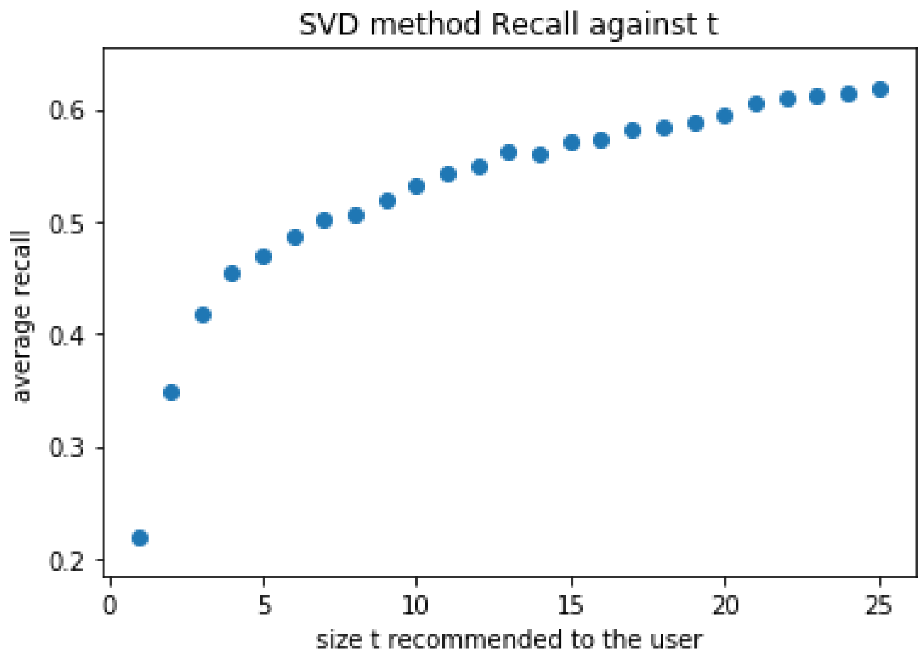

Figure 33: Recall Rate for SVD method against t items recommended to the user.

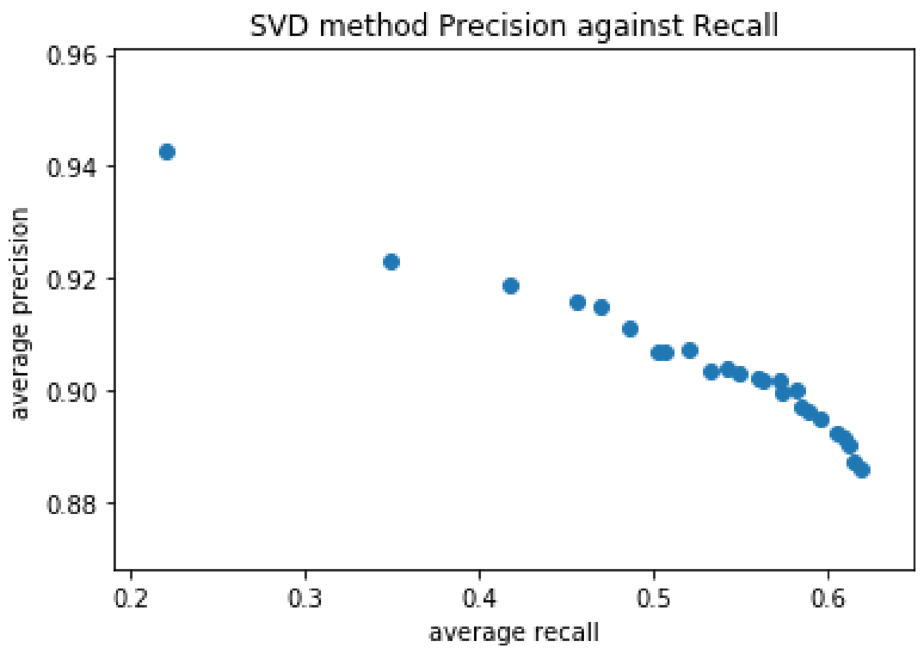

Figure 34: Precision against Recall for SVD method against t items recommended to the user.

Question 39

In this problem, we are going to plot the precision-recall curve obtained in questions 36, 37, and 38 in the same figure, and use this figure to compare the relevance of the recommendation list generated using kNN, NNMF, and MF with bias predictions. The plot is shown as Figure 35.

We can see from the plot, in general SVD method has the best performance, follow by k-NN method, and lastly NMF method. This result coincide with the previous analysis of the performance of these methods.

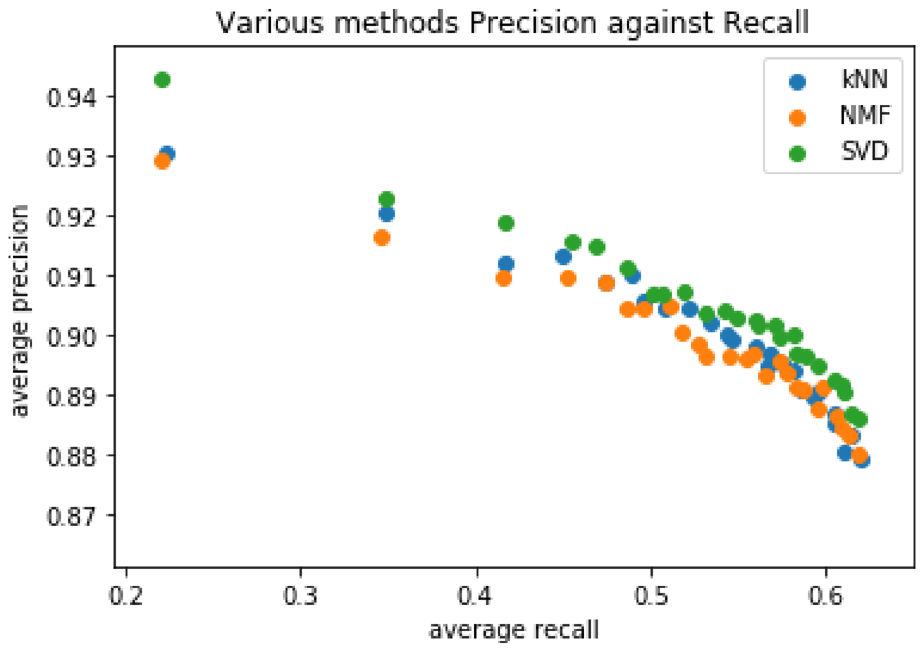

Figure 35: Precision against Recall rate for various methods.