EE219 Project 3
Collaborative Filtering
Winter 2018

Shaoming Cheng, 505034686
Yao Xie, 105036239
Jiahui Li, 004356402
Ruiyi Wu,304615036

1 Introduction

Recommender systems have become increasingly popular in recent years due to the rapidly growth of web applications.An increasing number of online companies are utilizing recommendation systems to increase user interaction and enrich shopping potential. Recommender systems are present in many web applications to guide our choices based on the personal interest information. The entity to which the recommendation is provided is referred to as the user, and the product being recommended is referred to as an item. Recommender systems typically produce a list of recommendations in one of two ways – through collaborative filtering or through content-based filtering. In this project, we build a recommendation system using collaborative filtering methods. Moreover, we try different collaborative filtering models on different data sets to analysis the recommendation results.

2 Collaborative Filtering Models

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. For example, a collaborative filtering recommendation system for television tastes could make predictions about which television show a user should like given a partial list of that user's tastes.

3 MovieLens Dataset

Question 1: Compute the sparsity of the movie rating dataset, where sparsity is defined by equation 1.

$$Sparsity = \frac{Total\ number\ of\ available\ ratings}{Total\ number\ of\ possible\ ratings}$$

Result:

In this task we imported pandas to read the csv files `import pandas as pd` and converted them into matrices.

The shapes of ‘ratings’ matrix is (100004, 4) and that of the ‘movies’ matrix is (9125, 3).

The sparsity is calculated according to equation 1 and the result is 0.01633285017250883.

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

Result:

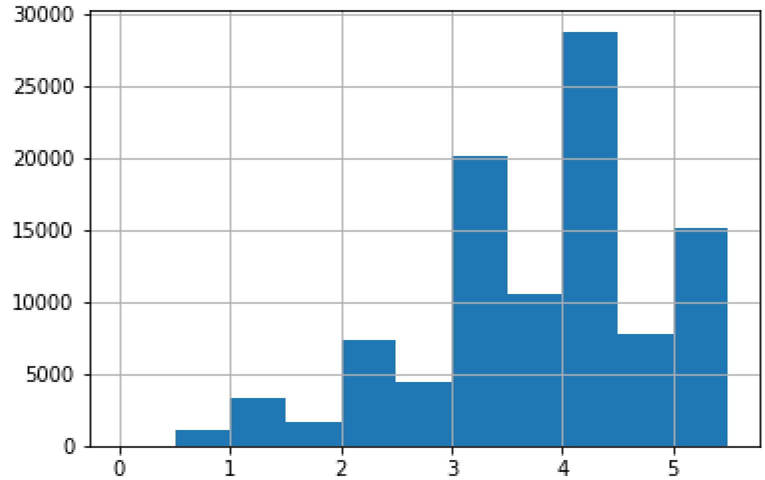


Figure 1. Histogram showing the frequency if rating values

Figure 1. Shows the frequency of rating values corresponding to all the movies listed. The horizontal axis is the rating values with the interval width of 0.5 and the vertical axis is the number of movies which belong to the intervals of the rating values. From the histogram we can know that the interval from 4 to 4.5 has the largest number of movies among all the intervals. And the interval which contains the second largest number of movies is 3 to 3.5. Interval with rating values from 0 to 0.5 has least number of movies, which can be roughly regarded as zero. About 15000 movies are rated 5.

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.

Result:

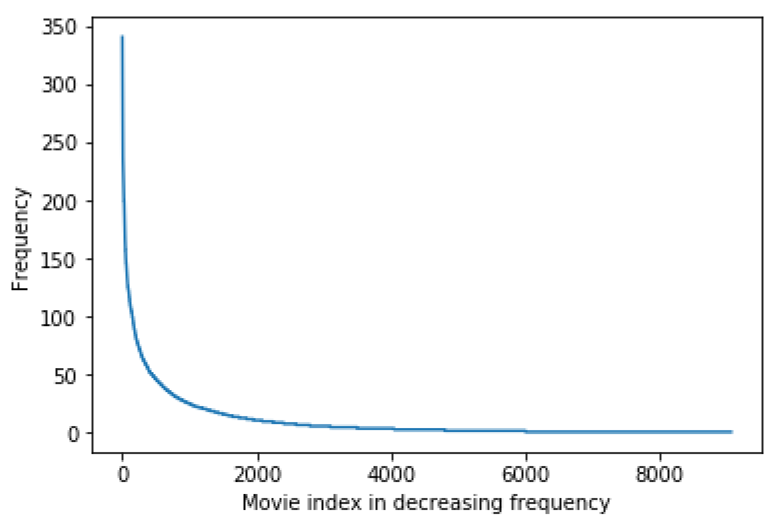


Figure 2. Distribution of ratings among movies

Figure 2 shows the distribution of ratings among movies, which is the number of ratings each movie received. The approximate number of movies in total is 9000 and the largest number of ratings one movie received is about 350.

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.

Result:

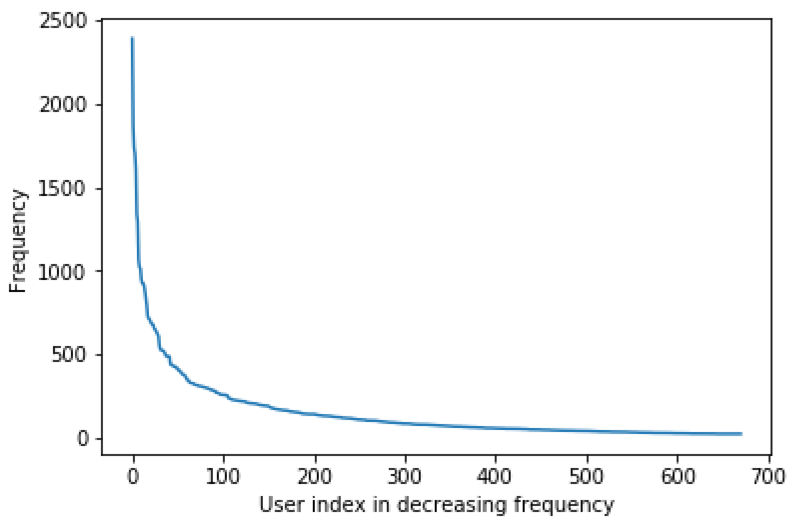


Figure 3. Distribution of ratings among users

Figure 3 shows the distribution of ratings among users, which is the number of movies each user has rated. There are more than 600 users who have given ratings to at least one movie. The user who rated most movies has about 2400 ratings among all the movies.

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

Result:

From figure 2 we can know that only few movies have received large numbers of ratings. There are about only 200 movies among 9125 movies which have more than 50 ratings. For recommendation process, movies with few ratings don't have much referential significance for users. Thus only those with many ratings should be included in the recommendation lists.

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

Result:

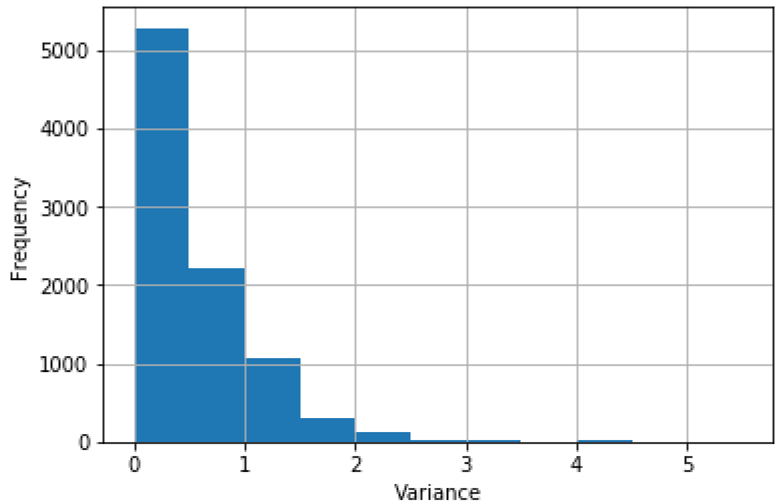


Figure 4. Distribution of the variances of the rating values received by movies

From figure 4 we can know that over 5,000 movies have variances less than 0.5. And the numbers of movies decrease with the increment of variances. In other words, the nearer the ratings are to the average rating, the more movies there are. The distribution of movies' ratings are similar to a normal distribution,

4 Neighborhood-based Collaborative Filtering

Question 7: Write down the formula for μ_u in terms of I_u and r_{uk}

Equation:

$$\mu_u = \frac{\sum_{k \in Iu} r_{uk}}{|I_u|}, \quad \forall u \in 1, \ldots, m$$

Question 8: In plain words, explain the meaning of $I_u \cap I_v$. Can $I_u \cap I_v$ =.
(Hint: Rating matrix R is sparse).

Explanation:

According to the definition, I_u is the set of item indices for which ratings have been specified by user u while I_v is that of user v. $I_u \cap I_v$ is the intersection of these two sets. In other words, this is a set of indices of items that are rated by both user u and v. If these two users have not rated even one same movie, the intersection will be empty and the set $I_u \cap I_v$ equals to $\varnothing$.

Question 9: Can you explain the reason behind mean-centering the raw ratings (rvj − μv) in the prediction function? (Hint: Consider users who either rate all items highly or rate all items poorly and the impact of these users on the prediction function)

Explanation:

The weighted average of the mean-centered rating of an item in the top-k peer group of the target user u is used to provide a mean-centered prediction. After this, the mean rating of the target users will be added back and provides a raw rating prediction $\widehat{r}_{uj}$ of target users u for item j. Pu is a set of k closest users to the target user u who have rated item j. After applying mean-centering, users who have high or poor ratings for all items, in other words, users with very low or negative correlations with the target user u, are sometimes filtered from Pu as a heuristic enhancement. This approach allows for a number of different variations in terms of how the similarity or prediction function is computed or in terms of which items are filtered out during the prediction process.

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).

Result:

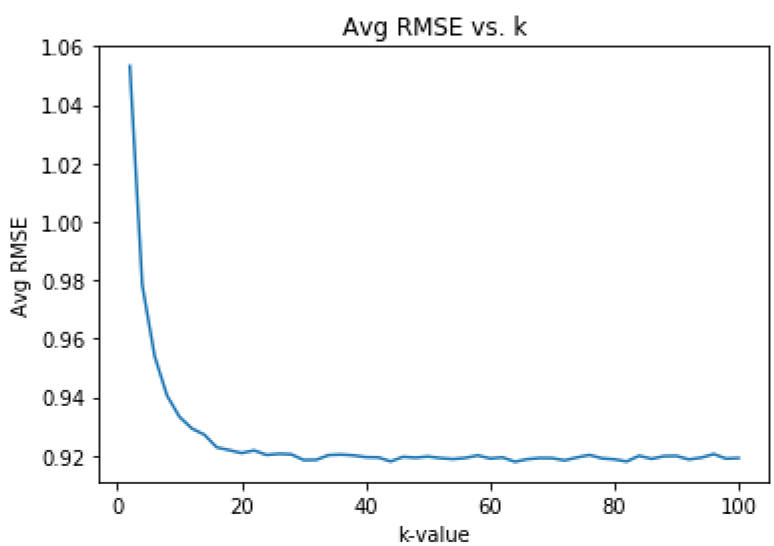


Figure 5. Average RMSE value against k

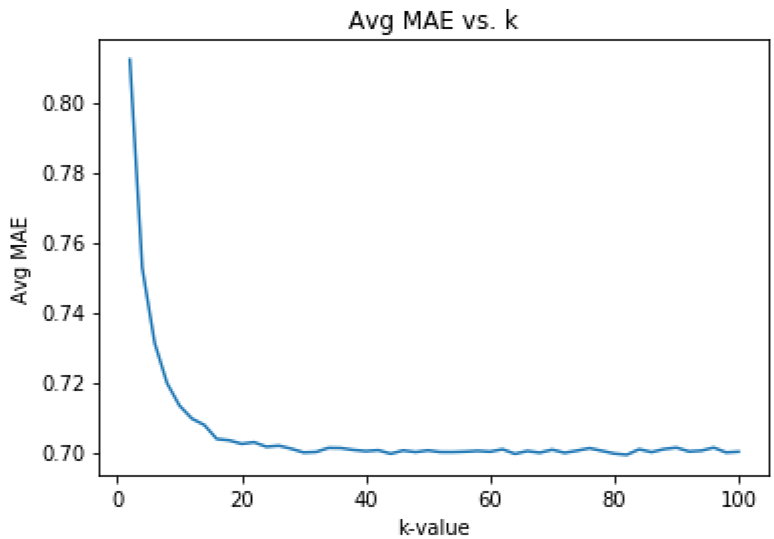


Figure 6. Average MAE value against k

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

Result:

K=16

```
Best k for MAE is index: 7 , with value  16.0
Best k for RMSE is index: 6 , with value 14.0
Minimum K = 16
```

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

Result:

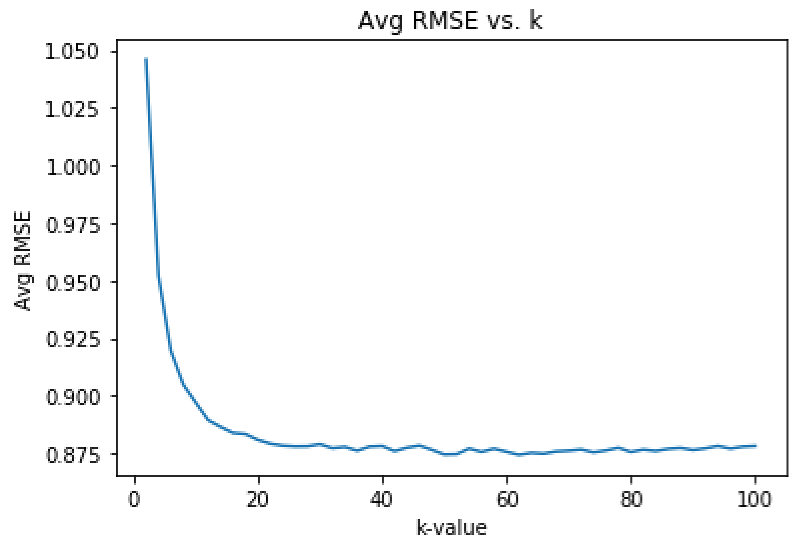


Figure 7. Average RMSE value against k in the popular movie trimmed test set

```
Minimum average RMSE = 0.874372365937651
```

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

Result:

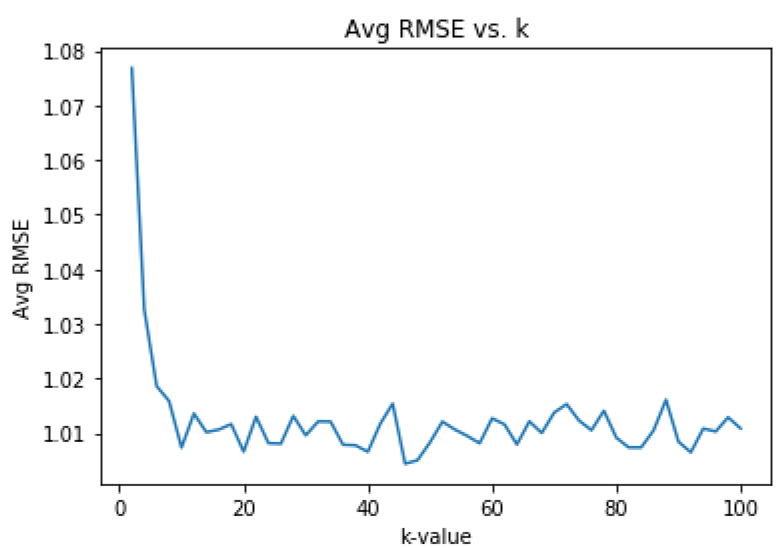


Figure 8. Average RMSE value against k in the unpopular movie trimmed test set

```
Minimum average RMSE = 1.0043608846615322
```

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

Result:

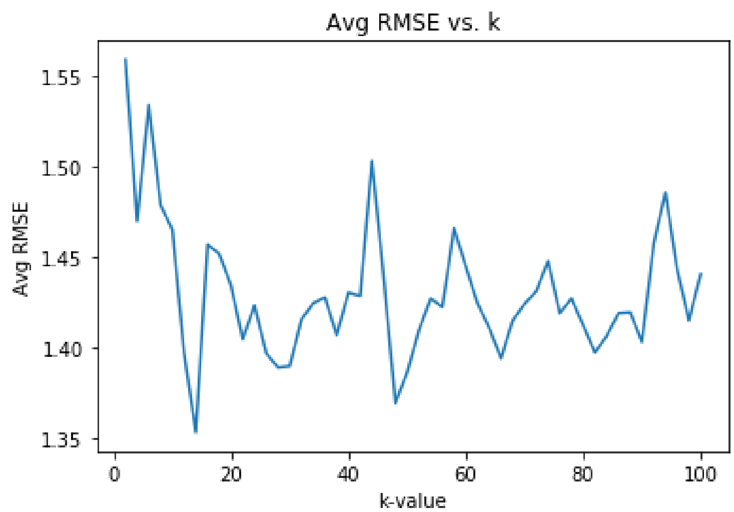


Figure 9. Average RMSE value against k in the high variance movie trimmed test set

```
Minimum average RMSE = 1.3532458392220084
```

Question 15: Plot the ROC curves for the k-NN collaborative filter 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.

Result:

```
K = 16
```

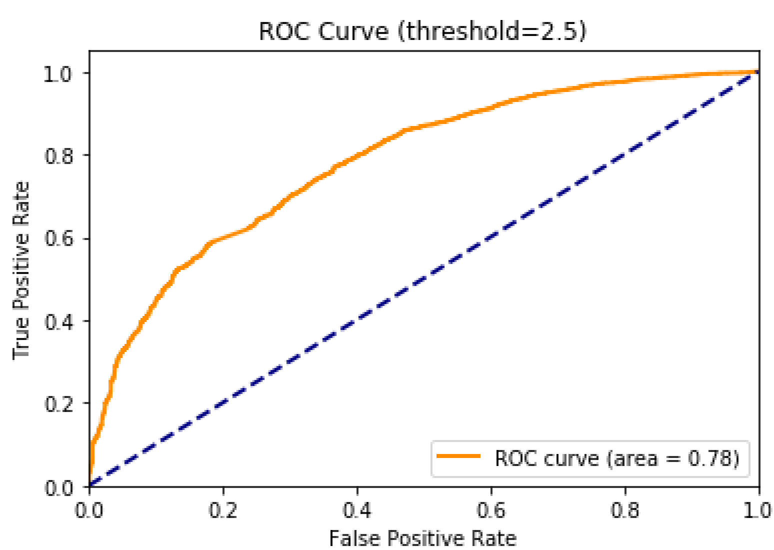


Figure 10. The ROC curves for the k-NN collaborative filter with threshold value = 2.5

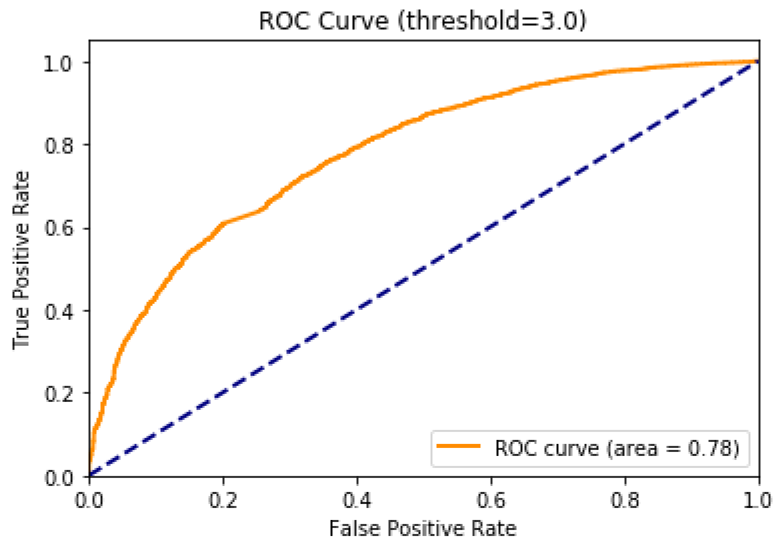


Figure 11. The ROC curves for the k-NN collaborative filter with threshold value 3.0

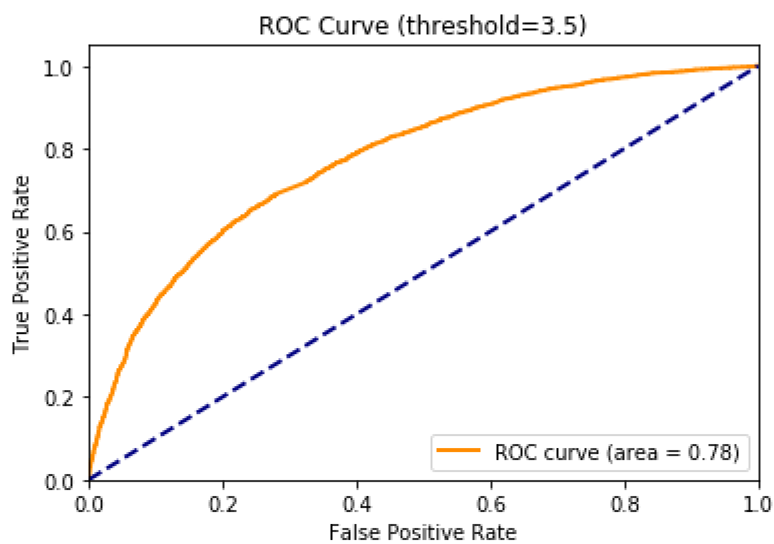


Figure 12. The ROC curves for the k-NN collaborative filter with threshold value 3.5

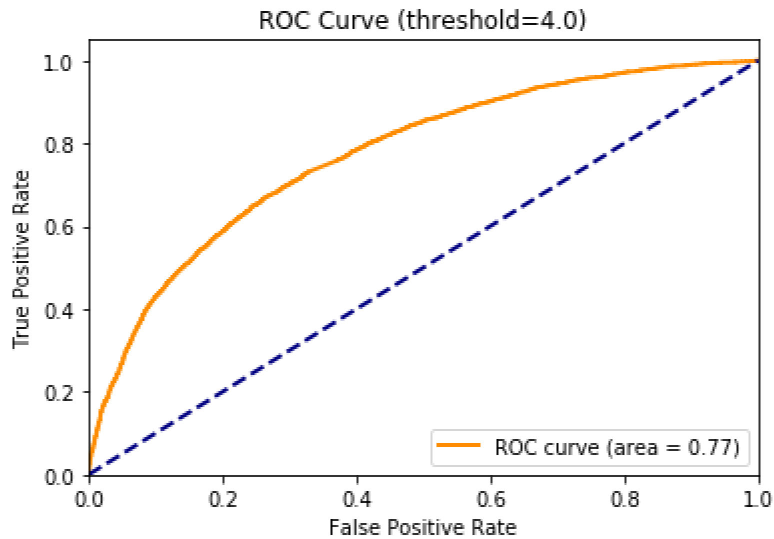


Figure 13. The ROC curves for the k-NN collaborative filter with threshold value 4.0

5 Model-based Collaborative Filtering

5.1 Non-negative matrix factorization (NNMF)

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.

Answer:

No. The optimization problem given by equation 5 is not a convex optimization problem because the function UV^T is not a convex function. For U fixed, the optimization problem can be formulated as a least-squares problem in the following form:

Let u_i be the i^{th} row of matrix U and v_j be the j^{th} row of matrix V. Then, equation 5 can be expressed as the following:

$$\underset{V}{\text{minimize}} \quad \sum_{i=0}^{m} \sum_{j=0}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2 =$$

$$\underset{v_1, v_2, \ldots, v_n}{\text{minimize}} \quad \sum_{i=0}^{m} \sum_{j=0}^{n} W_{ij}(r_{ij} - u_i v_j^T)^2 =$$

$$\underset{v_1, v_2, \ldots, v_n}{\text{minimize}} \quad \sum_{i=0}^{m} \sum_{j=0}^{n} (W_{ij} r_{ij} - W_{ij} u_i v_j^T)^2 =$$

$$\underset{v_1, v_2, \ldots, v_n}{\text{minimize}} \quad \sum_{i=0}^{m} \sum_{j=0}^{n} (W_{ij} u_i v_j^T - W_{ij} r_{ij})^2$$

The part of the cost function related to the variable v_j can be rewritten as

$$f_j(v_j) = \left\| \tilde{U}_j v_j - \tilde{r}_j \right\|,$$

where $\tilde{U}_j \in \mathbb{R}^{m \times k}$ and $\tilde{a}_j \in \mathbb{R}^m$ are defined as

$$\tilde{U}_j = \begin{bmatrix} W_{1j} U_{1,1:k} \\ W_{2j} U_{2,1:k} \\ \vdots \\ W_{mj} U_{m,1:k} \end{bmatrix}, \tilde{r}_j = \begin{bmatrix} W_{1j} r_{1j} \\ W_{2j} r_{2j} \\ \vdots \\ W_{mj} r_{mj} \end{bmatrix}$$

The original optimization problem can be rewritten as a least squares problem

$$\text{minimize} \left\| \begin{bmatrix} \tilde{U}_1 & & \\ & \ddots & \\ & & \tilde{U}_n \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} - \begin{bmatrix} \tilde{r}_1 \\ \vdots \\ \tilde{r}_n \end{bmatrix} \right\|^2,$$

where the variables is $(v_1, \ldots v_n) \in \mathbb{R}^{kn}$

5.1.1 Design and test via cross-validation

Question 17: Design a NNMF-based collaborative filter to predict the ratings of the movies in the MovieLens dataset 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.

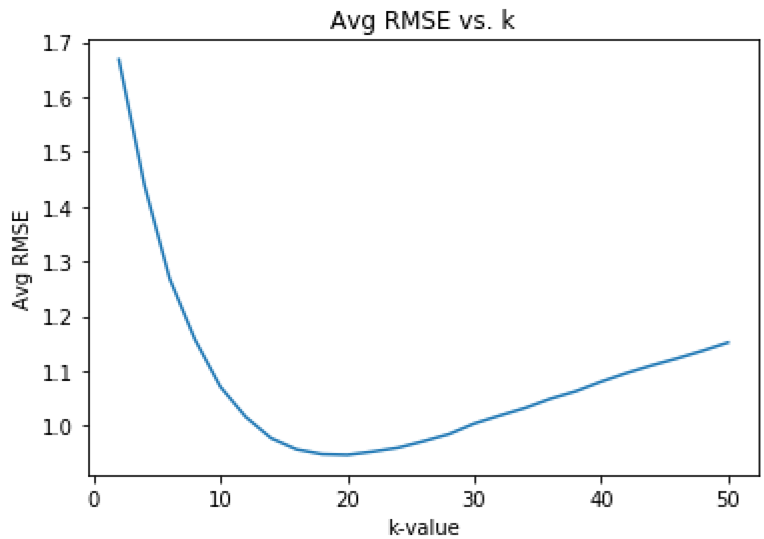


Figure 14. Average RMSE vs. k for NNMF Collaborative Filter

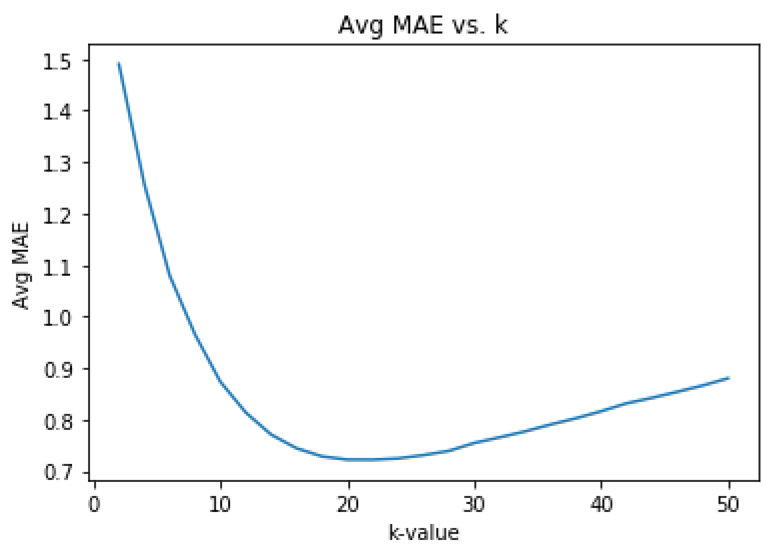


Figure 15. Average MAE vs. k for NNMF Collaborative Filter

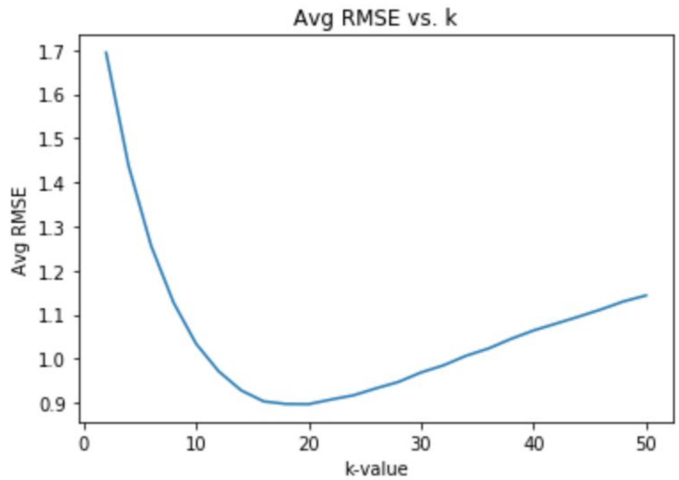
Question 18: Use the plot from question 17, to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE. Is the optimal number of latent factors same as the number of movie genres?

```
Best k for MAE is index: 7 , with value  16.0
Best k for RMSE is index: 6 , with value 14.0
Minimum K = 16
```

The optimal number of latent factors is 16 for RMSE and 14 for MAE, while the number of movie genres is 19, with the late one being “no genres listed”, which is close enough for the optimal number of latent factors.

5.1.2 NNMF filter performance on trimmed test set

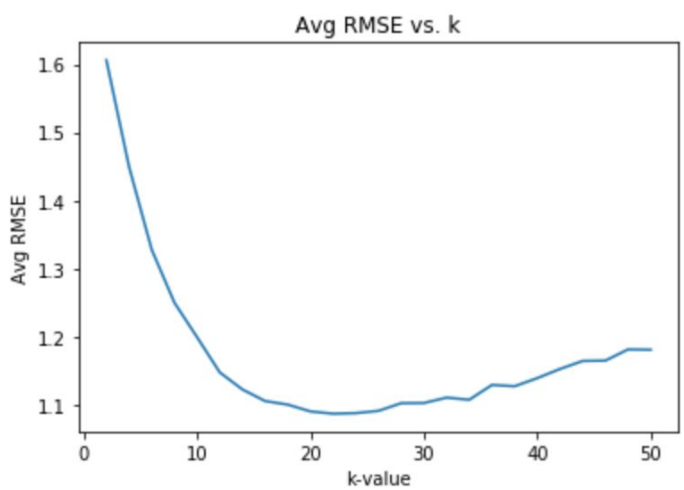
Question 19: Design a NNMF collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE




```
Minimum avg rmse = 0.8971
```

Figure 16. Average RMSE value against k in the popular movie trimmed test set for NNMF

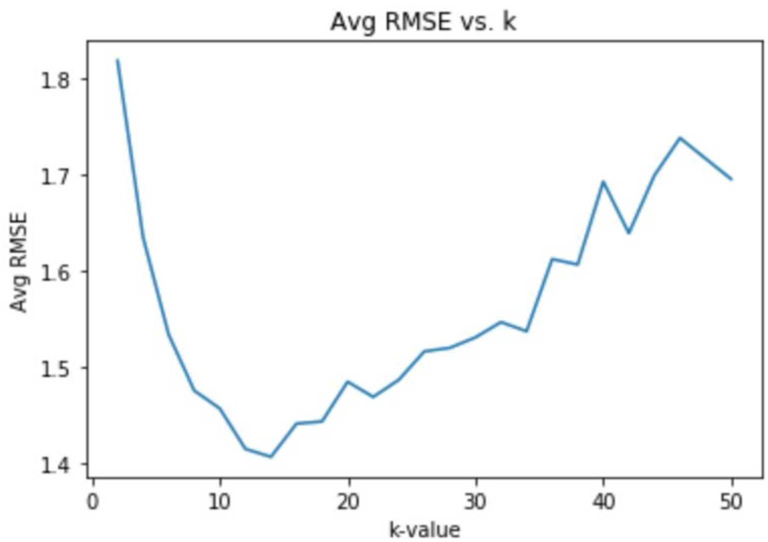
Question 20: Design a NNMF collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate it's performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE




```
Minimum avg rmse = 1.0867
```

Figure 17. Average RMSE value against k in the unpopular movie trimmed test set for NNMF

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.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 avg rmse = 1.4067
```

Figure 18. Average RMSE value against k in the high variance movie trimmed test set for NNMF

5.1.3 Performance evaluation using ROC curve

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 plot- ting 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.

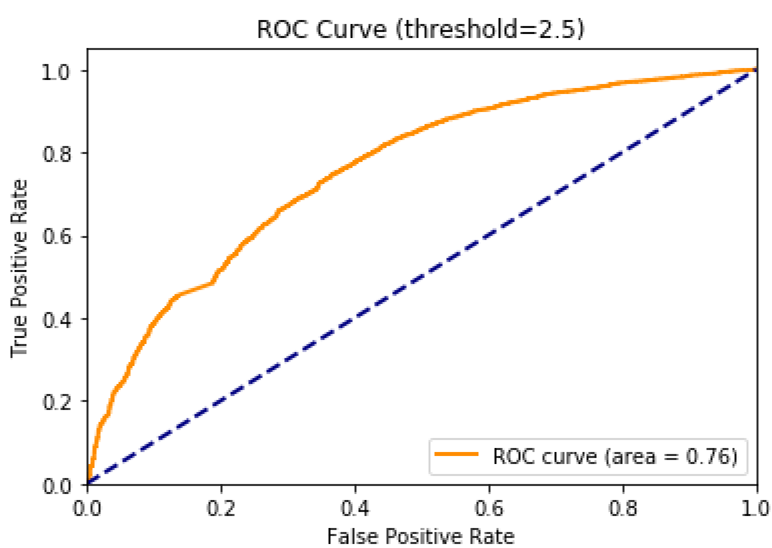


Figure 19. The ROC curves for the NNMF collaborative filter with threshold value 2.5

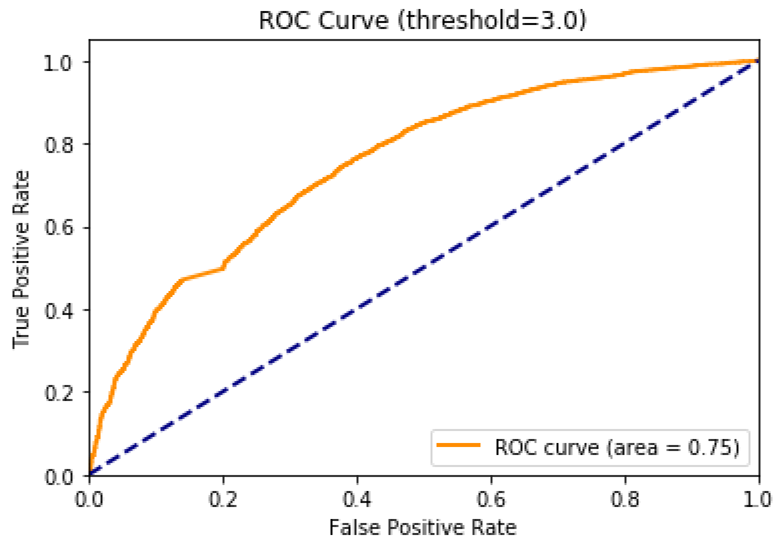


Figure 20. The ROC curves for the NNMF collaborative filter with threshold value 3.0

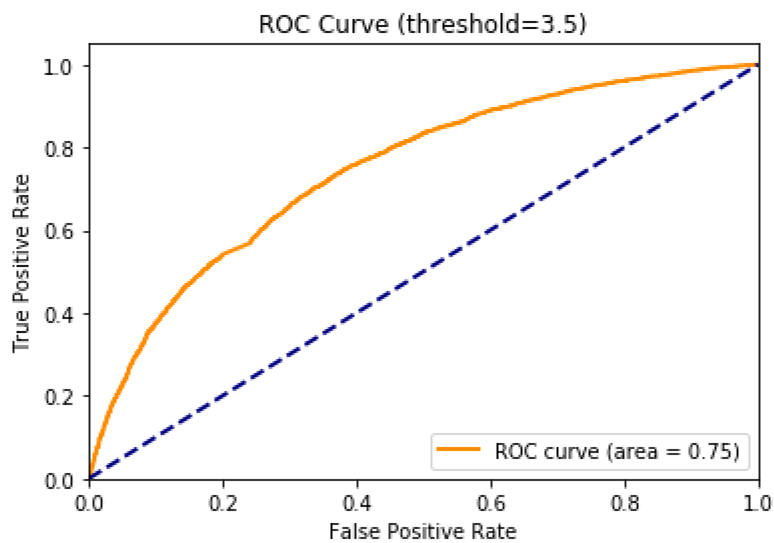


Figure 21. The ROC curves for the NNMF collaborative filter with threshold value 3.5

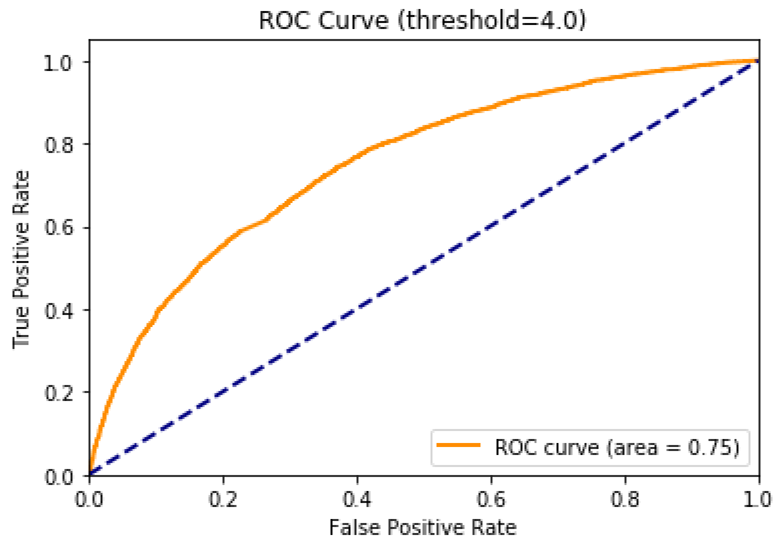


Figure 22. The ROC curves for the NNMF collaborative filter with threshold value 4.0

5.1.4 Interpretability of NNMF

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?

```
(9066, 20)
[6219, 4143, 4603, 2887, 6450, 5106, 6797, 33817, 6686, 4520]
['Action|Comedy', 'Horror|Mystery', 'Comedy', 'Comedy|Romance', 'Comedy|Musical|Romance', 'Ac
tion|Drama|Thriller', 'Drama|War', 'Action|Comedy|Crime|Fantasy', 'Crime|Drama', 'Drama|Roman
ce']
```

The top 10 movies are primarily comedy, drama and action movies in the first column, which means that the latent feature of first column selects movies that have a general mix of these three movie genres. Similarly, for second column, the top 10 movies belong to a small collection generally consisting of romance and crime, again indicating that these latent feature factors select movie groups with a mix of these movie genres.

5.2 Matrix factorization with bias (MF with bias)

5.2.1 Design and test via cross-validation

Question 24: Design a MF with bias 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.

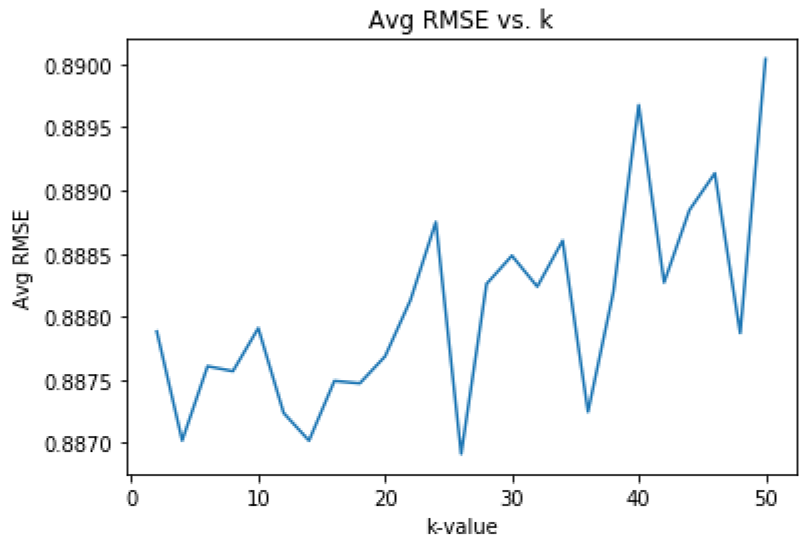


Figure 23. Average RMSE vs. k for MF with bias collaborative filter

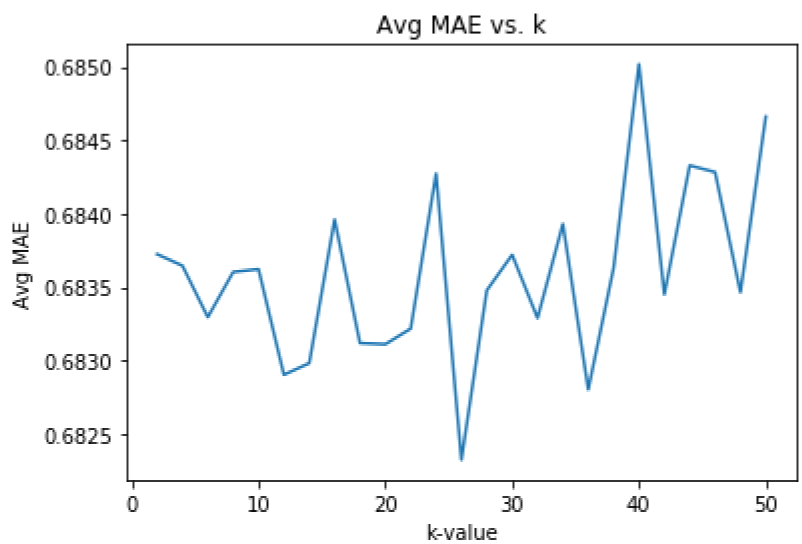


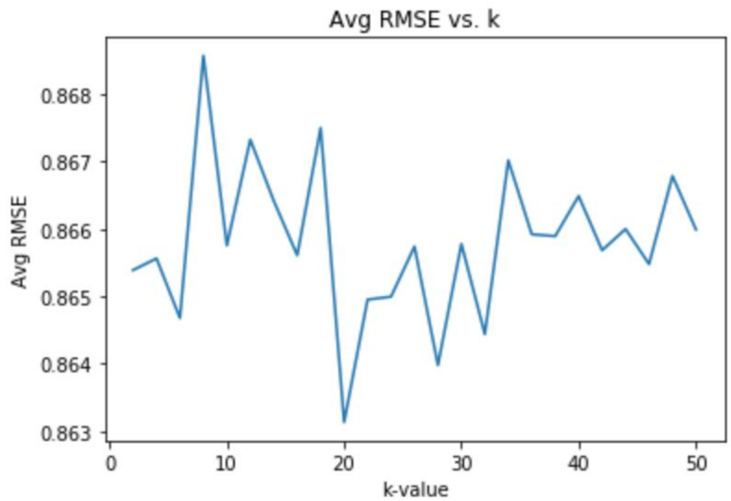
Figure 24. Average MAE vs. k for MF with bias collaborative filter

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.

```
Best k for MAE is index: 12 , with value  26.0
Best k for RMSE is index: 17 , with value 36.0
Minimum K = 36
```

5.2.2 MF with bias filter performance on trimmed test set

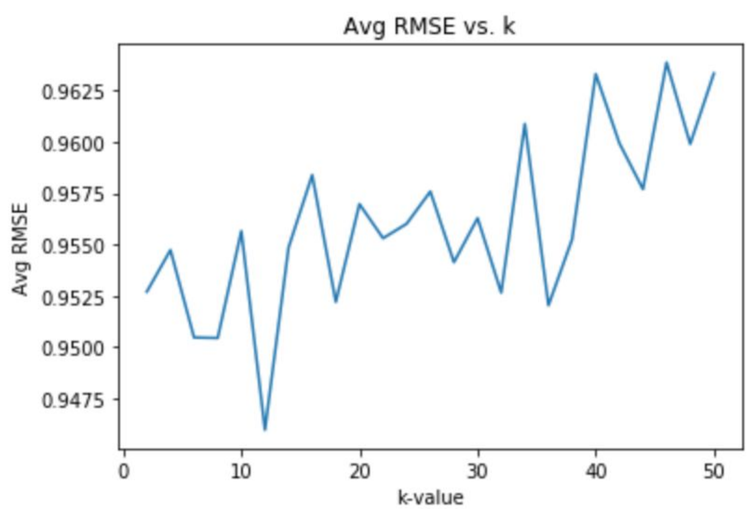
Question 26: Design a MF with bias collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate it's performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE




```
Minimum avg rmse = 0.8631
```

Figure 25. Average RMSE value against k in the popular movie trimmed test set for MF with bias

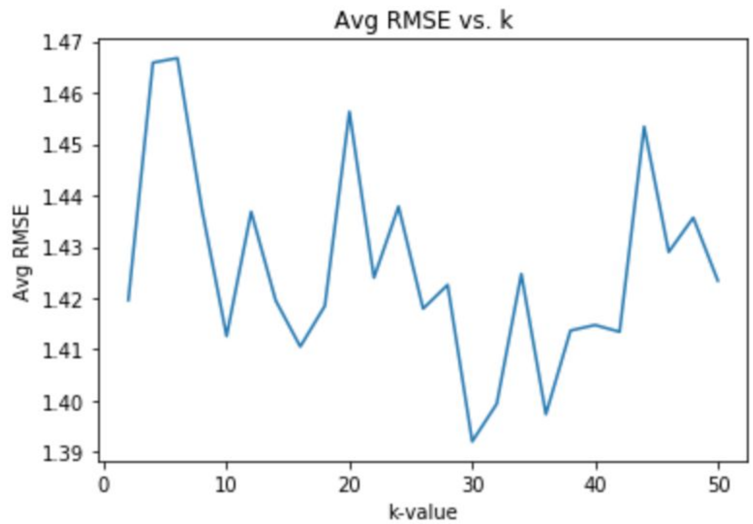
Question 27: Design a MF with bias collaborative filter to predict the rat- ings of the movies in the unpopular movie trimmed test set and evaluate it's performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE ob- tained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE




```
Minimum avg rmse = 0.9460
```

Figure 26. Average RMSE value against k in the unpopular movie trimmed test set for MF with bias

Question 28: Design a MF with bias collaborative filter to predict the rat- ings 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 ob- tained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE




```
Minimum avg rmse = 1.3921
```

Figure 27. Average RMSE value against k in the high variance movie trimmed test set for MF with bias

5.2.3 Performance evaluation using ROC curve

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 plot- ting 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.

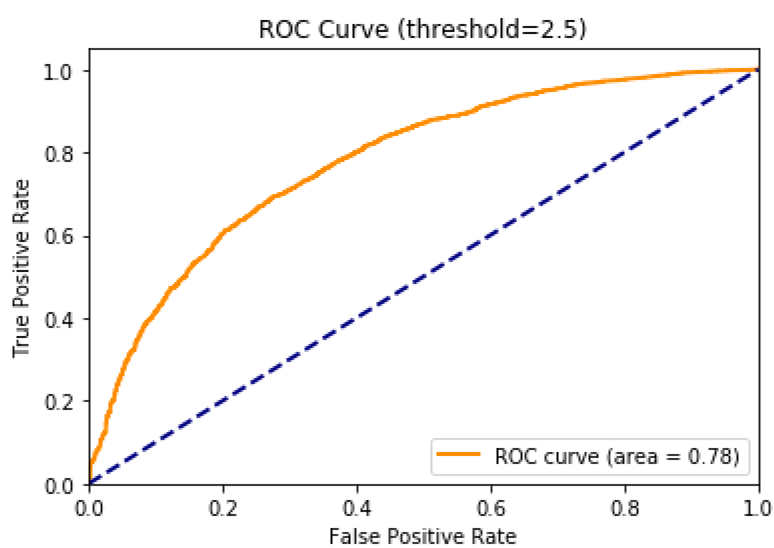


Figure 28. The ROC curves for the MF with bias collaborative filter with threshold value 2.5

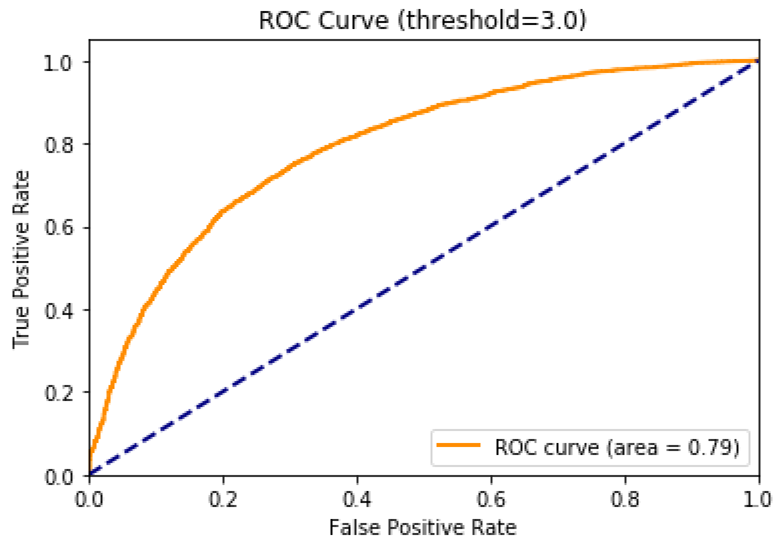


Figure 29. The ROC curves for the MF collaborative filter with threshold value 3.0

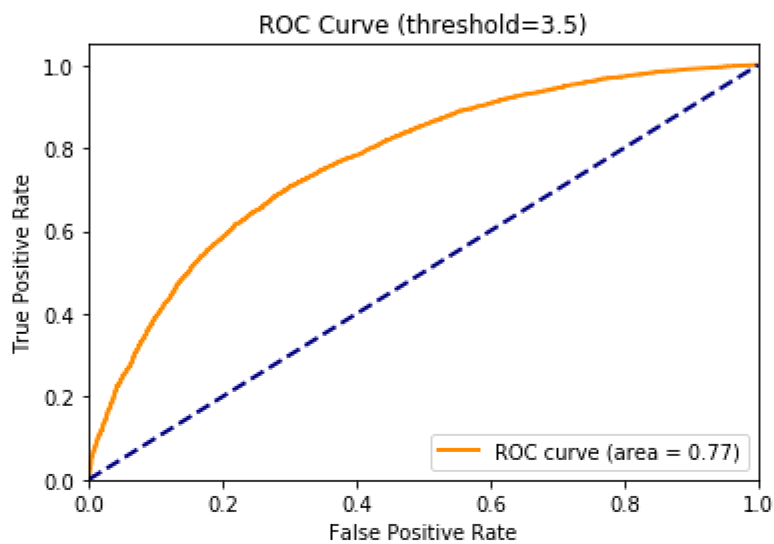


Figure 30. The ROC curves for the MF collaborative filter with threshold value 3.5

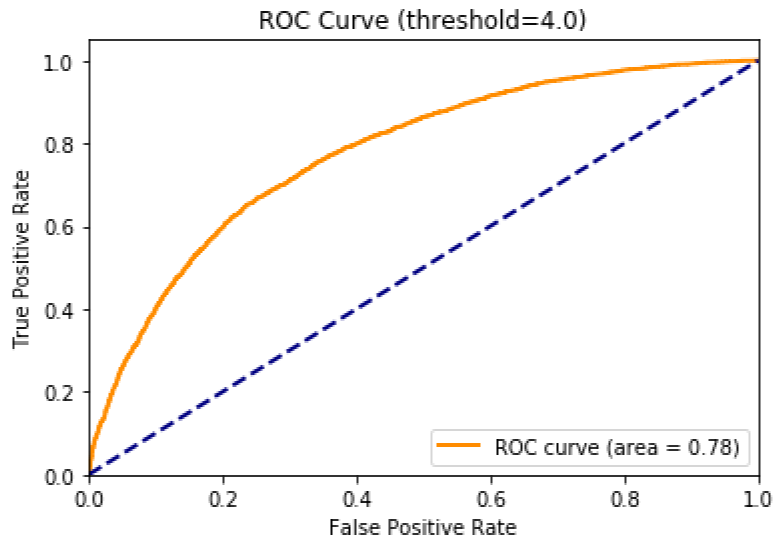


Figure 31. The ROC curves for the MF collaborative filter with threshold value 4.0

6 Naive Collaborative Filtering

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.

```
Average RMSE = 1.3830372661290398
```

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. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

```
Average RMSE = 1.4183049759785384
```

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. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

```
Average RMSE = 1.2909739122289225
```

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. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

```
Average RMSE = 1.5968301985877393
```

Explanation:
As what we expected, naive collaborative filtering yields the worst results compared to other methods. Among all the test sets, high variance movie trimmed test set has the worst result. It makes sense since the data set only contains the movies with high variance which means people usually have different ideas about the movies. Thus, a navies collaborative filtering will not be able to predict an accurate result.

On the other hand, the unpopular movie trimmed test set has the best results. The test set in this group only contains the popular movie which makes the prediction more accurate.

7 Performance Comparison

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.

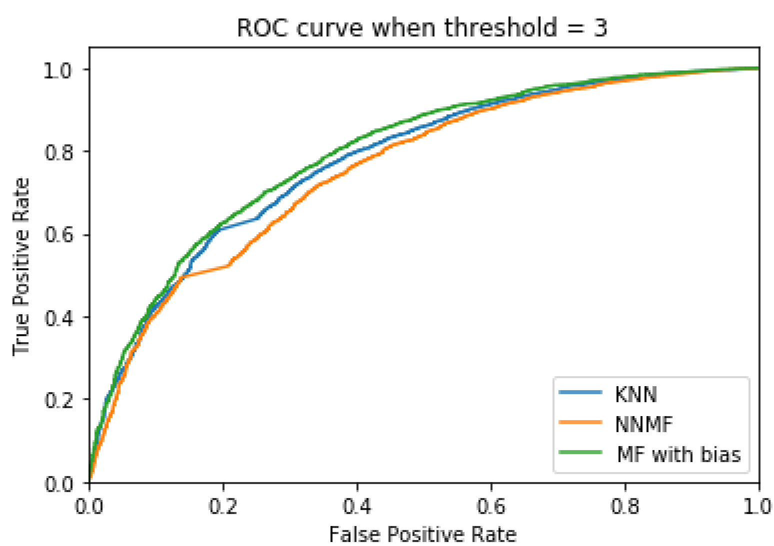


Figure 32. The ROC curves for the KNN, NNMF and MF with bias collaborative filters with threshold value 3.0

As we can see from the ROC curves of these three collaborative filters, MF with bias has better performance of predicting the rating of movies than the other two methods.

8 Ranking

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.

Explanation:
Precision is the set of movies both liked by users and recommended in the list divided by the set of items in the recommendation list. In other words, precision is the number of

correct results divided by the number of all returned results. It is a fraction refers to the percentage of correction in the returned results.

Recall is the set of movies both liked by users and recommended in the list divided by the set of all movies liked by the users (ground truth positive). In other words, it is the number of correct results divided by the number of results that should be returned. It is the percentage of all results should be returned that are actually returned.

For example, condition positive (P) means the number of real positive cases in the data and condition negatives (N) means the number of real negative cases in the data. Thus true positive (TP), true negative (TN), false positive (FP) and false negative (FN) are corresponding to items with ground truth positive and predicted as positive, with ground truth positive and predicted as negative, with ground truth negative and predicted as positive and with ground truth negative and predicted as negative, respectively.
Precision = $\frac{TP}{TP+FP}$
Recall = $\frac{TP}{TP+FN}$

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). 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.

Result:

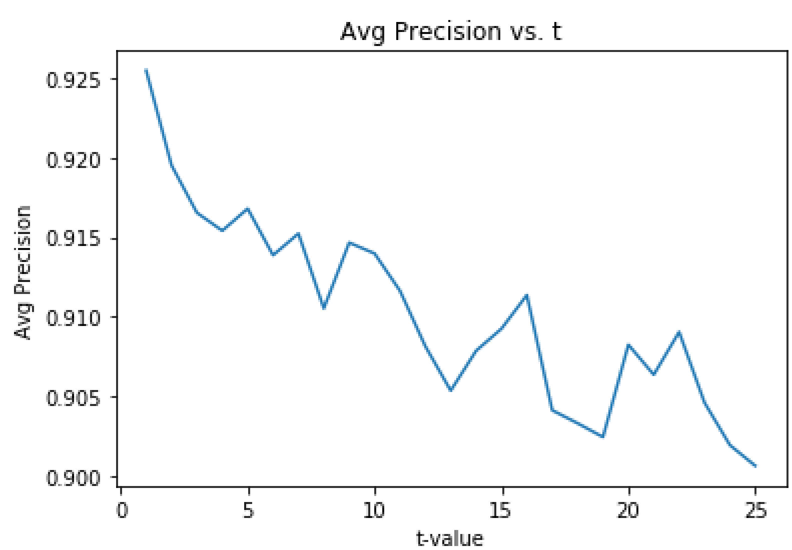


Figure 33. Average precision vs. t using k-NN collaborative filter prediction

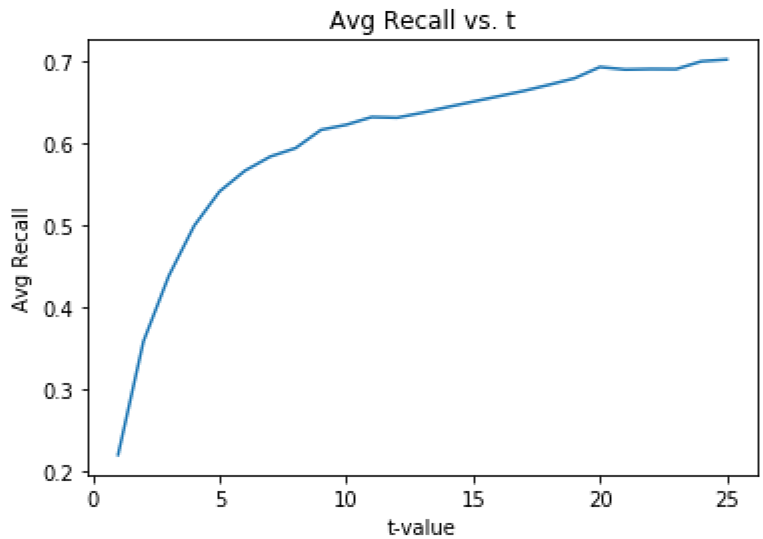


Figure 34. Average recall vs. t using k-NN collaborative filter prediction

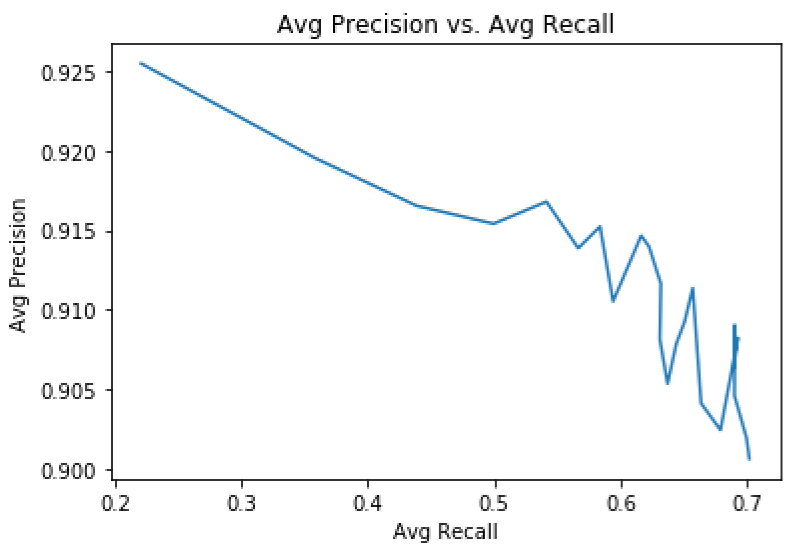


Figure 35. Average precision vs. average recall using k-NN collaborative filter prediction

From figures of K-NN above, the plots of Average precision vs. t, Average recall vs. t and Average precision vs. average recall are shown. From figure 33 the overall average precision decreases as t increases while it fluctuates at some values of t. Figure 34 shows that the average recall increases with the value of t. In the average precision vs. average recall plot, in general, the larger the recall is, the smaller the precision is. However, fluctuation still exists.

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.

Result:

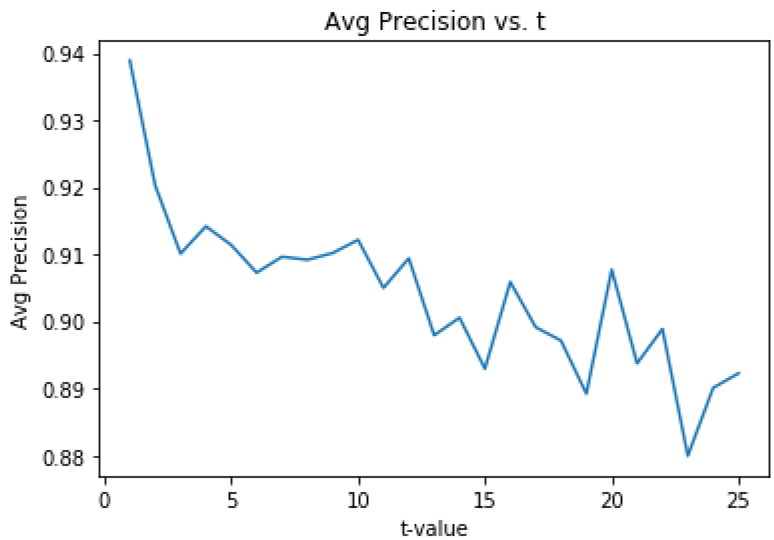


Figure 36. Average precision vs. t using NNMF-based collaborative filter prediction

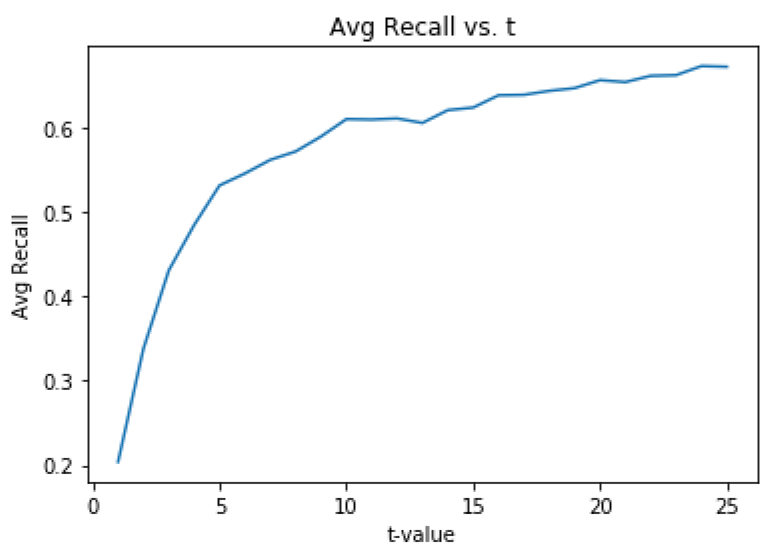


Figure 37. Average recall vs. t using NNMF-based collaborative filter prediction

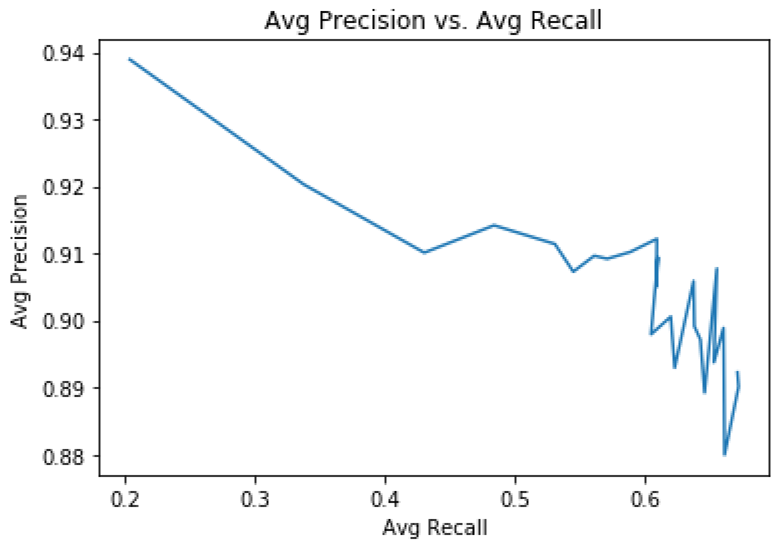


Figure 38. Average precision vs. average recall using NNMF-based collaborative filter prediction

From figures above, the plots of Average precision vs. t, Average recall vs. t and Average precision vs. average recall of NNMF are shown. They are similar of those of K-NN. From figure 36 the overall average precision decreases as t increases while it fluctuates at some values of t. Figure 37 shows that the average recall increases with the value of t. In the average precision vs. average recall plot, in general, the larger the recall is, the smaller the precision is. However, fluctuation still exists.

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.

Result:

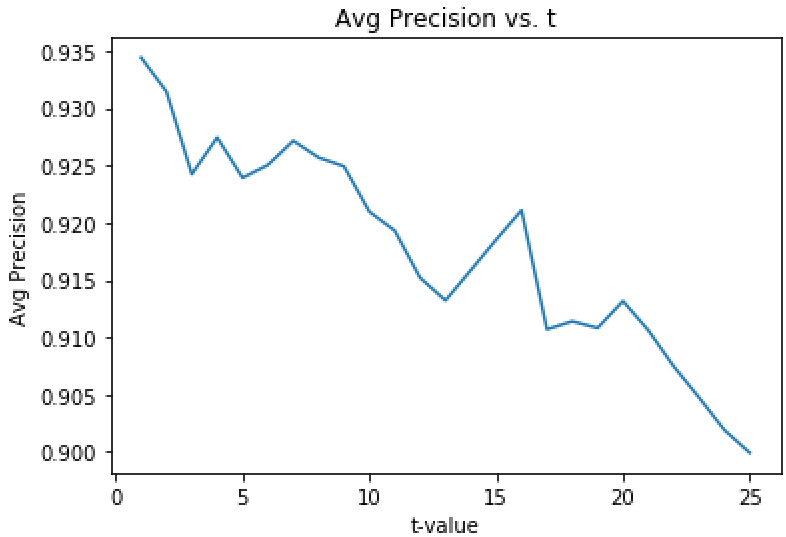


Figure 39. Average precision vs. t using MF with bias-based collaborative filter prediction

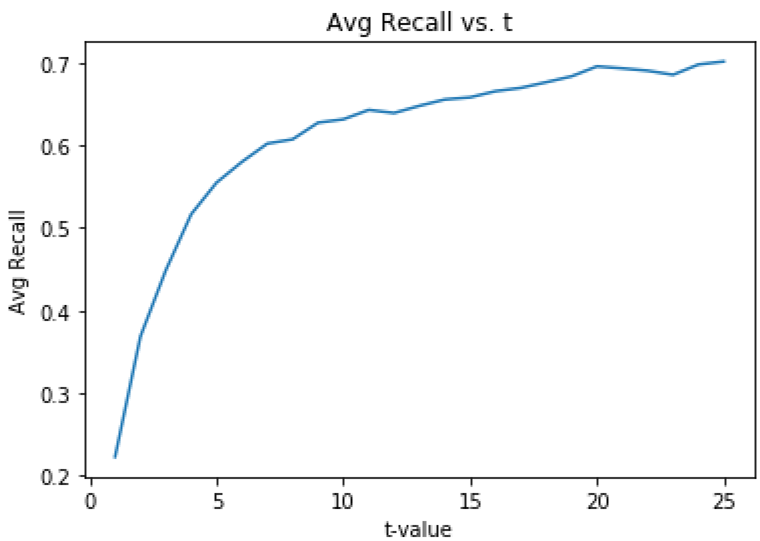


Figure 40. Average recall vs. t using MF with bias-based collaborative filter prediction

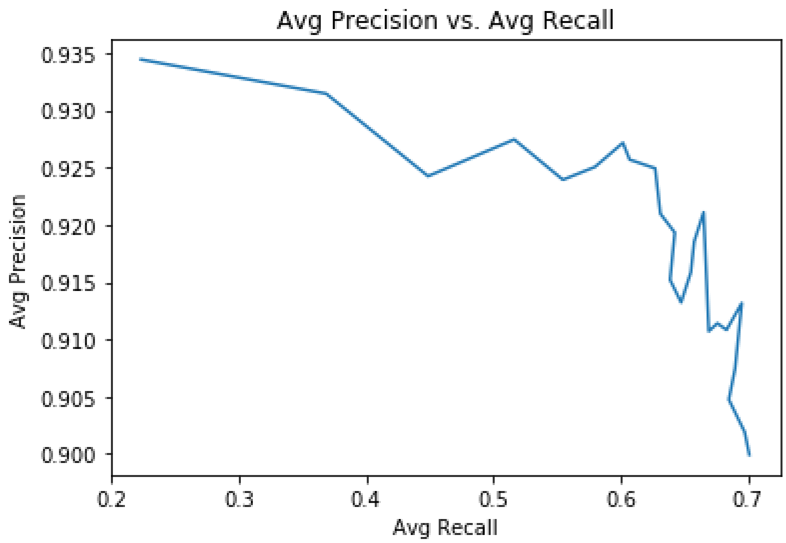


Figure 41. Average precision vs. average recall using MF with bias-based collaborative filter prediction

From figures above, the plots of Average precision vs. t, Average recall vs. t and Average precision vs. average recall of MF are shown. They are similar of those of K-NN and NNMF, too. From figure 39 the overall average precision decreases as t increases while it fluctuates at some values of t. Figure 40 shows that the average recall increases with the value of t. In the average precision vs. average recall plot, in general, the larger the recall is, the smaller the precision is. However, fluctuation still exists.

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.

Result:

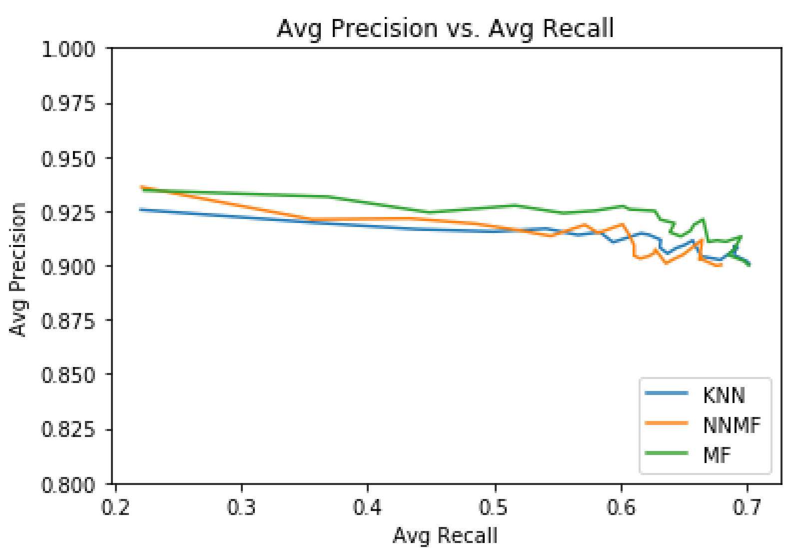


Figure 42. Average precision vs. average recall using MF with bias-based collaborative filter prediction

Figure 42 shows the comparison of the average precision vs. average recall plots for K-NN, NNMF and MF with bia predictions. These three plots are similar, in which the values of average precision decreases when average recall increases and there are fluctuations when the values of average recall are high. All the three methods have average precision range from 0.900 to 0.950 and have average recall from 0.2 to 0.7.

There are mainly two ways to compare the algorithms. The first is average comparison, the higher the value of (P+R)/2 is, the better an algorithm is. This method is not good in some cases, especially when either of P or R is high and the other is nearly 0. However, it works well in this problem. The second method is F1 score comparison. $F1 = 2\frac{PR}{P+R}$. The larger the F1 score is, the better the performance of an algorithm is.

For convenience, we use the average comparison. When the value of average recall is unchanged, MF with bias has the highest average precision and K-NN has the lowest precision among all the three methods. However, when the value of the average precision is unchanged, MF has the highest value of average recall and K-NN has the lowest average recall. In conclusion, MF with bias prediction has the best performance and NNMF has a relative better performance than K-NN.