EE 219 Project 3
Collaborative Filtering
Winter 2018

Jui Chang
Wenyang Zhu
Xiaohan Wang
Yang Tang

1 Introduction

The basic idea of recommendation system is to predict customers interest based on the dataset, which is the feedback of users like or dislike an item.

The basic models for recommender systems works with two kinds of data:

1. User-Item interactions such as ratings
2. Attribute information about the users and items such as textual profiles or relevant keywords

Models use type 1 are attributed as collaborative filtering methods. Models use type 2 are attributed to content based methods. In this Project, we build recommendation system using collaborative filtering methods.

2 Collaborative filtering models

Collaborative filtering model is use the collaborative power for multiple users rating on items to make recommendations. The main challenge is that the rating matrices is sparse. The idea of collaborative filtering is that ratings can be estimated because the observed ratings are often highly correlate with users and items.

In this project, we will use and analyze the performance of two types of collaborative filtering methods:

1. Neighborhood-based collaborative filtering
2. Model-based collaborative filtering

3 MovieLens dataset

In this project, we will build a recommendation system to predict the rating of the movies in MovieLens dataset. In this project, we only use movie rating information in the dataset. The rating matrix R, is a mxn matrix, m is the user and n is the movie. r_{ij} is the rating of user I on movie j. In this part, we analyze and visualize some properties of the dataset.

Question 1: Compute the sparsity of the movie rating dataset.

$$\textbf{Sparsity} = 1 - \frac{\boldsymbol{Total\ number\ of\ available\ ratings}}{\boldsymbol{Total\ number\ of\ possible\ ratings}}$$

Result:

The sparsity of this rating dataset is 0.9836, which means that in our rating matrix, 98.36% there are no ratings provided, which is common because there are 671 users and 9066 movies, it is impossible for all users to watch most of the movie and rate on the movie.

Question 2: Plot a histogram showing the frequency of rating values.

Result:

From figure 2 below, we can see the rating of the movie are commonly high, in the rating scale of 0.5 to 5, the ratings are mostly above 3. The top three largest histograms are 4 to 4.5, 4.5 to 5, and 4 to 3.5.

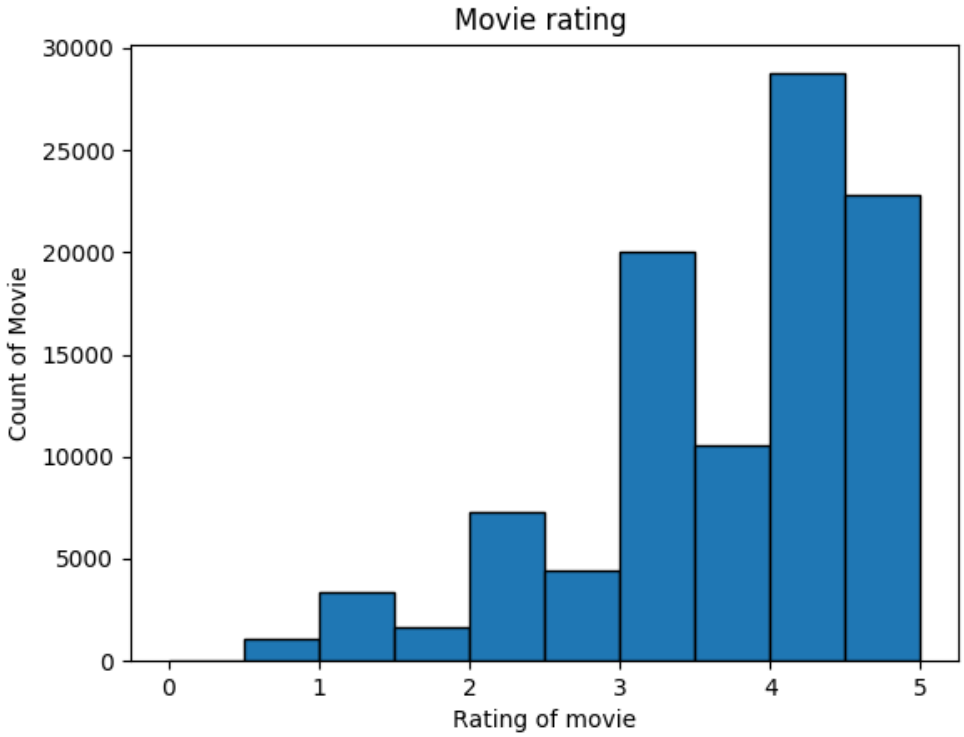


Figure 1. Movie Rating Histogram

Question 3: Plot the distribution of ratings among movies. X-axis should be movie index in the order of decreasing frequency. Y-axis should be the number of ratings the movie received.

Result:

From figure 3 below, we can observe that the top few most rating frequencies occupy the most number of total ratings, which means few movies are popular, and rate by a great number of users, and most movies are not popular, they only received less 40 ratings.

There are too many indices so take the movie index out and shown below.

The top 10 movieId are [356, 296, 318, 593, 260, 480, 2571, 1, 527, 589]

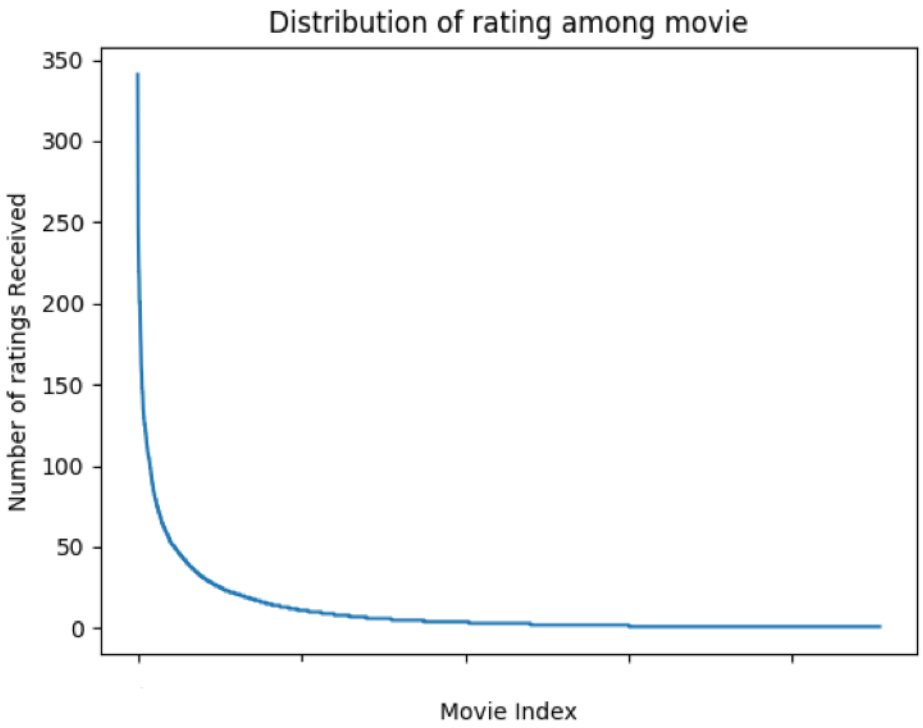


Figure 2. Distribution of Rating Frequency among Movies

Question 4: Plot the distribution of ratings among users. X-axis should be user index in the order of decreasing frequency. Y-axis should be the number of movie user rates.

Result:

From figure 4 below, we can observe that the top few most rating frequencies occupy the most number of total ratings, which means few people rating most of the movie, and most people rate less than 300 of the movie, which is very few compared to total 9066 movies.

There are too many indices so take the movie index out and shown below.

The top 10 userId are [547, 564, 624, 15, 73, 452, 468, 380, 311, 30]

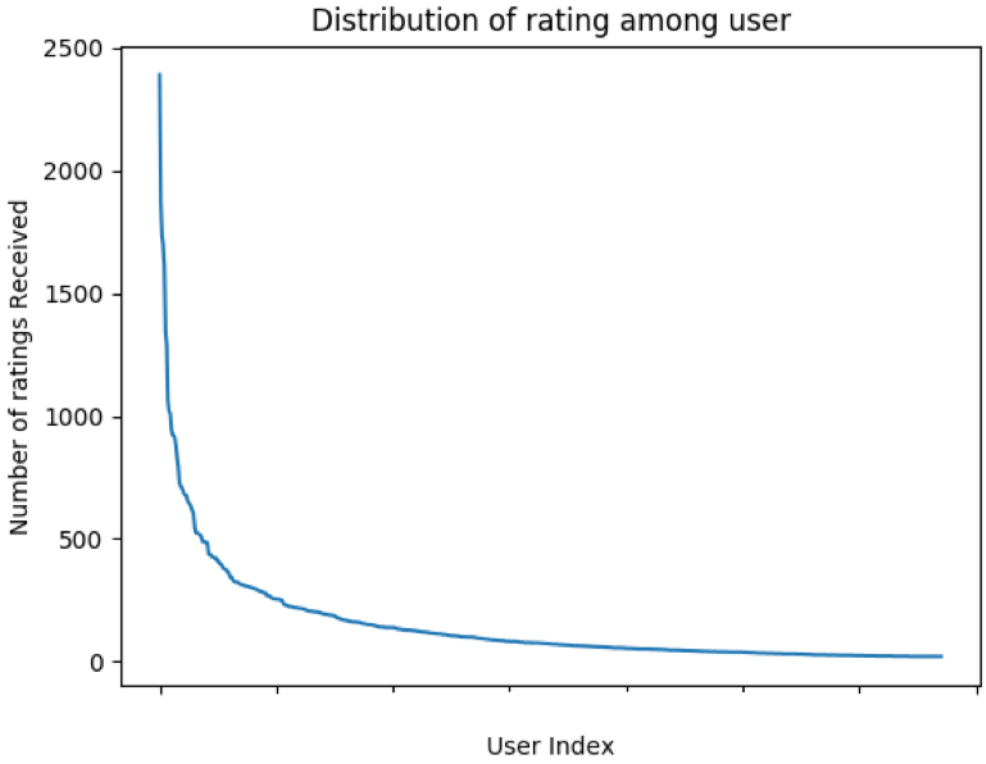


Figure 3. Distribution of Rating Frequency among Users

Question 6: Compute the variance of the rating value received by each movie. Plot a histogram showing the frequency of variance rating values.

Result:

From the figure 2 below, we can see as variance increases, there are less movie. Therefore, we can say that most movie with small variance (less than 1), which means that the ratings of most movies remains in a small range.

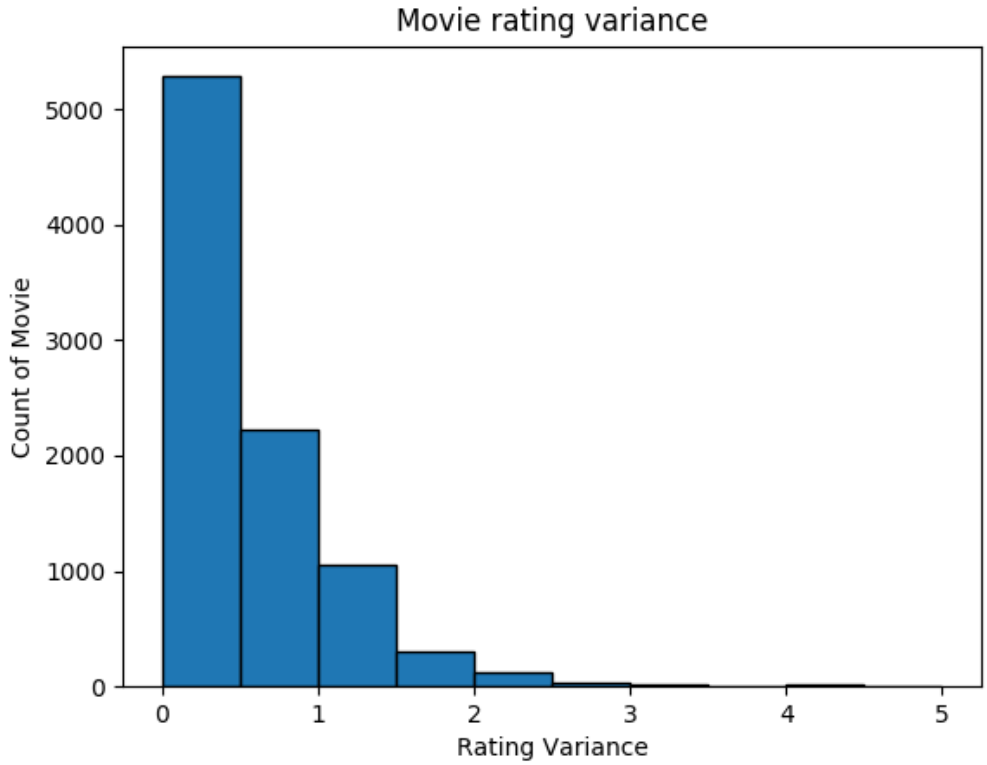


Figure 4. Movie Rating Variance Histogram

Question 5: Explain the salient features of the distribution found in question 3 and their implications for the recommendation process.

Result:

From figure 3 above, we can observe that the top few most rating frequencies occupy the most number of total ratings, which means few movies are popular, and rate by a great number of users. For popular movies, many people rate on them, so it is easier for recommendation system to collect the ratings and predict precisely whether to recommend that movie to the user or not.

4 Neighborhood-based collaborative filtering

The neighborhood-based method is to use either user-user similarity or item-item similarity to make predictions from rating matrix. Two basic principles used in neighborhood-based models as follows. Here we use user-based collaborative filtering.

1. User-based models: people have similar rating on the same item are similar users.
2. Model based models: Items are rated in the similar way by the same user are similar items.

4.1 User-based neighborhood models

User-based neighborhoods are defined to identify similar users to the target user. To decide the neighborhood of target user u, its similarity to other users is computed. Thus, we must find similarity matrix of ratings specified by users. Here we use Pearson-correlation coefficient to computer the similarity between users.

4.2 Pearson-correlation coefficient

Pearson-correlation coefficient between two users, u and v, denoted as Pearson(u,v), shows the similarity between users u and v.

I_u : Set of item indices for which ratings have been specified by user u

I_V : Set of item indices for which ratings have been specified by user v

μ_u: Mean rating for user u computed using her specified ratings

r_{uk}: Rating of user u for item k

$$Pearson(u,v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)(r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}}$$

Question 7: Write down the formula for μ_u in terms of I_u and r_u

Result:

The items u has rating on is the set of I_u, and rating for user u on item k is r_{uk}, so the sum of total rating is $\sum_{k \in I_u} r_{uk}$, and the frequency user u rate is $|I_u|$, so I have the following equation calculate the mean rating for user u.

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

Question 8: In plain words, explain the meaning of $I_u \cap I_v$. Can $I_u \cap I_v = \varnothing$

Result:

$I_u \cap I_v$ is set of the item indices user u and user v share. If user u and user v does not have any set of item indices share, then $I_u \cap I_v = \varnothing$.

4.3 k-Nearest neighborhood (k-NN)

Above we define similarity matrix. Now we define k-Nearest neighbor of users. K-Nearest neighbor of user u, P_u, is the top k user with highest Pearson-correlation to user u.

4.4 Prediction function

We can define prediction function for user-based neighborhood model.

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

In LHS, $\hat{r}_{uj}$ is the predicted rating for user u with item j.

Question 9: Explain the reason behind mean centering of the raw ratings in the prediction functions.

We would like to know how a movie to a user is, but for every user, the rating is different: if some people are stricter, they give all score low, so if we want to know if he/she likes the movie, we should compare to the person's rating average. Therefore, we subtract the raw ratings by the person's average ($r_{vj} - \mu_u$), vice versa.

4.5 k-NN collaborative filter

4.5.1 Design and test via cross-validation

In this part, we will design a k-NN collaborative filter and test its performance with 10-fold cross validation. For a 10-fold cross-validation, a dataset would be spitted to 10 equal-size subsets. Take 9 subset as training set to train the filter and take 1 subset as validation set. The cross-validation is then repeated 10 times, with each subset to be validation set once.

Question 10: Design a k-NN collaborative filter to predict the rating of movies in MovieLens dataset and use 10-fold cross validation to evaluate its performance. Implement k from 2 to 100 with step size 2, and compute RMSE and MAE for each k. Here we use Pearson-correlation function as the similarity metric.

Result:

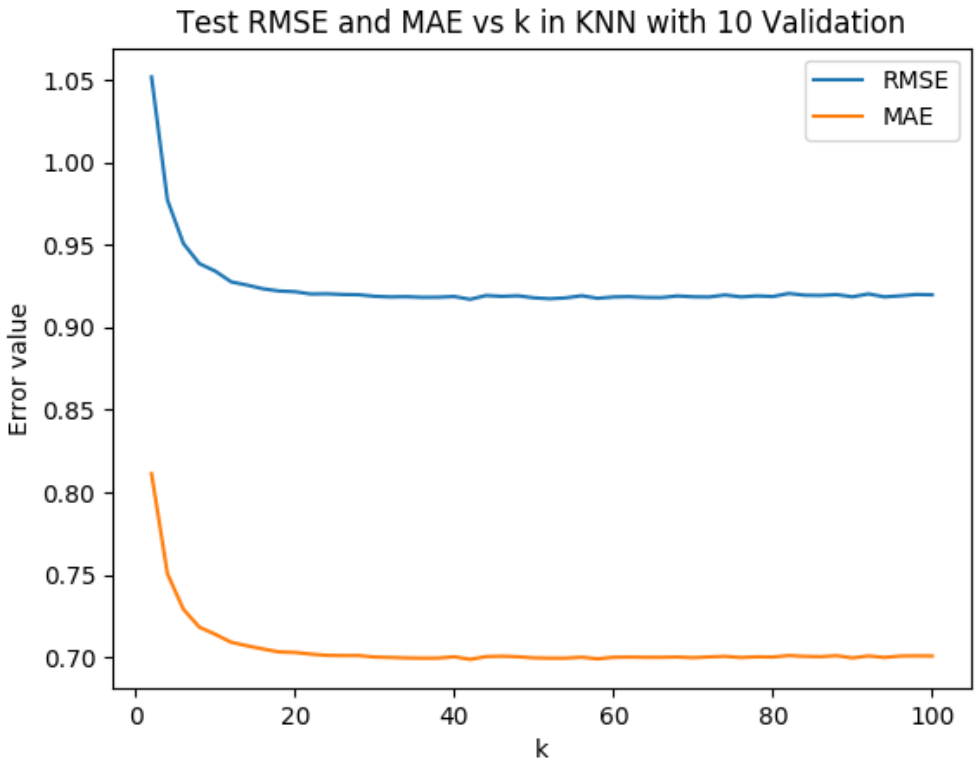


Figure 5. RMSE and MAE vs k

Question 11: Use the plot from question 10 to find a ‘minimum k’

Result:

From figure 10 above, as k increases, RMSE and MAE decrease. When k >24, the RMSE and MAE reaches a steady state. When increase k above 24, where RSME = 0.9201, MAE = 0.701, RMSE and MAE would not drop significantly as k increases. Therefore, we choose minimum k= 24. We take k=24 as optimal k because increasing k would not make filter perform well, and larger k means more computing time and more cost.

4.6 Filter performance on trimmed test set

In this part, we will perform k-NN collaborative filter to examine the performance in predicting the ratings of movies in the trimmed dataset. The test set can be trimmed as follows.

- Popular movie trimming: In this trimming, in the test set, we keep the data(Movie) with more than 2 ratings in the test set
- Unpopular movie trimming: In this trimming, in the test set, we keep the data(Movie) with less than or equal to 2 ratings.
- High variance movie trimming: In this trimming, in the test set, we keep the data(Movie) has rating variance at least 2 and at least 5 ratings.

With the definition above, we can evaluate the performance of k-NN filter in predicting ratings on these trimmed test set. Compute RMSE by averaging across 10 folds.

Question 12: Design a k-NN collaborative filter to predict the rating of popular movies in test set and use 10-fold cross validation to evaluate its performance. Implement k from 2 to 100 with step size 2, and compute RMSE for each k.

Result:

From figure 12 below, in popular movie trimming test set, as k increases, RMSE decrease. When k reaches 26, the RMSE becomes flattened, and the minimum average RMSE is 0.901. It shows that when k is small, the performance becomes better when k increases until k reaches 26. After k >26, the performance cannot get better obviously as k increases. In popular movie trimming test set, we can improve the performance of k-NN collaborative filter because the in popular movie trimming test set, the bias rating effect can be minimized due to the higher rating frequency. For example, one bias rating is 2.5 for a movie, but for the same movie, 3 more normal rating is 4.5. If we first use small k, we may predict rating as 3.0 and lead to high RMSE, but later as k increases, we may predict the movie rating as 4.4, and it lead to small RMSE because we get closer to 3 rating and far from 1 rating.

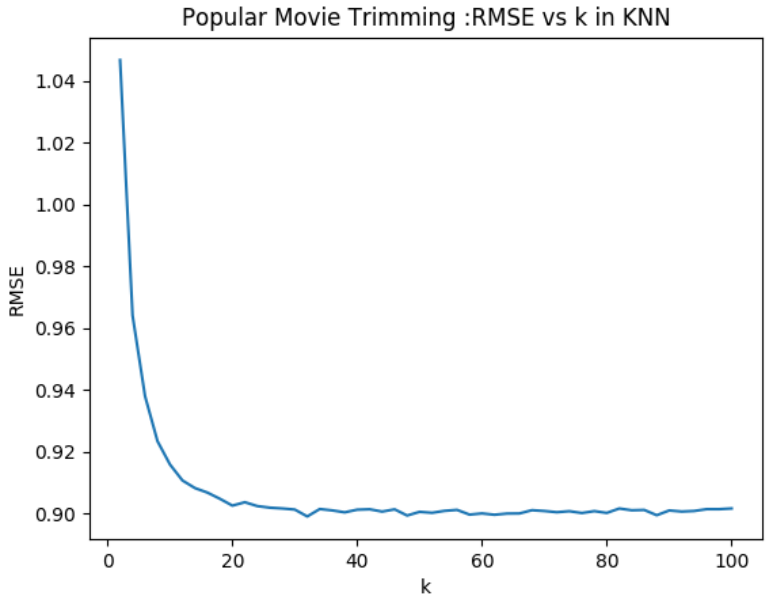


Figure 6. Popular Movie Trimming: RMSE vs k

Question 13: Design a k-NN collaborative filter to predict the rating of unpopular movie in test set and use 10-fold cross validation to evaluate its performance.

Result:

From figure 13 below, in unpopular movie trimming test set, the RMSE does not decrease as k increases, which means we cannot improve RMSE as k increases. The minimum average RMSE is 1.190. In unpopular movie trimming test set, for each movie, the rating frequency is no more than 2, which means the ground truth label set is small for a movie, so it results in large error to predict a movie, and cannot improved by increasing k.

For example, there are two ratings for an unpopular movie: they are 1.5 and 4.5. There are three cases, one is that the predict rating as 1.6 and lead to high RMSE, because it is far from rating 4.5, and the second case is that we predict rating as 4.3 and also lead to high RMSE, because it is far from rating 1.5. The last case is that we predict rating as 3.0, which is in the middle of two ground truth rating, but it still leads to high RMSE, because it is not close to both ground truth rating. Therefore, no matter what k is, the RMSE is hard to improve.

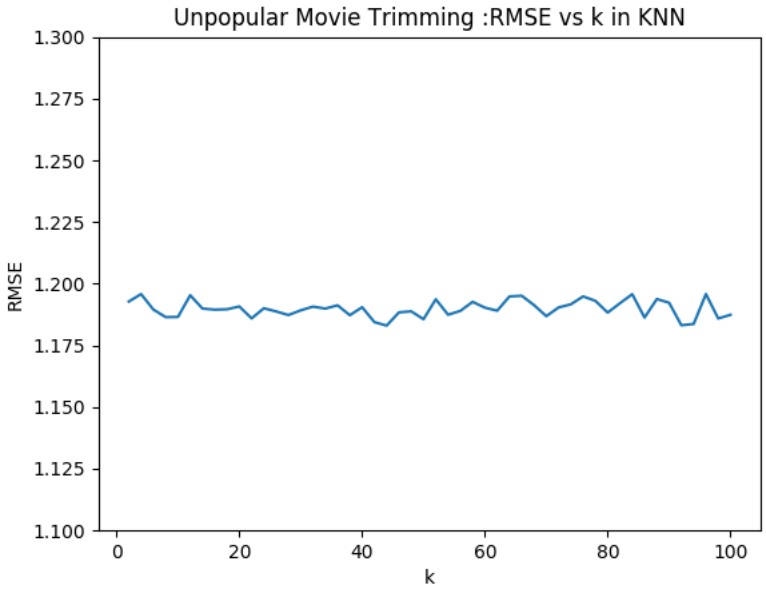


Figure 7. Unpopular Movie Trimming: RMSE vs k

Question 14: Design a k-NN collaborative filter to predict the rating of high variance movie in test set and use 10-fold cross validation to evaluate its performance. Implement k from 2 to 100 with step size 2, and compute RMSE for each k.

Result:

From figure 14 below, in high variance movie trimming test set, from k =2 to 6, RMSE decrease. For k > 6, the RMSE is between the range of 1.6 to 1.67, the value. The minimum average RMSE is 1.647 when k=6. In this result, we can indicate as k increases, RMSE improves, but it reaches its best performance fast. In the beginning, k is small, so the prediction may deviate from the ground truth, but as k becomes large, the error decreases. As k becomes larger, RMSE stays in a range which means the performance not improved because our high variance data has high variance, no matter what rating you predict, there is an error rate, and each movie in the high variance movie set has >= 5 rating, so the RMSE would vibrate.

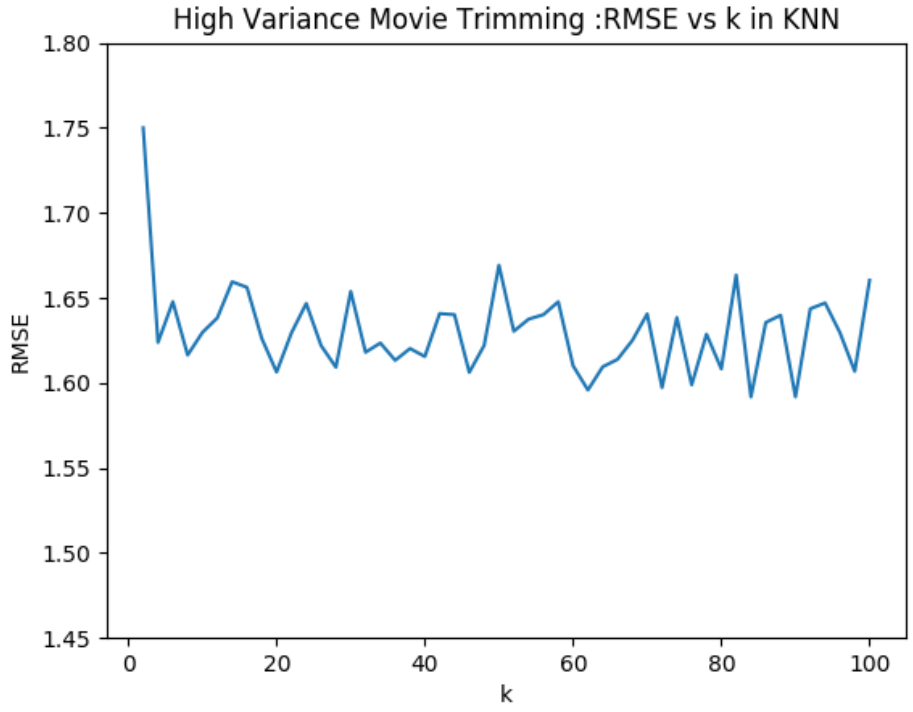


Figure 8. High Variance Movie Trimming: RMSE vs k

4.6.1 Performance evaluation using ROC curve

ROC curve is a method to visualize the performance of our binary classifier. It plots true positive rate (TPR) vs false positive rate (FPR).

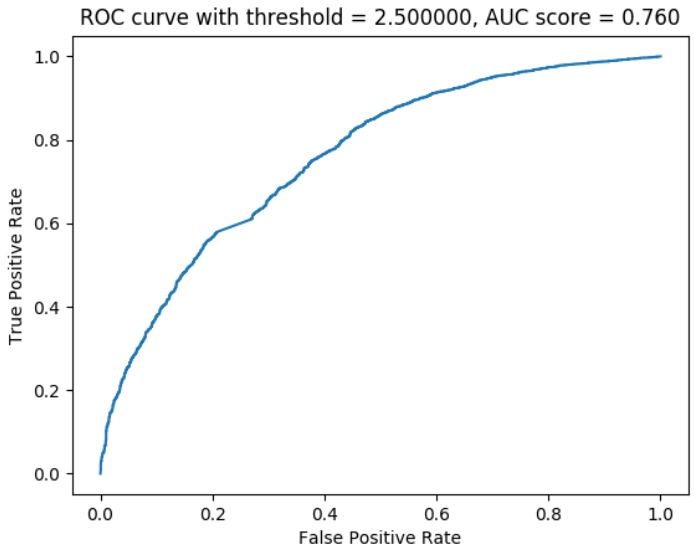
In recommendation systems, we have continuous rating (0-5), so we must make a threshold to change the continuous rating to binary rating. If the rating >= threshold, we set it to 1(imply the customer like the item), or we set it to 0. (imply the customer dislike the item)

Question 15: Plot ROC curves for the k-NN collaborative filter designed in question 10, using threshold [2.5, 3, 3.5, 4], use k found in question 11 to plot ROC, and report the AUC.

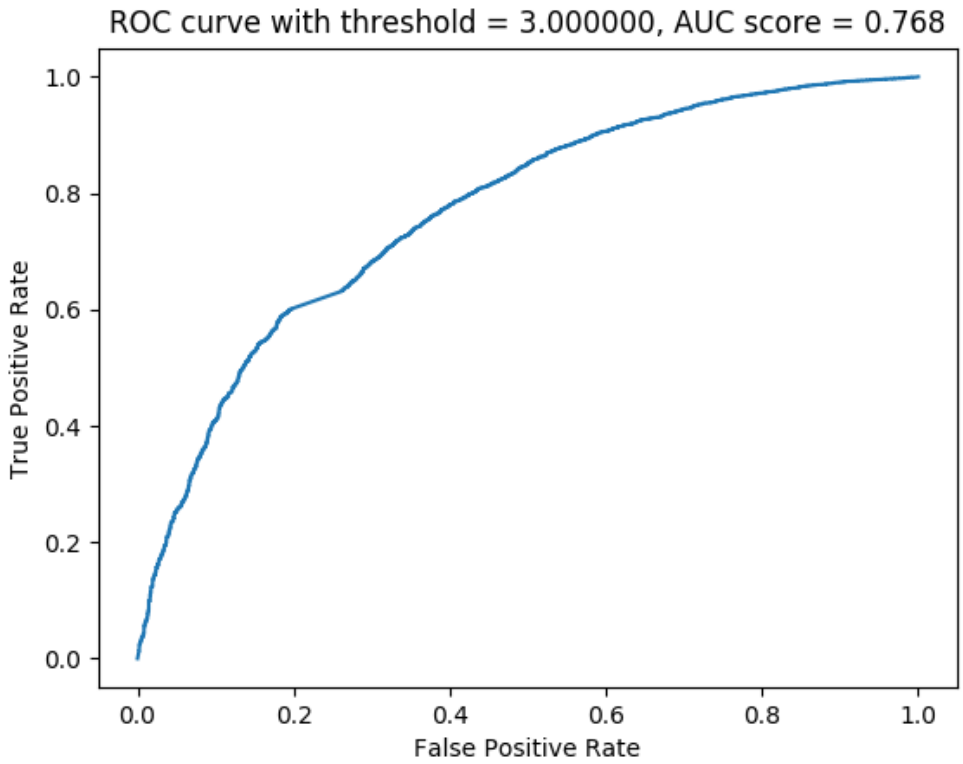
Result:

When ROC curve with threshold=3.5, the AUC value is the largest (0.773) among all the threshold, which means it gives us more true Positive rate and less False negative rate. It also indicates that in this recommendation system, a threshold = 3.5 can discriminate the users like or dislike the item, because in this threshold, the predicted value (0/1) best matches the ground truth rating of the users.

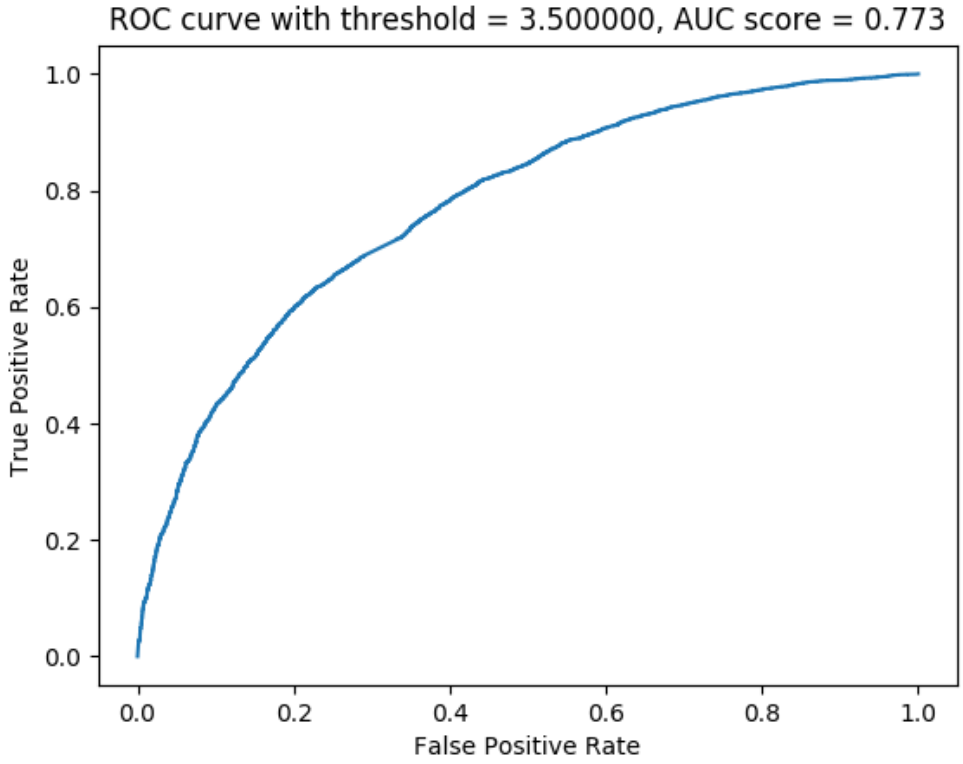
(a) Threshold = 2.5, AUC = 0.76



(b) Threshold = 3.0, AUC = 0.768



(c) Threshold = 3.5, AUC = 0.773



(d) Threshold = 4.0, AUC = 0.764

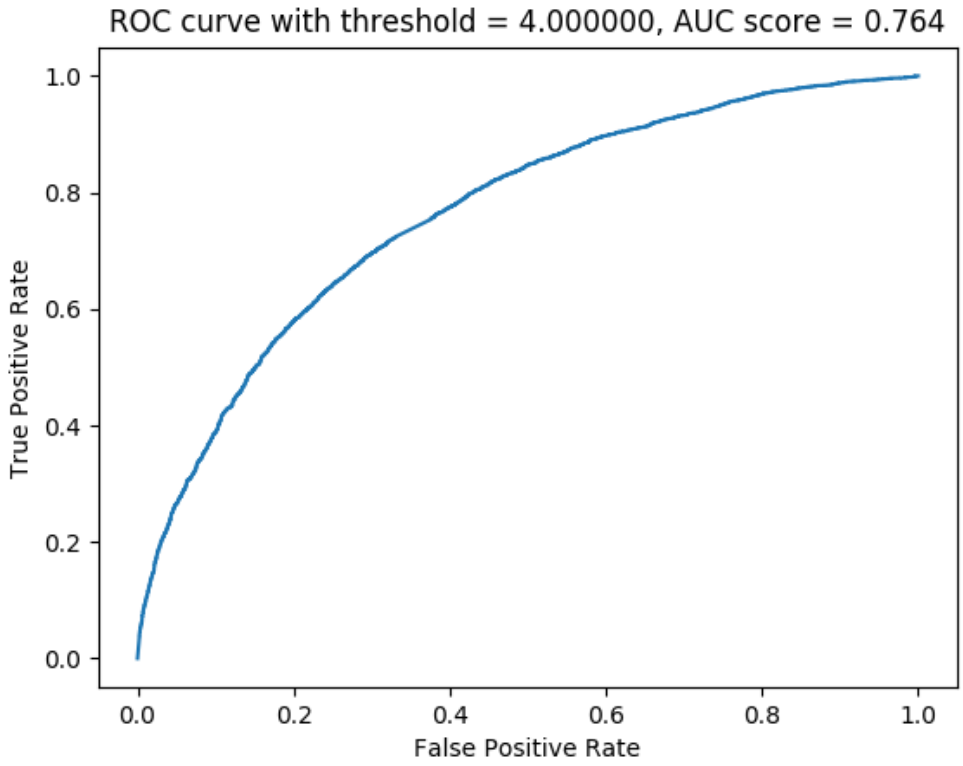


Figure 9. ROC curve with threshold

5 Model-based Collaborative Filtering

In model-based collaborative filtering, we aim to use models to predict the unrated movies for different users. As in this project, we select the latent factor based models for collaborative filtering, and use two variations: non-negative matrix factorization (NNMF) and matrix factorization with bias (MF with bias).

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 latent factor based models are equivalent to solving the optimization problem given by:

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

The optimization problem for this equation, however, is not convex. This is because the Hessian matrix for this equation is not positive definite. On the other hand, given that U is fixed, the procedure we are solving for V is to solve a least-square problem. Similarly, given V fixed, the procedure to solve U is also a least-square problem. To prove this, we choose a fixed U and formulate the least squares problem as below.

Let y=UVT, where U is the parameter and V is the independent variable. The ground truth label of y_{true} is r_{ij} and the prediction label y_{pred} is UVT_{ij}. It becomes a least squares problem with loss function:

$$\underset{U,V}{\text{minimize}} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(y_{true} - y_{pred})^2$$

5.1 Non-negative matrix factorization (NNMF)

In this project, we will use stochastic gradient descent (SGD) optimization algorithm for non-negative matrix factorization. Here, we just use the below python package to do the work:
`surprise.prediction_algorithms.matrix_facotrization.NMF`.

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).

In this problem, we need to use the 10-fold cross-validation to evaluate the performance of NNMF-based collaborative filter for the prediction of ratings. We select the number of latent factors k to be 2, 4, 6, …, 50. Then for each k, we need to calculate the values for average RMSE and average MAE across the 10 folds. To show the results, we plot the average values against k. The plots for both average values are shown below.

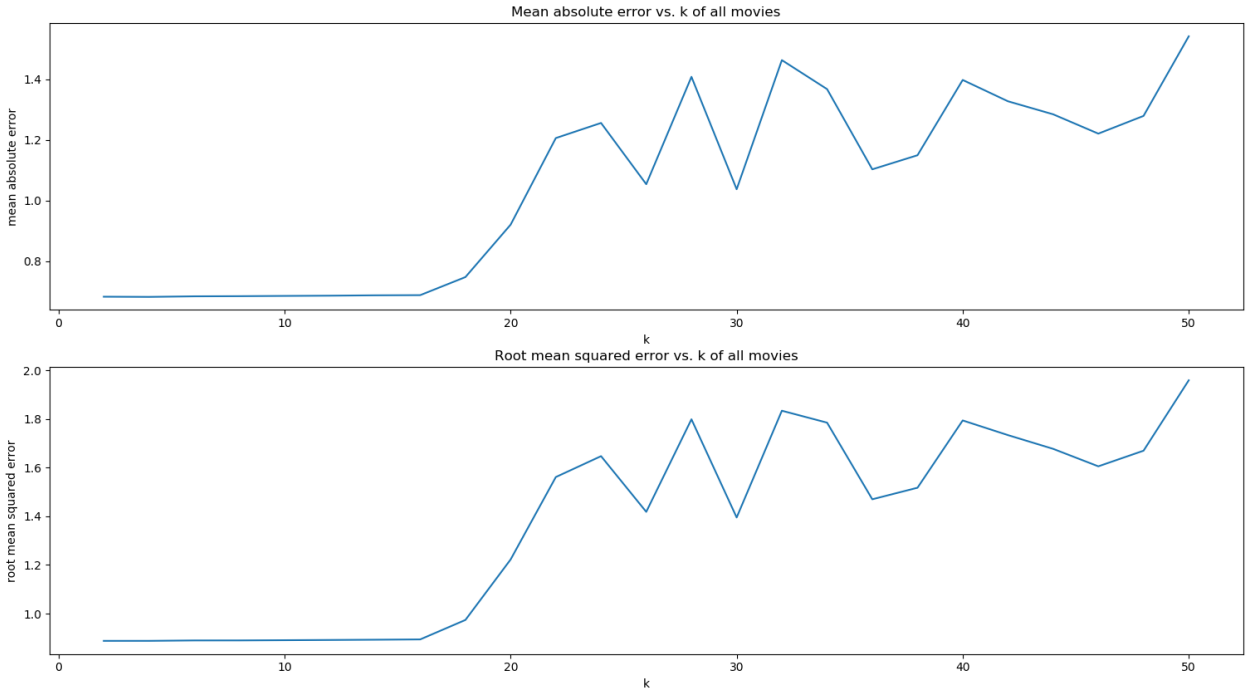


Figure 10. Average MAE and Average RMSE v.s. k

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?

According to the figures in Question 17, we can see that both errors are quite small when k is less than 18. When k becomes even larger, the errors become quite large. To make sure the best value for latent factor k regarding the errors, we record the errors for small k's in the following table.

Table 1. Average MAE and RMSE with respect to small k's

k	Average MAE	Average RMSE
2	0.682	0.888
4	0.681	0.888
6	0.683	0.890
8	0.684	0.890
10	0.685	0.891
12	0.686	0.892
14	0.687	0.893
16	0.688	0.894
18	0.747	0.974

From the table, we can see that average MAE reaches the minimum value when k = 4, and average RMSE reaches the minimum value when k = 2 or 4. Therefore, the optimal number of latent factor is 4, the minimum average MAE is 0.681, and the minimum average RMSE is 0.888.

According to the CSV file, the number of movie genres is 20. Therefore, the optimal number of latent factors here is not the same as 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 its performance using 10-fold cross validation.

The procedure in this problem is approximately the same as those in question 17 and 18. The only difference is to use the popular movie trimmed test set. The plot of both average RMSE and average MAE is shown in the below graph.

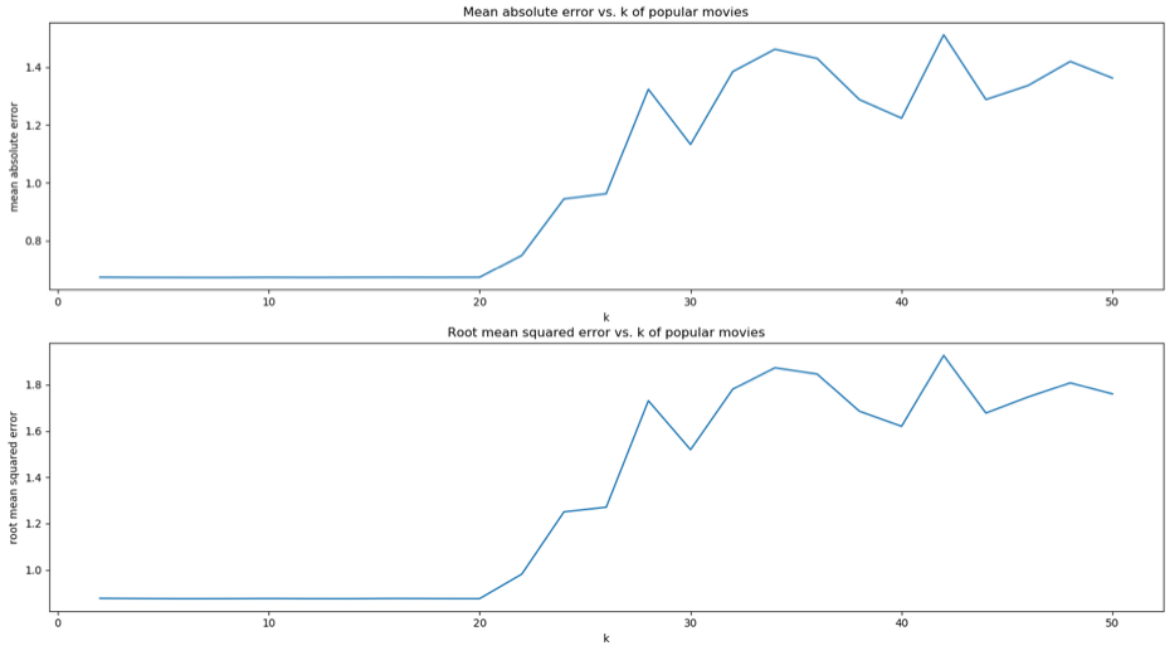


Figure 11. Average MAE and Average RMSE v.s. k for popular movie trimmed data set

From the output, we find the minimum average RMSE for popular movie trimmed test set is 0.8753, which occurs when k = 6.

Question 20: Design a NNMF 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.

For this question, we will use the unpopular movie trimmed test set instead. The plot of both average RMSE and average MAE is shown in the below graph.

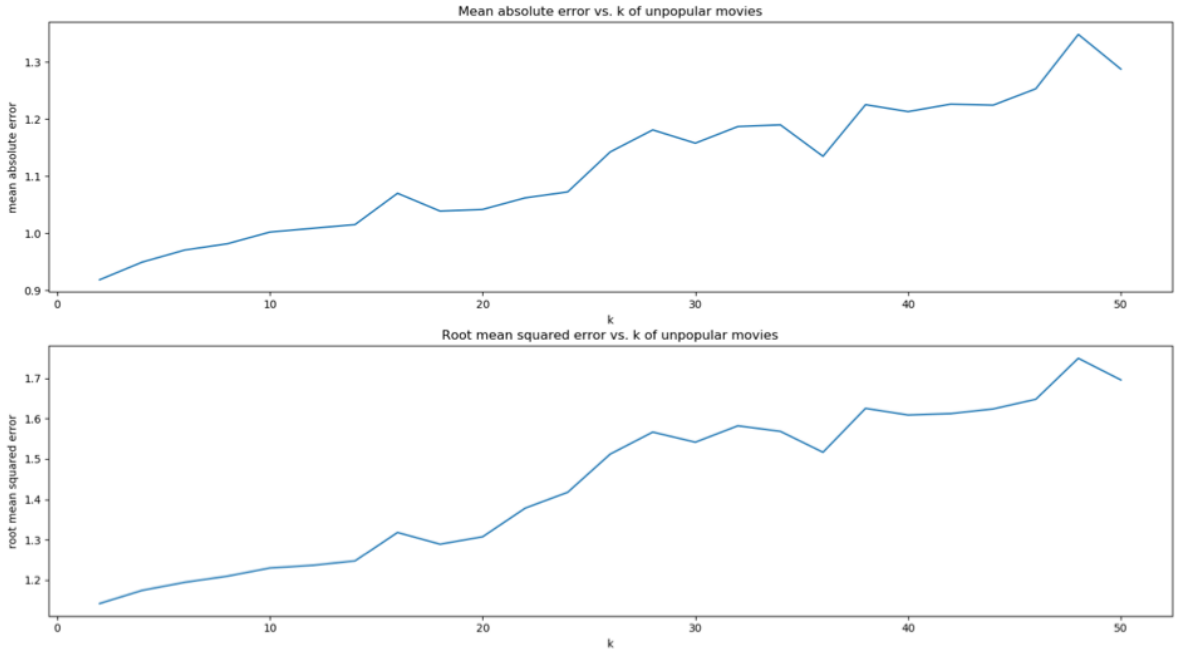


Figure 12. Average MAE and Average RMSE v.s. k for unpopular movie trimmed data set

From the output, we can find the minimum average RMSE for the unpopular movie trimmed test set is 1.142, which occurs when k = 2.

Question 21: Design a NNMF 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.

For this problem, we will use the high variance movie trimmed test set. The plot of both average RMSE and average MAE is shown in the below graph.

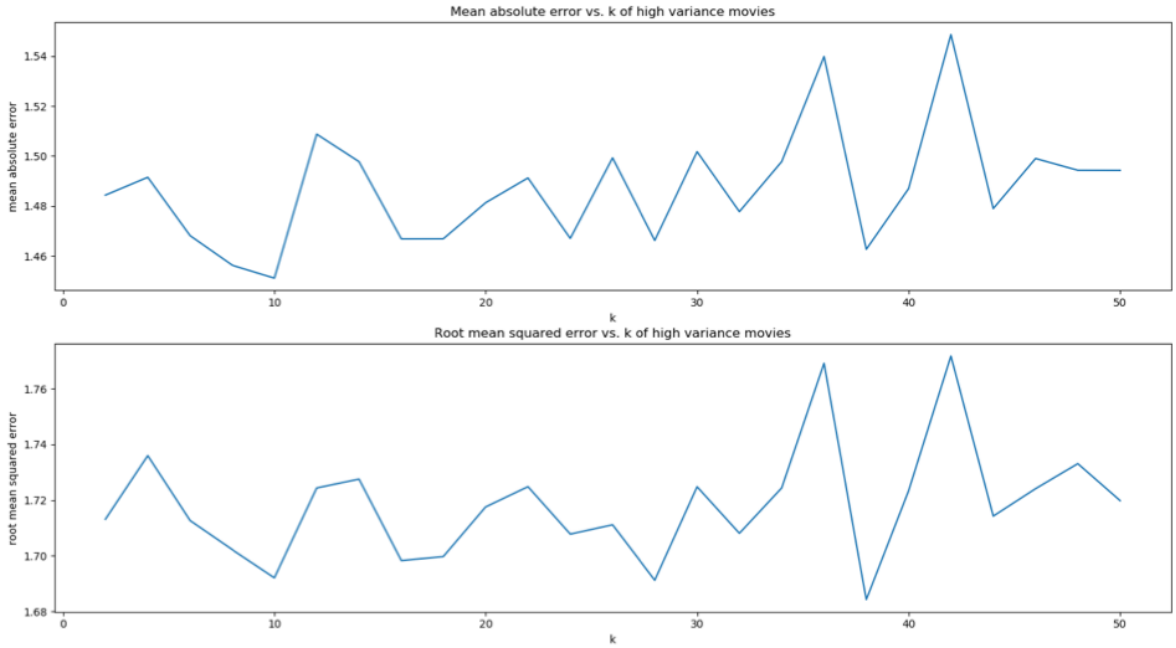


Figure 13. Average MAE and Average RMSE v.s. k for high variance movie trimmed data set

From the output, we find the minimum average RMSE for high variance movie trimmed test set is 1.692, which occurs when k = 6.

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].

In this question, we firstly plot the ROC curves for the collaborative filter in question 17 with threshold values 2.5, 3, 3.5 and 4, using the latent factor k = 4. The area under the curve (AUC) for each ROC plot is shown in the bottom-right of the graph. We also record each value in the table below.

Table 2. AUC for different threshold with k = 4

Threshold	AUC
2.5	0.75
3	0.76
3.5	0.77
4	0.76

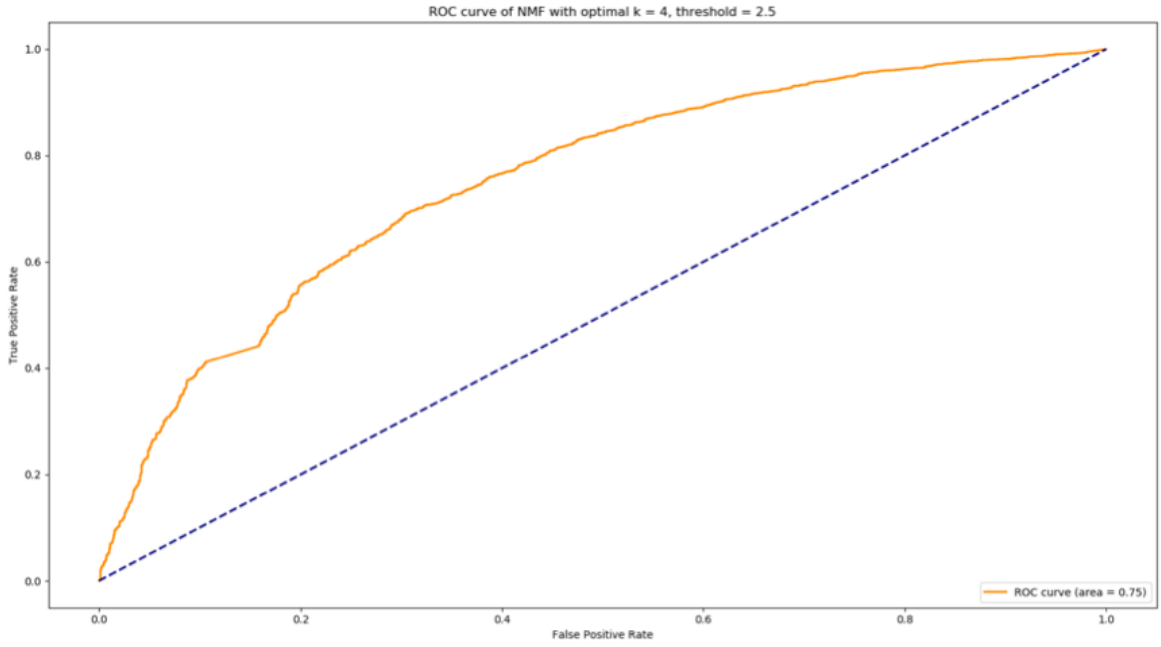


Figure 14. ROC curve with threshold value 2.5

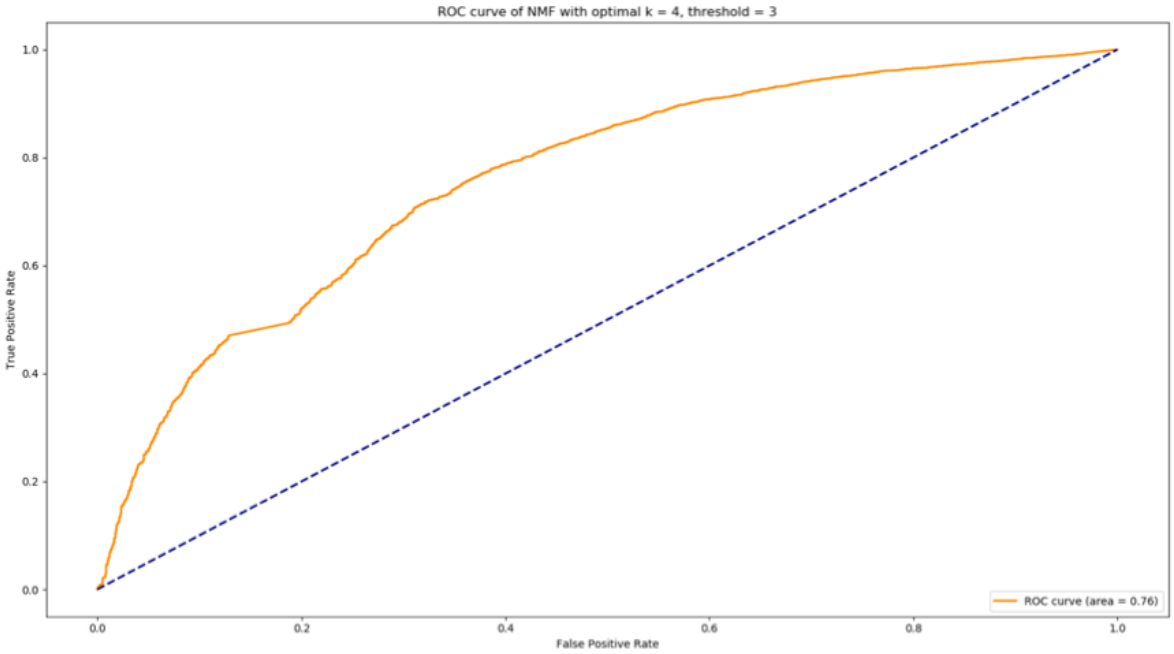


Figure 15. ROC curve with threshold value 3

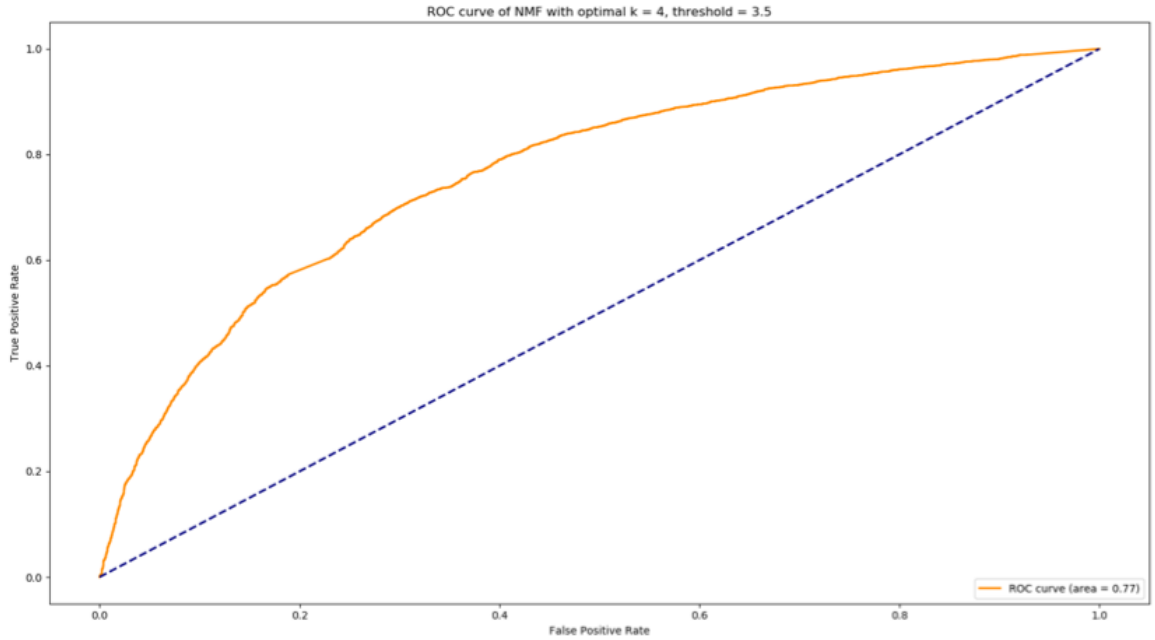


Figure 16. ROC curve with threshold value 3.5

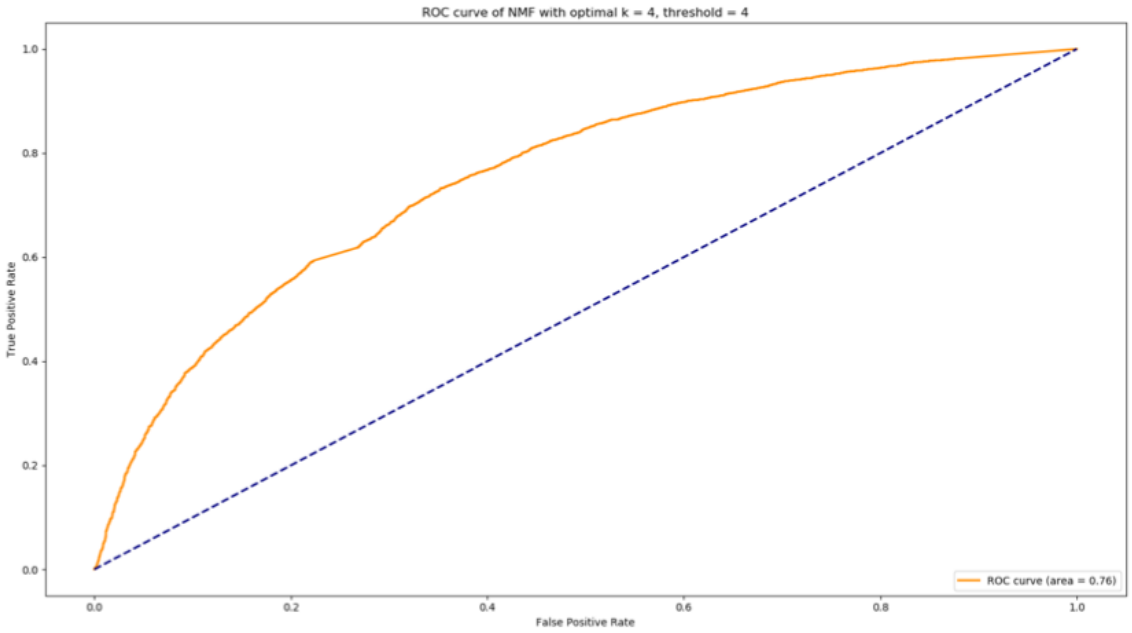


Figure 17. ROC curve with threshold value 4

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). Do the top 10 movies belong to a or a small collection of genre? Is there a connection between the latent factors and the movie genres?

In Question 18, we noticed that the best value for latent factor k is not the same as the number of movie genres. Therefore, until now, we have no idea about the interpretability of NNMF. To explore the relationship between latent factors and the number of movie genres, we did the following operations. Firstly, we choose the latent factors to be the same as the number of movie genres, which is k = 20. Then, within each column of V, we chose the top 10 movies and see if they belong to a certain genre, or a combination of several genres.

Since the latent factor k is equal to 20, the number of columns in V should be 20. Here, I will only illustrate the first two columns in order to give an intuition about the connection between latent factors and movie genres. Below are the output results.

Col 1:

```
Documentary, Drama, Action|Drama|Thriller, Drama, Action|Comedy,
Drama|Thriller, Drama|Fantasy|Sci-Fi, Comedy|Drama, Comedy|Romance,
Action|Adventure|Fantasy
```

Col 2:

```
Drama|Mystery|Sci-Fi, Action|Comedy, Comedy, Action|War, Comedy, Drama,
Comedy|Crime|Mystery|Thriller, Children|Comedy, Drama, Comedy
```

As we can see from Col 1, the movie genres of the top 10 movies have something in common. 6 out of 10 movie genres are marked with “Drama”. So, for this column, the latent factor largely indicates the movie set in the column has much to do with the drama movie genre. Similarly, in Col 2, 6 out of 10 movie genres have the label “Comedy”. Therefore, the movie set for this latent factor aims at picking out the comedy movies in the column. Actually, this phenomenon happens for every column in the 20 columns. Therefore, I don’t expand the results for the other 18 columns here.

So we may reasonably guess that, for each column corresponding to one latent factor, the movie genres have a high similarity. On the other hand, for each movie, the movie label is marked with a combination of several genres, so we cannot guarantee that 20 movie genres indicate the best value for latent factor k. In fact, the real choices for k have already been given in the previous questions based on the experimental results.

5.2 Matrix Factorization with Bias (MF with bias)

There is another variation to the unconstrained matrix factorization formula, which is the matrix factorization with bias. The difference from the previous part is that we add a bias term for each user and each item in the cost function for the optimization, as well as the prediction function. The procedure through this algorithm is the same as that for non-negative matrix factorization. Therefore, I will just repeat the above procedures here.

Question 24: Design a MF with bias collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross-validation.

With MF with bias method, we use the 10-fold cross-validation to evaluate the performance for the prediction of ratings. We select the number of latent factors k to be 2, 4, 6, …, 50. Then for each k, we need to calculate the values for average RMSE and average MAE across the 10 folds. In order to show the results, we plot the average values against k. The plots for both average values are shown below.

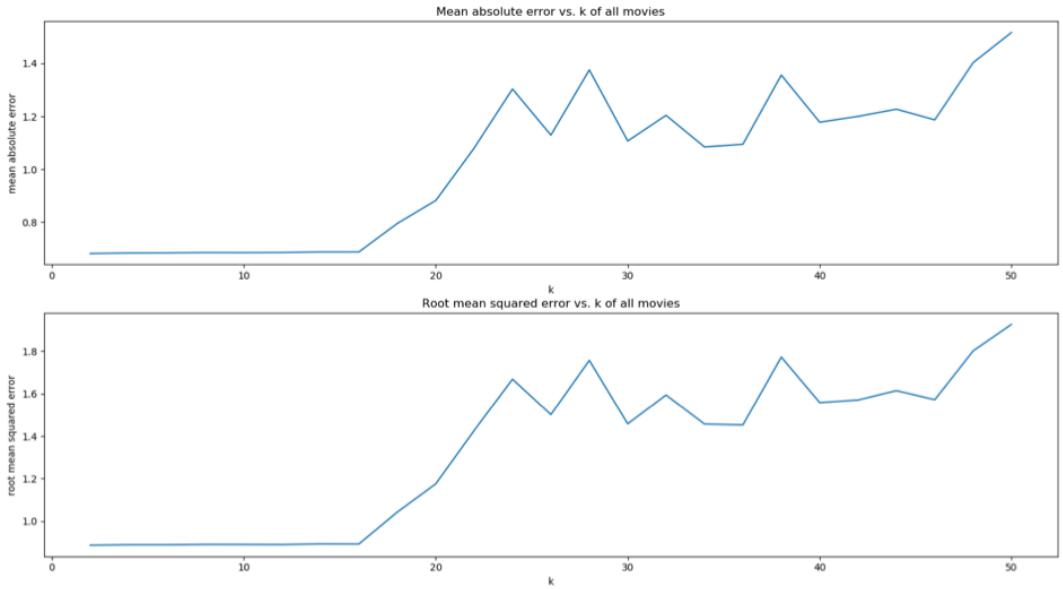


Figure 18. Average MAE and Average RMSE v.s. k

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.

According to the figures in Question 24, we can see that both errors are quite small when k is less than 18. When k becomes even larger, the errors become quite large. In order to make sure the best value for latent factor k regarding the errors, we record the errors for small k's in the following table.

Table 3. Average MAE and RMSE with respect to small k's

k	Average MAE	Average RMSE
2	0.681	0.887
4	0.683	0.889
6	0.684	0.889
8	0.685	0.891
10	0.685	0.891
12	0.685	0.890
14	0.687	0.893
16	0.687	0.893
18	0.795	1.043

From the table, we can see that the optimal number of latent factor is 2, the minimum average MAE is 0.681, and the minimum average RMSE is 0.887.

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.

The procedure in this problem is approximately the same as those in question 24 and 25. However, we will use the popular movie trimmed test set. The plot of both average RMSE and average MAE is shown in the below graph.

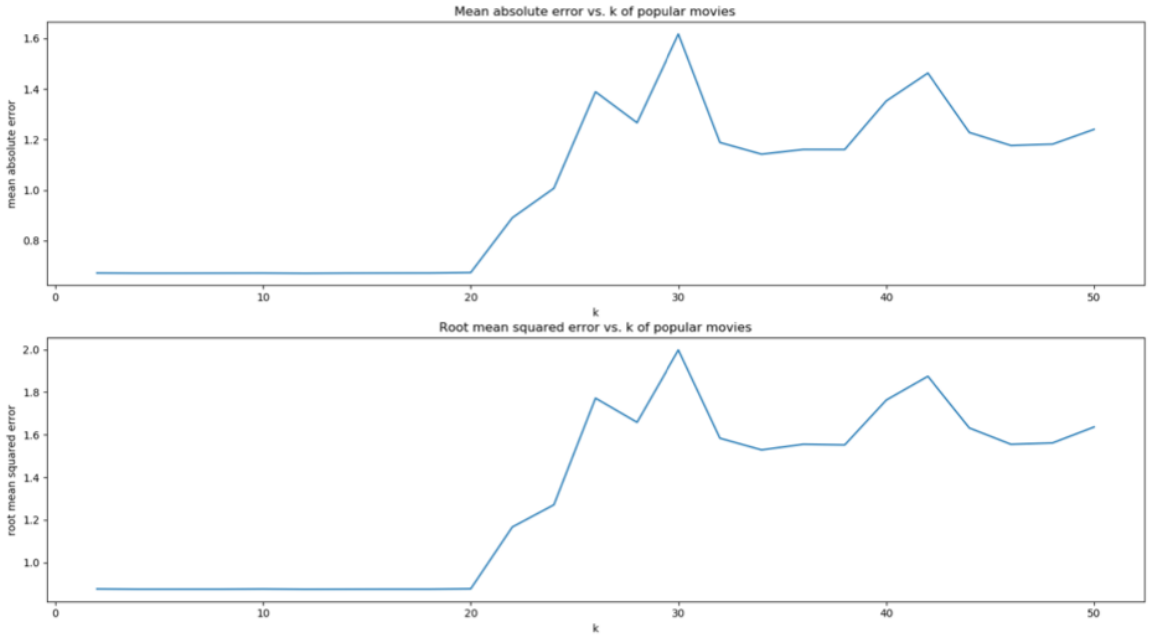


Figure 19. Average MAE and Average RMSE v.s. k for popular movie trimmed data set

From the output, we find the minimum average RMSE for popular movie trimmed test set is 0.8744, which occurs when k = 12.

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.

For this question, we will use the unpopular movie trimmed test set instead. The plot of both average RMSE and average MAE is shown in the below graph.

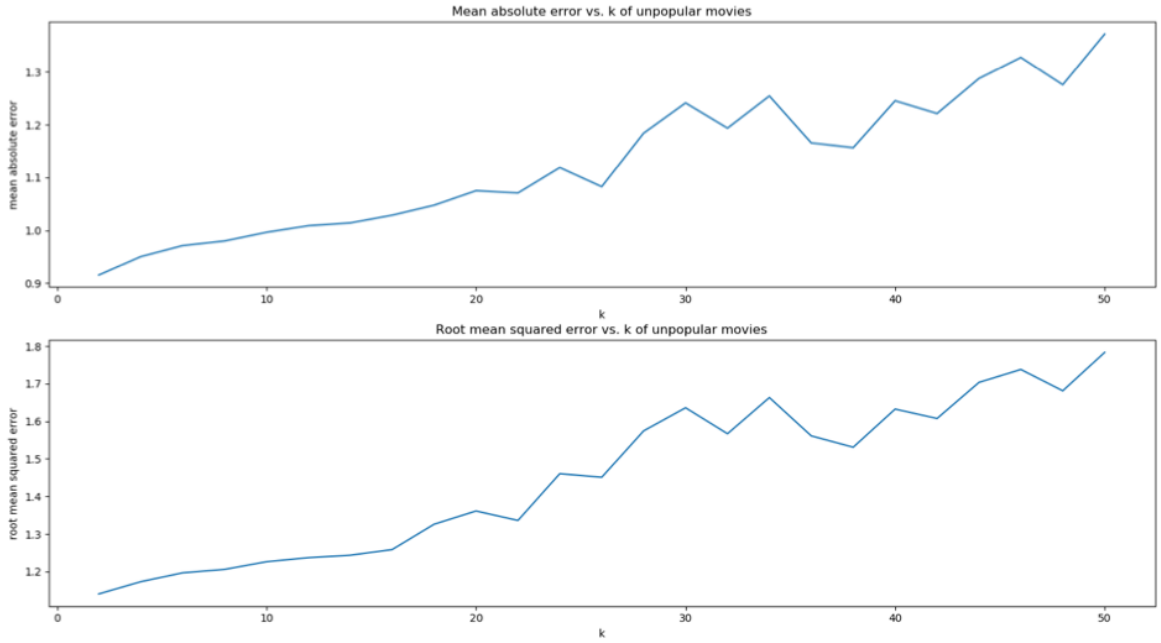


Figure 20. Average MAE and Average RMSE v.s. k for unpopular movie trimmed data set

From the output, we can find the minimum average RMSE for the unpopular movie trimmed test set is 1.140, which occurs when k = 2.

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.

For this problem, we will use the high variance movie trimmed test set. The plot of both average RMSE and average MAE is shown in the below graph.

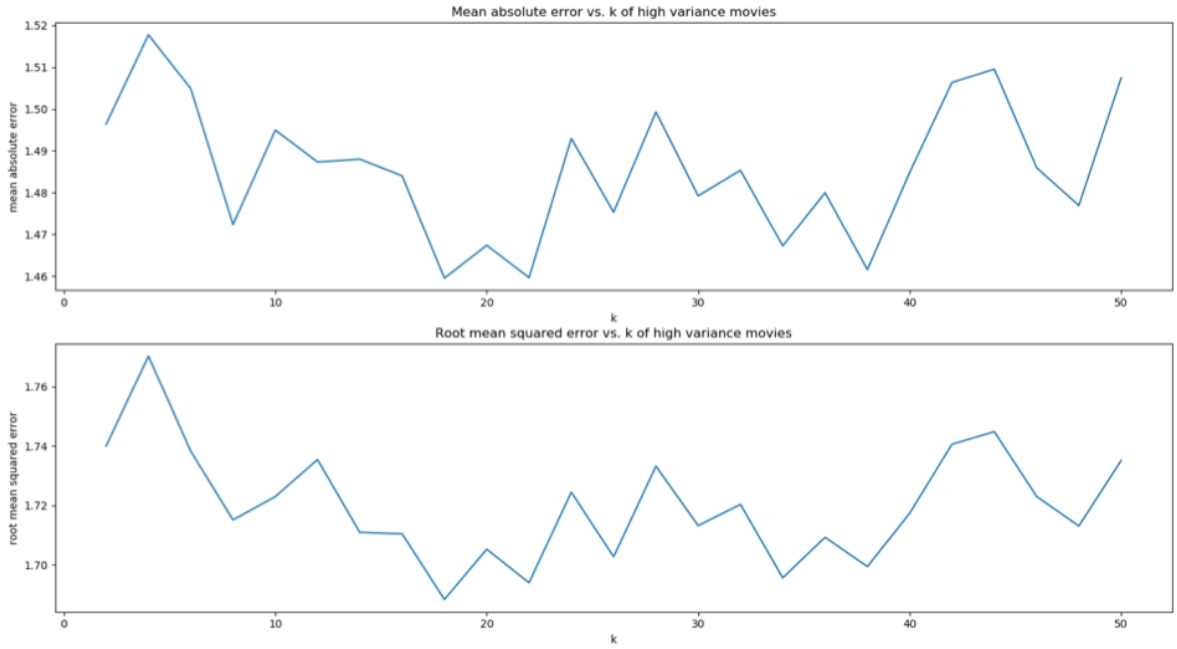


Figure 21. Average MAE and Average RMSE v.s. k for high variance movie trimmed data set

From the output, we find the minimum average RMSE for high variance movie trimmed test set is 1.688, which occurs when k = 18.

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].

In this question, we firstly plot the ROC curves for the collaborative filter in question 24 with threshold values 2.5, 3, 3.5 and 4, using the latent factor k = 2. The area under the curve (AUC) for each ROC plot is shown in the bottom-right of the graph. We also record each value in the table below.

Table 4. AUC for different threshold with k = 2

Threshold	AUC
2.5	0.80
3	0.80
3.5	0.78
4	0.78

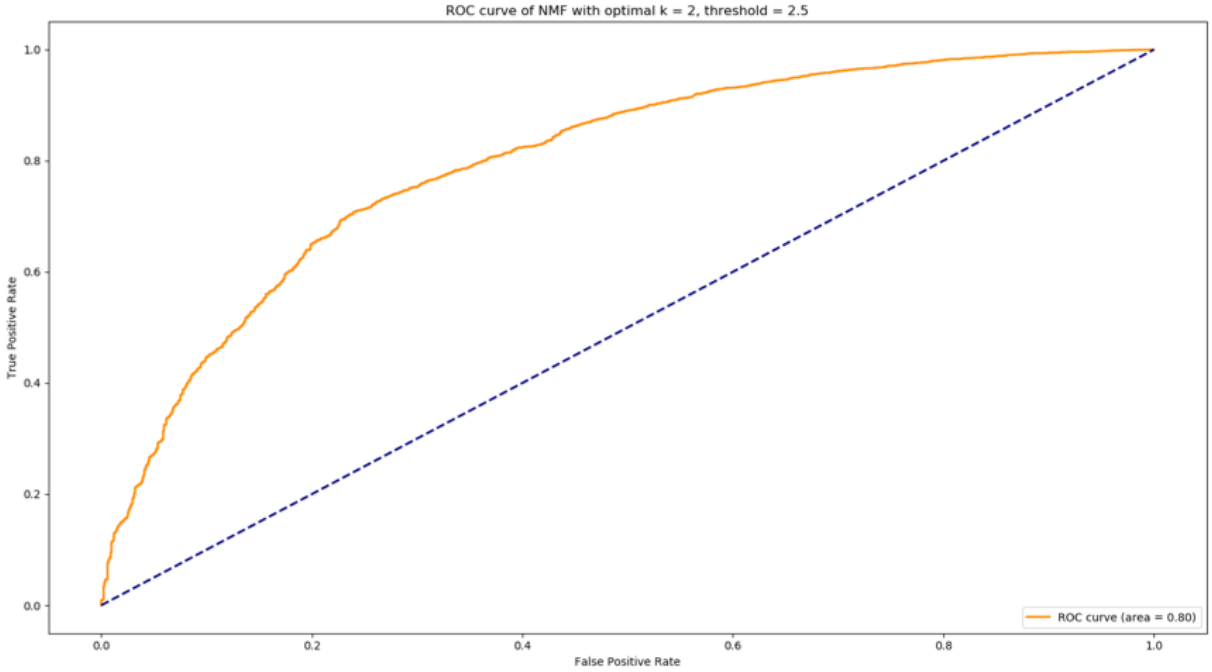


Figure 22. ROC curve with threshold value 2.5

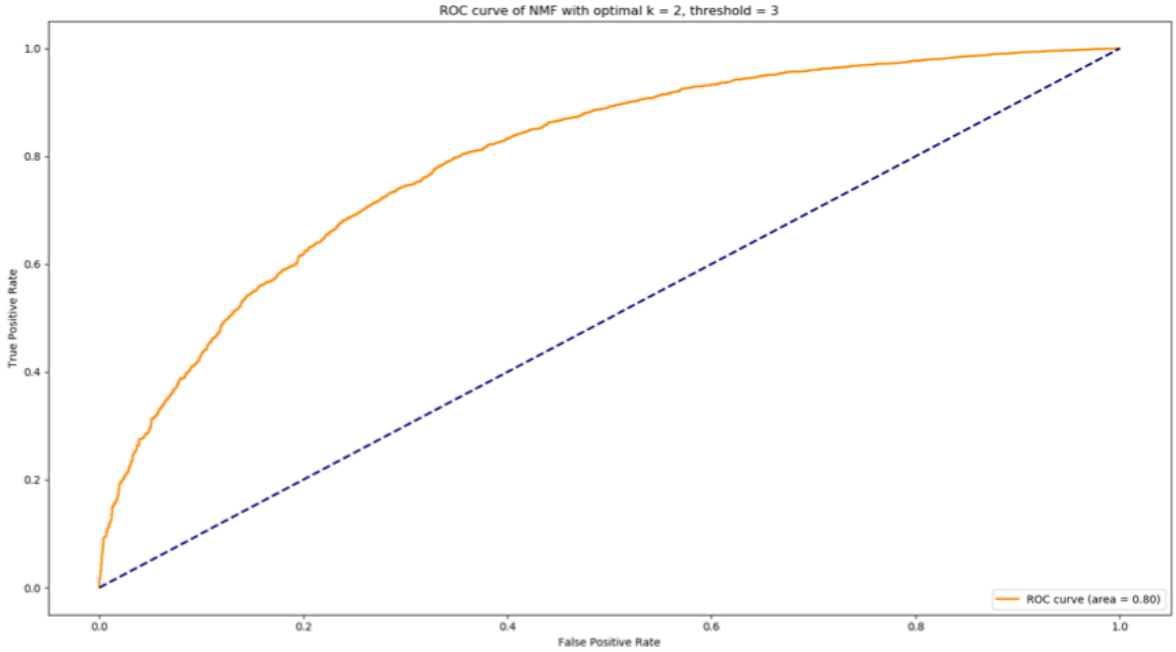


Figure 23. ROC curve with threshold value 3

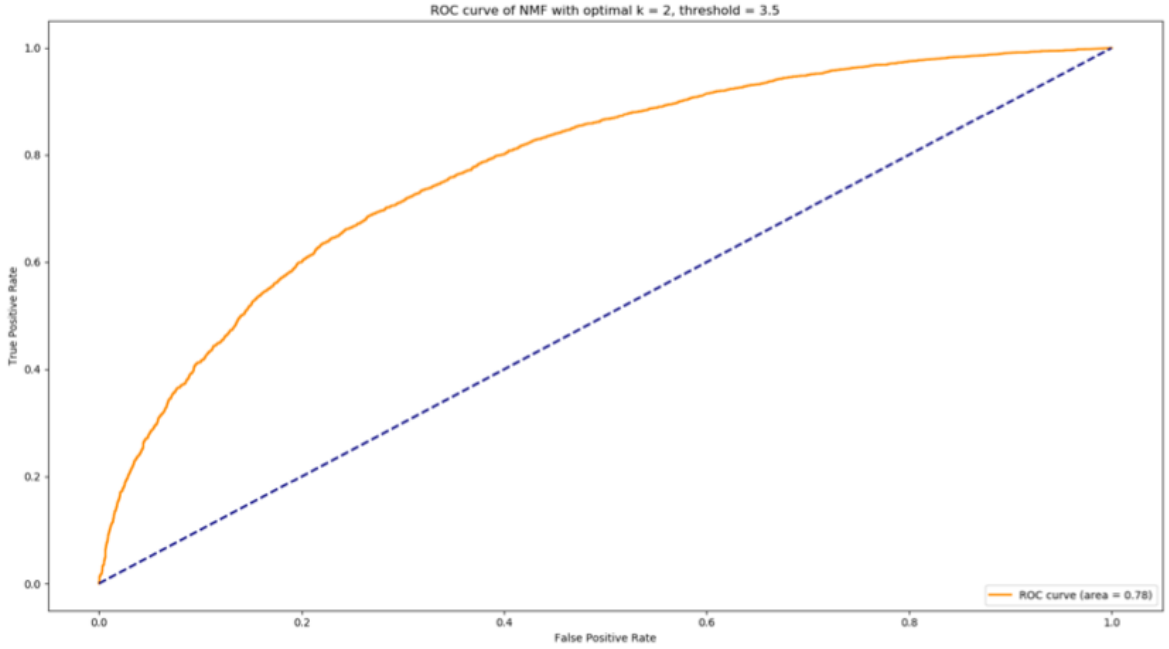


Figure 24. ROC curve with threshold value 3.5

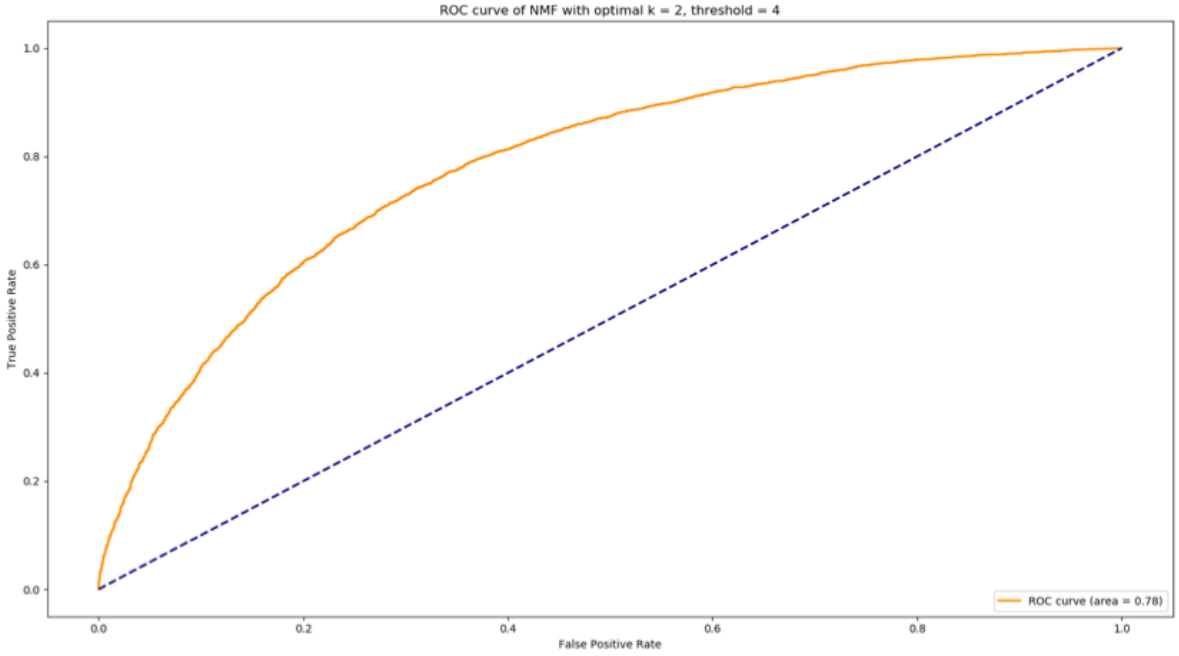


Figure 25. ROC curve with threshold value 4

6 Naïve collaborative filtering

In this part, we use Naïve collaborative filtering to predict the ratings of the movies. The prediction function is as below:

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

where $\hat{r}_{ij}$ denotes the predicted rating of user i for item j, and μ_i is the mean rating of user i. In other words, for this naïve collaborative filtering method, the specific rating from a user is independent on other users, instead, the rating would be the mean rating of the whole rated movies from this particular user.

Question 30 Design a naive collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

In this question, we need to implement a class named NaïveCollaborativeFilter with two methods fit and estimate. In fit method, we calculate the mean rating of each user. And in estimate method, we use each mean rating of each user to predict specific movie rating. If there is no rating data for a particular user, then we will set the predicted ratings as 0 and later we will use "0" to identify and delete the impossible estimations. Last, we use 10-fold cross validation to calculate each fold's RMSE and the average RMSE of the 10 folds. Below are our results:

Table 5. RMSE with different testing folds (MovieLens dataset)

Testing fold	RMSE
1st fold	1.155
2nd fold	1.157
3rd fold	1.169
4th fold	1.182
5th fold	1.125
6th fold	1.226
7th fold	1.173
8th fold	1.165
9th fold	1.191
10th fold	1.137
Average across all 10 folds	1.168

From the table above, we can see that the RMSE of each fold are in the range of 1.15 to 1.23, and the average RMSE across all 10 folds is 1.168.

Question 31 Design a naive 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. Report the average RMSE.

In this part, we use the naïve collaborative filter in question 30 to predict the ratings of the movies in the popular movie trimmed set which received more than 2 ratings. Below are our results:

Table 6. RMSE with different testing folds (popular movie dataset)

Testing fold	RMSE
1st fold	1.106
2nd fold	1.134
3rd fold	1.115
4th fold	1.118
5th fold	1.106
6th fold	1.131
7th fold	1.128
8th fold	1.138
9th fold	1.139
10th fold	1.116
Average across all 10 folds	1.123

From the table above, we can see that the RMSE of each fold are in the range of 1.10 to 1.14, and the average RMSE across all 10 folds is 1.123.

Question 32 Design a naive 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. Report the average RMSE.

In this part, we use the naïve collaborative filter in question 30 to predict the ratings of the movies in the unpopular movie trimmed set which received at most 2 ratings. Below are our results:

Table 7. RMSE with different testing folds (unpopular movie dataset)

Testing fold	RMSE
1st fold	1.242
2nd fold	1.255
3rd fold	1.220
4th fold	1.202
5th fold	1.223
6th fold	1.269
7th fold	1.234
8th fold	1.254
9th fold	1.308
10th fold	1.240
Average across all 10 folds	1.245

From the table above, we can see that the RMSE of each fold are in the range of 1.20 to 1.31, and the average RMSE across all 10 folds is 1.245.

Question 33 Design a naive 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. Report the average RMSE.

In this part, we use the naïve collaborative filter in question 30 to predict the ratings of the movies in the high variance movie trimmed set which received at least 5 ratings and the variance is equal or more than 2. Below are our results:

Table 8. RMSE with different testing folds (high variance movie dataset)

Testing fold	RMSE
1st fold	1.736
2nd fold	1.730
3rd fold	1.855
4th fold	1.647
5th fold	1.731
6th fold	1.726
7th fold	1.689
8th fold	1.609
9th fold	1.740
10th fold	1.685
Average across all 10 folds	1.715

From the table above, we can see that the RMSE of each fold are in the range of 1.60 to 1.75, and the average RMSE across all 10 folds is 1.715. Compared to 3 previous datasets, the high variance movie dataset has average higher RMSE than the other 3 datasets.

7 Performance comparison

In this part, we compare the performance of k-NN (with k = 24), NNMF (with k = 4) and MF with bias (with k = 2) based collaborative filters by plotting their ROC curves into one figure.

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.

Below are our results:

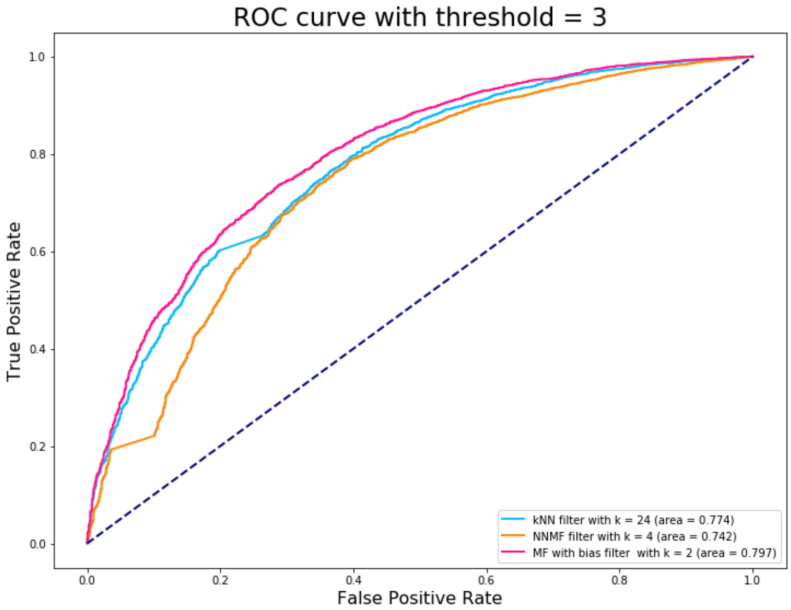


Figure 26. ROC curve with threshold = 3

From the figure, we can find that with the best k setting and the threshold = 3, MF with bias based collaborative filter has the largest AUC area about 0.797, and NNMF collaborative filter has the lowest AUC area about 0.742. This result indicates that MF with bias based collaborative filtering achieves higher TPR and lower FPR, and its predictions of ratings best match the ground truth of the ratings users provided.

8 Ranking

In this part, we do some ranking operations based on the prediction results in the previous sections. The ranking and recommendation pipeline is as below:

1) For each user, compute the predicted ratings for all movies with the previous filtering techniques and store them in a list L.
2) Sort the list L in descending order.
3) Recommend the first top `t` movies to specific user.

Question 35: Please explain the meaning of precision and recall in your own words.

Precision: It is the percentage that relevant and recommended (ground-truth positives) items take up in recommended items.

Recall: It is the percentage that relevant and recommended (ground-truth positives) items occupy in relevant items. Relevant items are those items a user really likes.

Question 36: Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using k-NN collaborative filter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis).

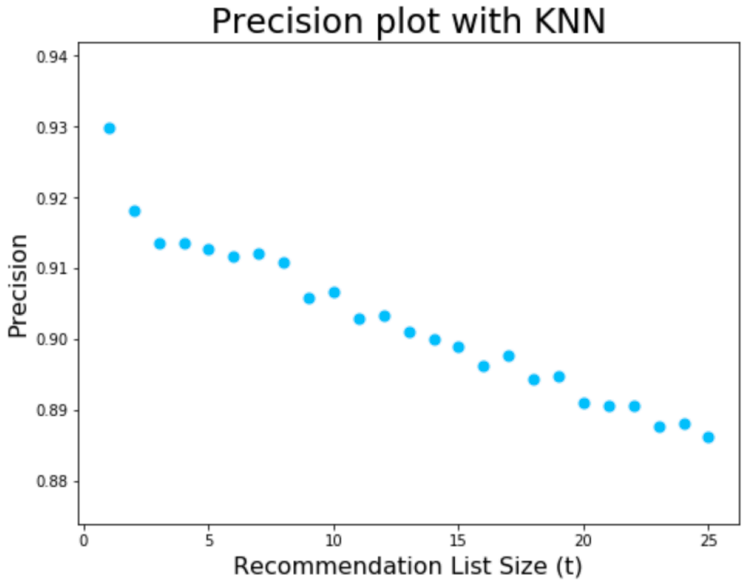


Figure 27. Precision plot with KNN filter

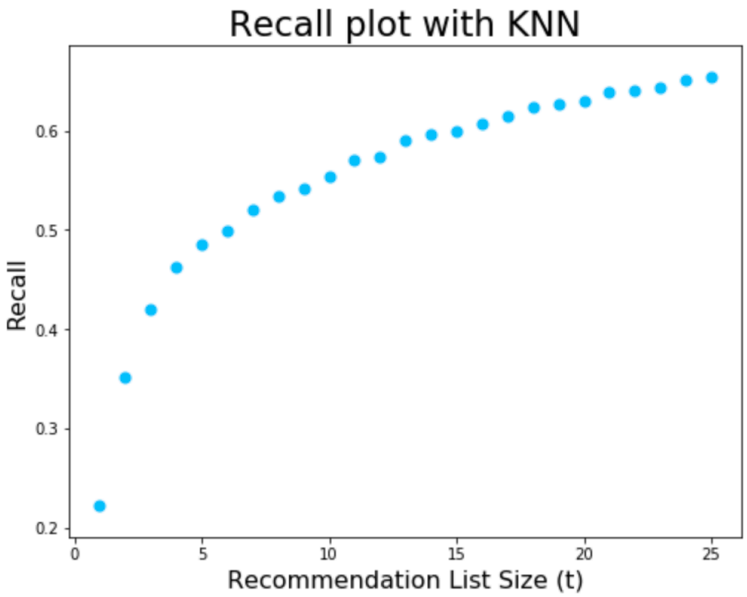


Figure 28. Recall plot with KNN filter

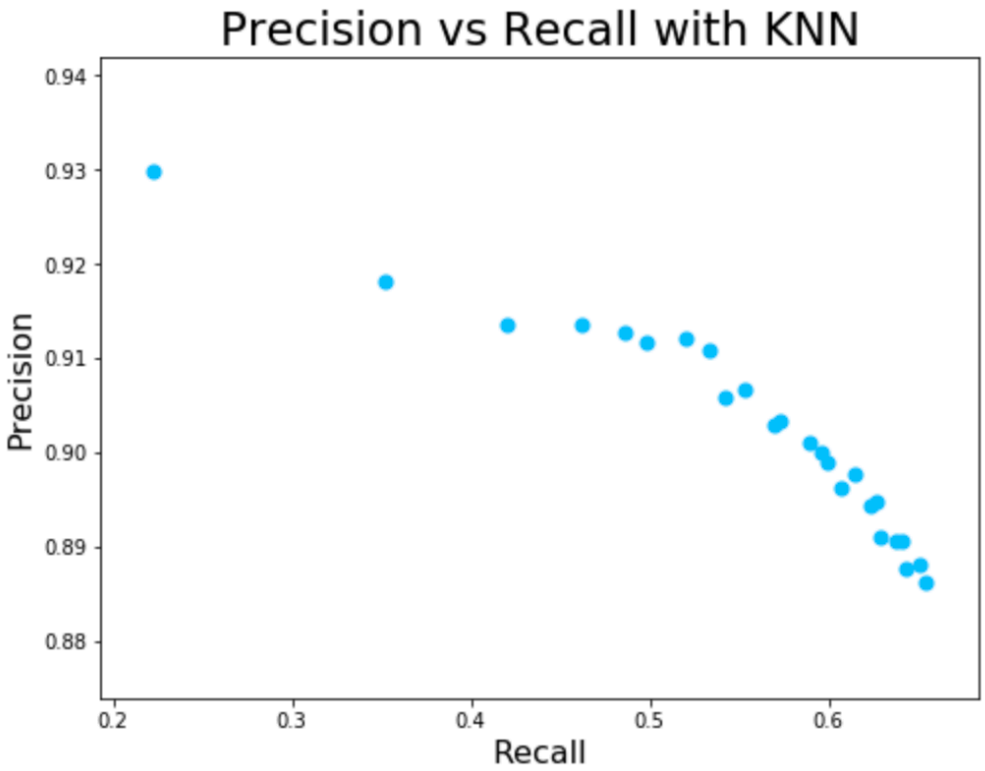


Figure 29. Precision-Recall curve

From the figures above, we can see that the precision plot is not monotonic in t, since in equation of precision, both its numerator and denominator are functions of t and may change with t differently. While recall plot is monotonic, since only its numerator is a function of t. And the precision-recall curve is not monotonic either because of the non-monotonic precision plot.

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.

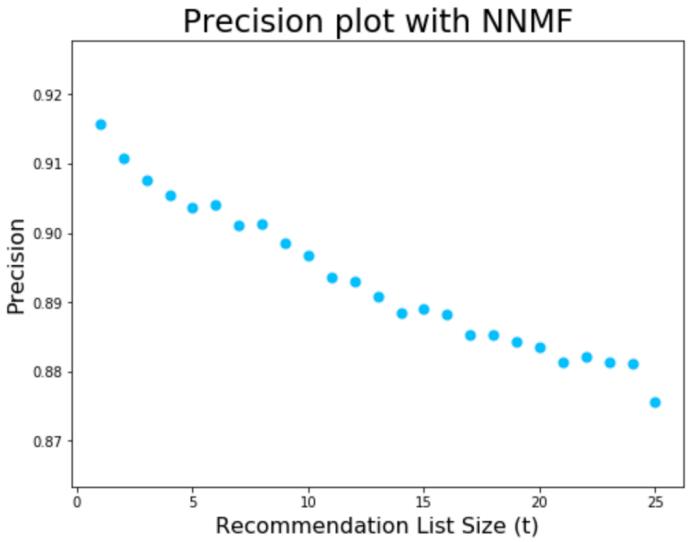


Figure 30. Precision plot with NNMF

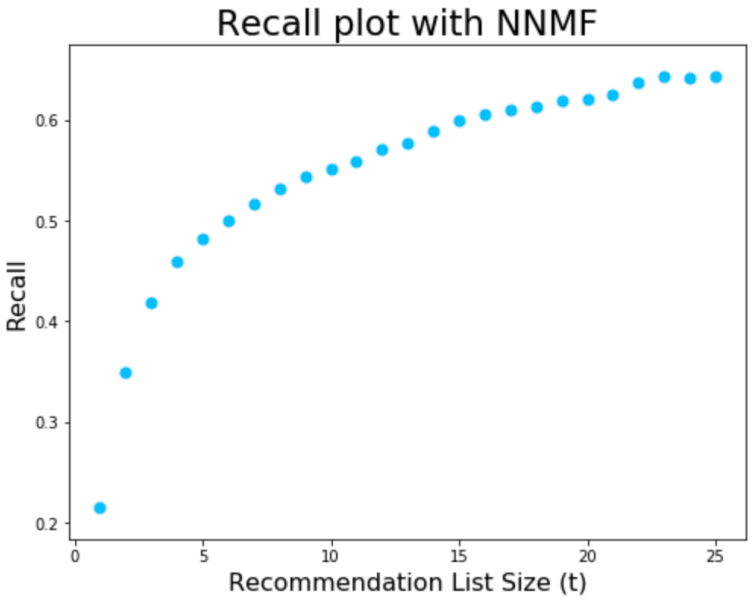


Figure 31. Recall plot with NNMF

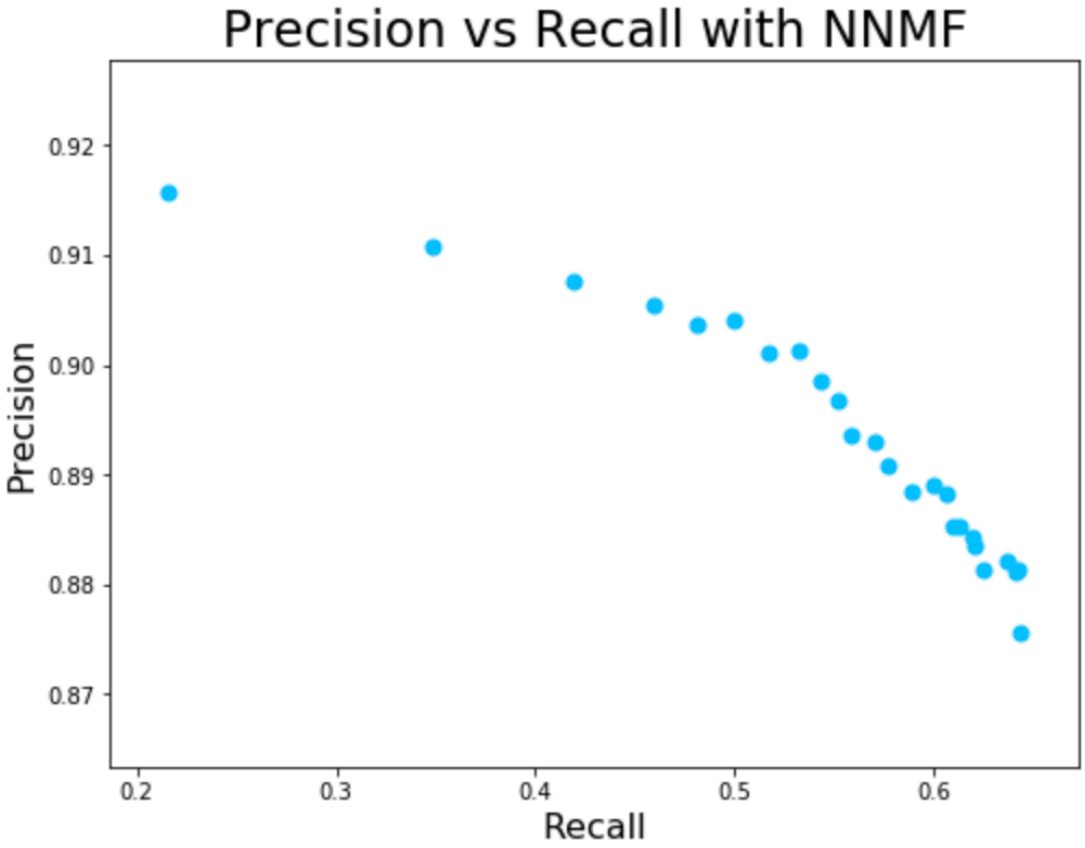


Figure 32. Precision-Recall curve with NNMF

From the figures above, we can see that the precision plot is not monotonic in t, since in equation of precision, both its numerator and denominator are functions of t and may change with t differently. While recall plot is monotonic, since only its numerator is a function of t. And the precision-recall curve is not monotonic either because of the non-monotonic precision plot.

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.

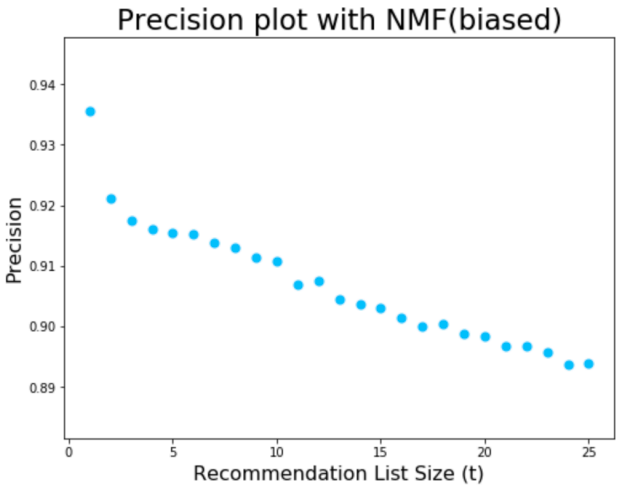


Figure 33. Precision plot with NMF (biased)

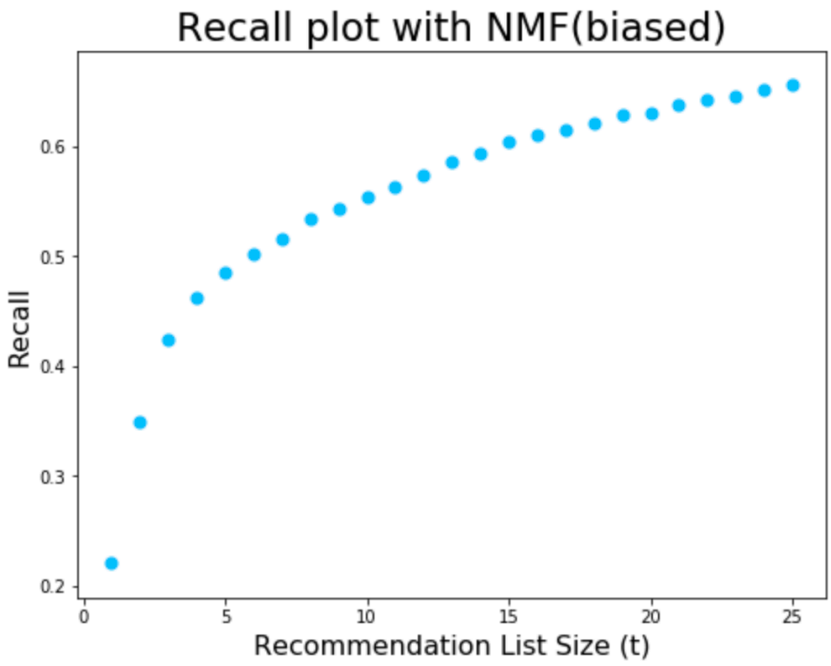


Figure 34. Recall plot with NMF (biased)

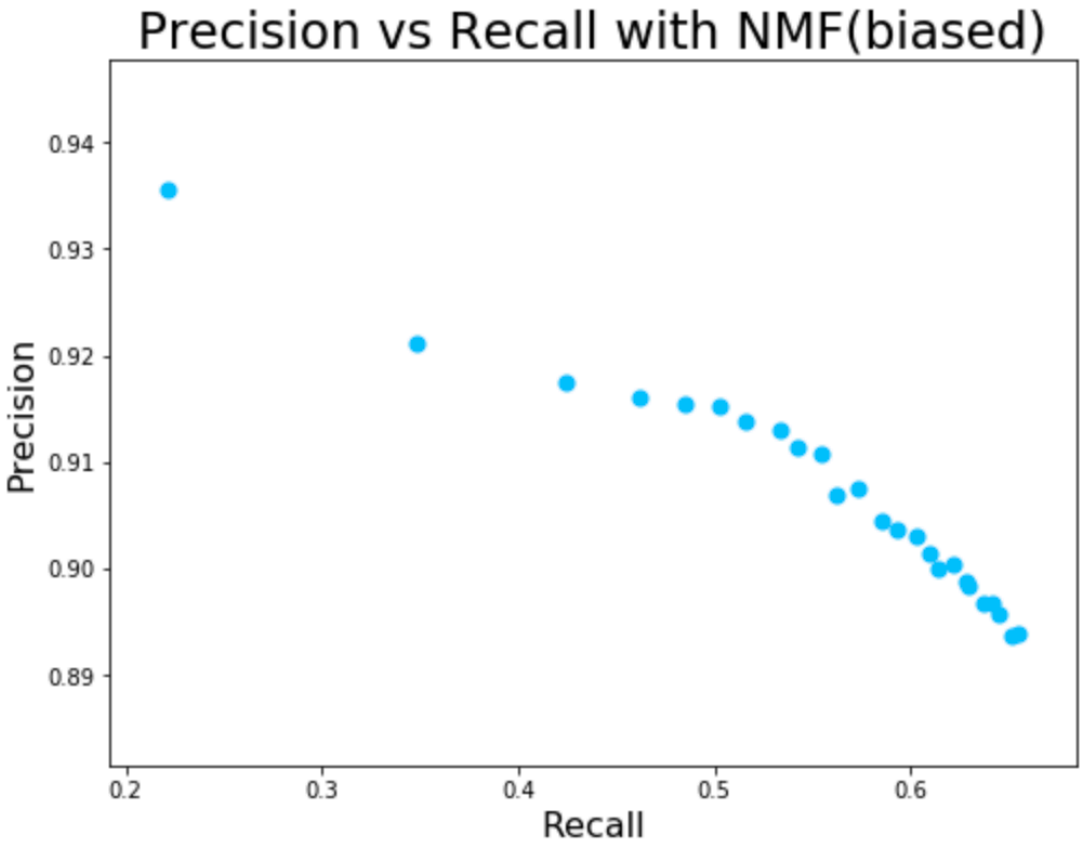


Figure 35. Precision-Recall curve with NMF (biased)

From the figures above, we can see that the precision plot is not monotonic in t, since in equation of precision, both its numerator and denominator are functions of t and may change with t differently. While recall plot is monotonic, since only its numerator is a function of t. And the precision-recall curve is not monotonic either because of the non-monotonic precision plot.

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.

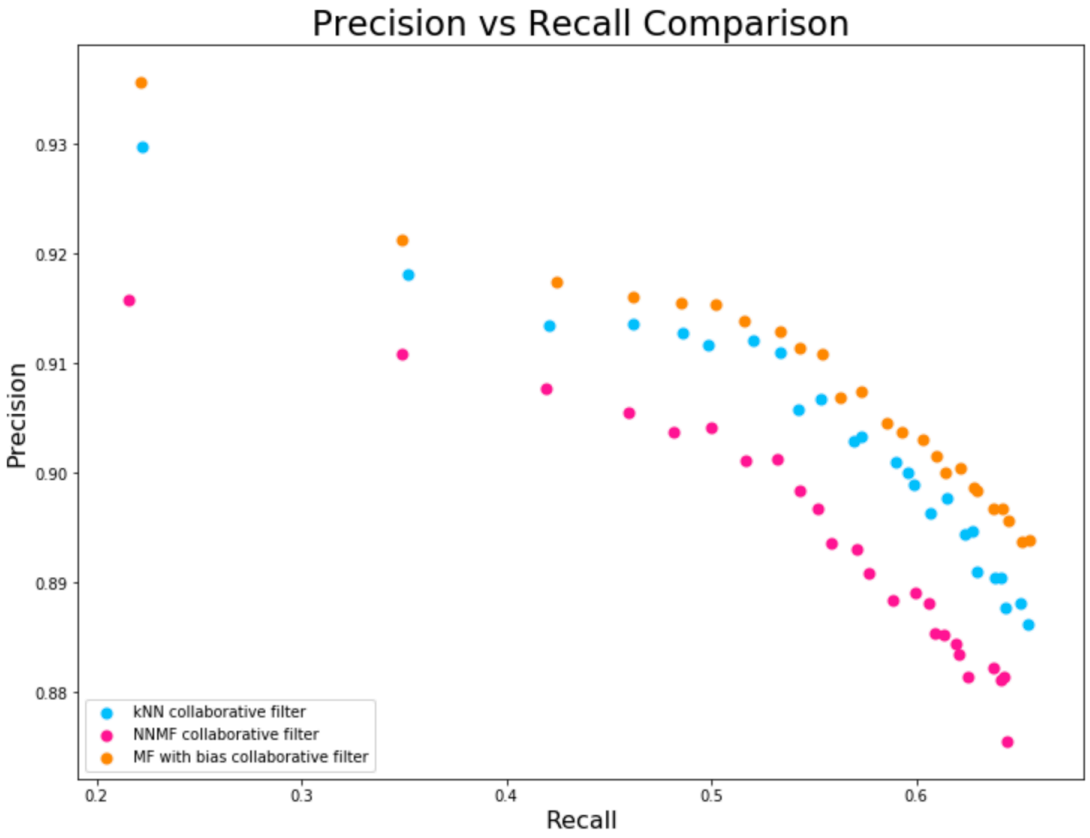


Figure 36. Precision-Recall curve comparison

From the figure above, we can see that MF with bias collaborative filter has higher precision and recall compared to the other two methods, which indicates that it has the best relevance of recommendation. NNMF collaborative filter has lowest precision and recall in these 3 methods, which achieves worse match of what users really want. The effectiveness of KNN filter is between NNMF and MF with bias filter.