ECE 219
Large Scale Data Mining: Models and Algorithm

Project 3
Collaborative Filtering

Haitao Wang (UID: 504294402)
Weiqian Xu (UID: 404297854)
Xinyi Gu (UID: 205034975)
Yutan Gu(UID: 005034976)

3. MovieLens Dataset

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

$$Sparsity = \frac{\text{Total number of available ratings}}{\text{Total number of possible ratings}} \quad (1)$$

In this part, our program found there are a total of 10004 samples, 9066 different movies and 671 different users. To compute the sparsity of the dataset, we divide the total amount of samples by product of total amount of movies and users. The sparsity of the data set is 0.016439.

Question 2: Plot a histogram showing the frequency of the ratings 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.

The distribution of ratings we found are shown in the figure below. We can discover that most of the ratings are above 2.5. Furthermore, 4 receive the most amount of ratings.

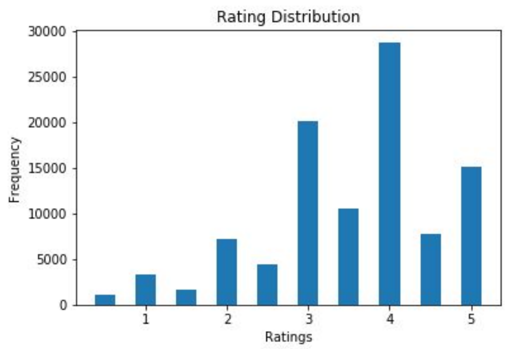


Figure 1: Plot for distribution of ratings

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.

The frequency of ratings of different movies is shown in the figure below. We can see that only a small portion of movie receives the amount ratings higher than 50.

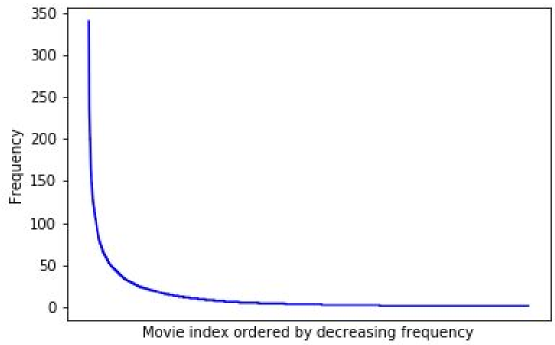


Figure 2: Plot for movie frequency

Question 4: Plot the distribution of ratings amount 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.

The result is shown in the figure below. As one can discover, only a small portion of user generate the most of the ratings.

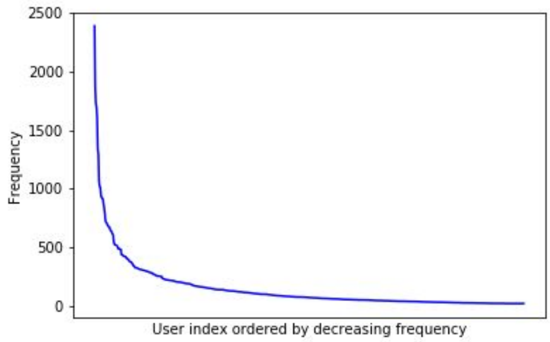


Figure 3: Plot for user rating frequency

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

It is very obvious that a small portion of movie receives a large amount of ratings by users. For user-based models, it is very likely that different users generate ratings on this portion of movies and they can be critical for the collaborating filtering algorithms in this project to relate users with similar tastes.

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 vinned intervals and use this count as the vertical axis. Briefly comment on the shape of the histogram.

The result is shown in the figure below. As one can discover, a large amount of movie has variance of ratings between 0 to 0.5. It can be cause by the fact that some movies only receives a small amount of ratings such as one or two ratings that make the total variance to be very low. As the variance increases, the amount of movies that receive a high variance get smaller.

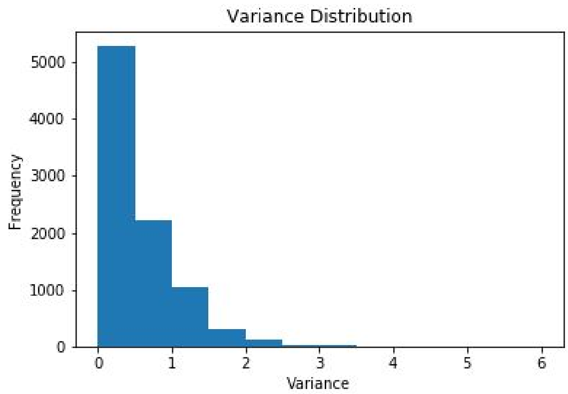


Figure 4: Plot for distribution of ratings variances

4. Neighbourhood-based collaborative filtering

Question 7: Write down the formula for μu in terms of Iu and ruk.

The formula for μ_u is:

$$\mu_u = \frac{\sum_{k=1}^{I_u} r_{uk}}{len(I_u)}$$

Question 8: In plain words, explain the meaning of Iu ∩ Iv. Can Iu ∩ Iv = Ø (Hint: Rating matrix R is sparse).

Iu ∩ Iv means the set of item dices which both user u and v have rated. Because the matrix R is sparse, it's possible that there's a set of item dices which neither user u nor user v have rated. So Iu ∩ Iv = Ø can be true.

Question 9: Can you explain the reason behind mean-centering the raw ratings (rvj - v) 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).

The reason behind mean-centering the raw ratings is to reduce the bias of the estimator. If a user rates too high or too low, the absolute value will significantly impact the prediction. Therefore, we need to mean-center the raw ratings in order to reduce the weights of extreme ratings. After mean-centering, the ratings can reflect the true rating result of the movie.

Question 10: Design a k-NN 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 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).

The result plots are shown below:

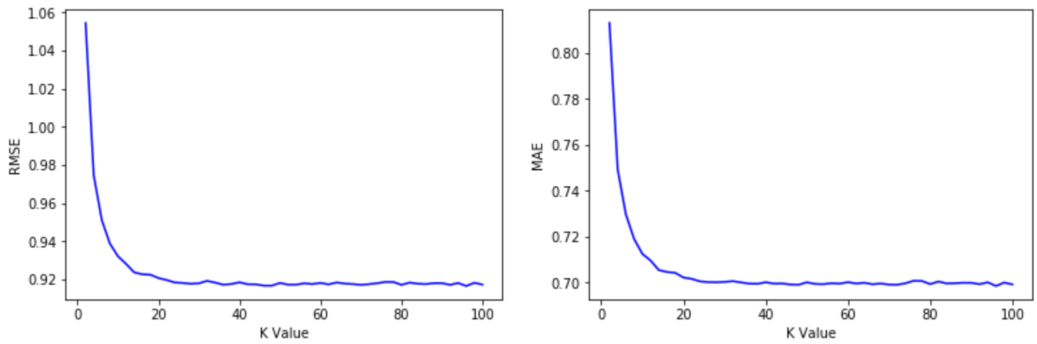


Figure 5: Plots for RMSE and MAE against K

Question 11: Use the plot from question 10, to nd 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

From the plot above, the minimum k for RMSE is k = 24, the minimum k for MAE is k = 16. The steady state value of average RMSE is 0.921191, and the steady state value of average MAE is 0.702794.

Question 12: Design a k-NN 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 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

The K value for minimum RMSE is 38, and the minimum RMSE is 0.872702. The result is shown below:

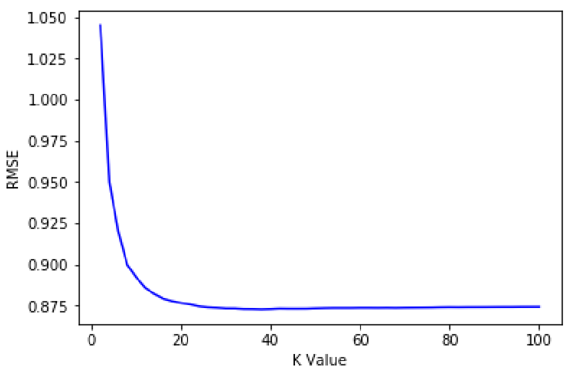


Figure 6: Plot for RMSE against K with popular trimmed testset

Question 13: Design a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of 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

The K value for minimum RMSE is 48, and the minimum RMSE is 0.998106. The result is shown below:

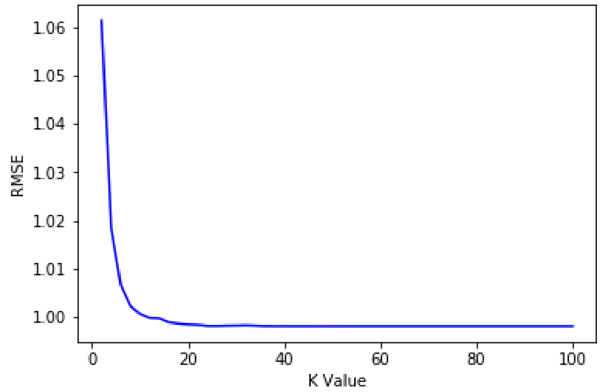


Figure 7: Plot for RMSE against K with unpopular trimmed testset

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

The K value for minimum RMSE is 20, and the minimum RMSE is 1.734602. The result is shown below:

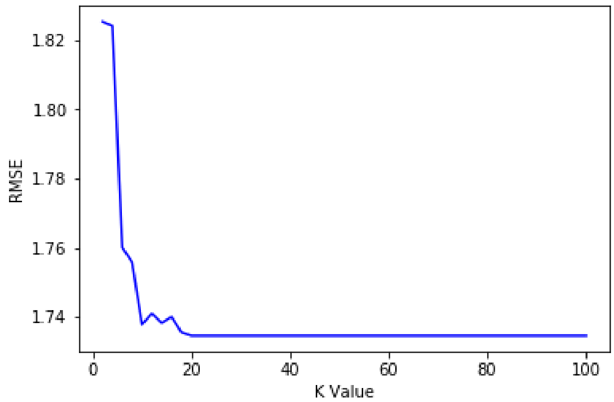


Figure 8: Plot for RMSE against K with high variance trimmed testset

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

The ROC curves and AUC values are shown below. We can discover that when the threshold is set to 4 the ROC curve has the highest AUC values which means the best performance. We can actually use 4 as the threshold to define whether a user like a movie or not.

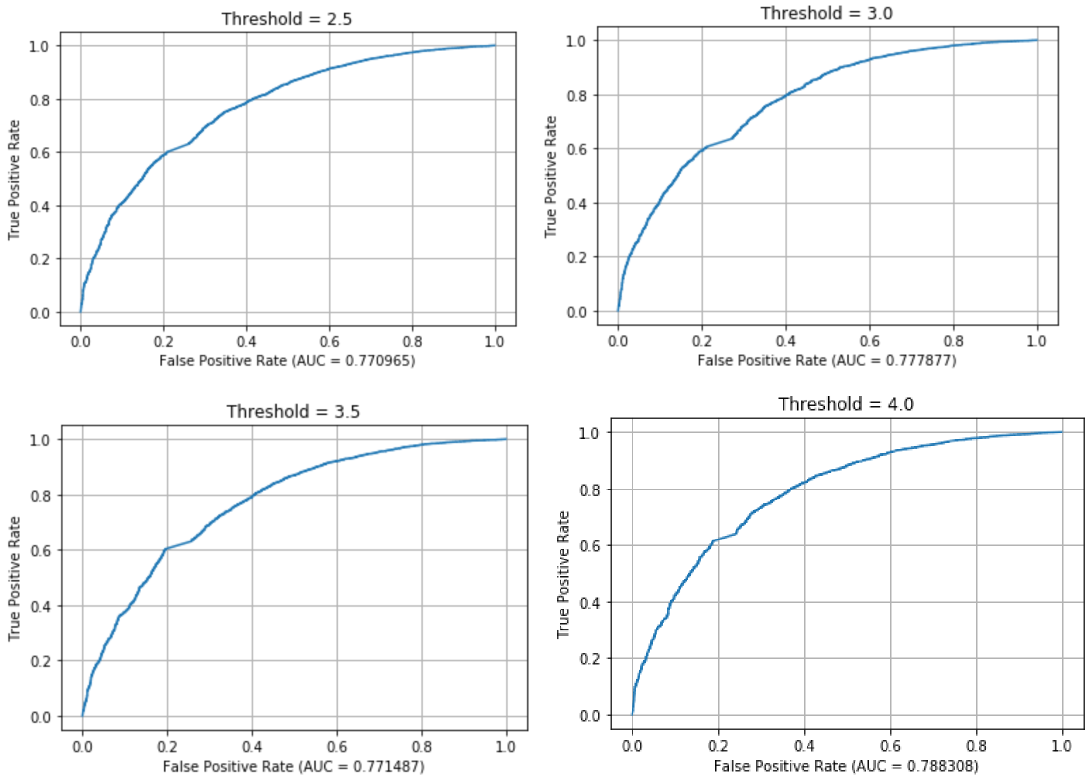


Figure 9: Plots for ROC curves using threshold 2.5, 3, 3.5 and 4

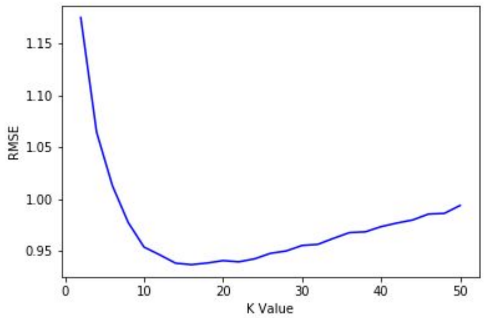
5. Model-based collaborative filtering

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.

Yes, the optimization problem is convex. We can take double derivative and check if the matrix is positive semidefinite to confirm the result.

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.

The results are shown below.



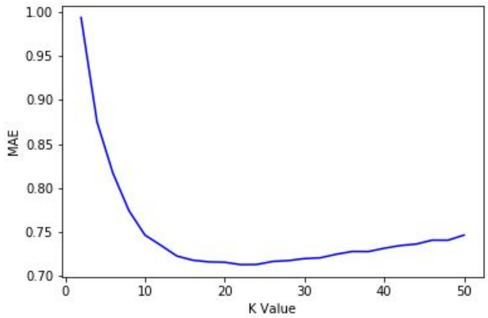


Figure 10: Plots for RMSE and MAE against K

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?

For RMSE, the optimal number of k is 16 and for MAE is 22. There are a total of 18 categories of movie in the data set and the result gerneated by RMSE is closer than that of MAE. The minimum of RMSE is 0.9378 and the minimum of MAE is 0.7122.

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.

In this part, the minimum RMSE is reached when the latent factor is 14. The minimum RMSE value is 0.8935

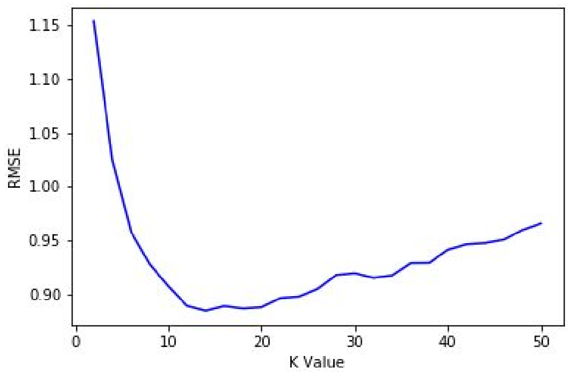


Figure 11: Plot for RMSE against K with popular trimmed testset

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. 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.

In this part, the minimum RMSE is reached when the latent factor is 26. The minimum RMSE value is 1.011

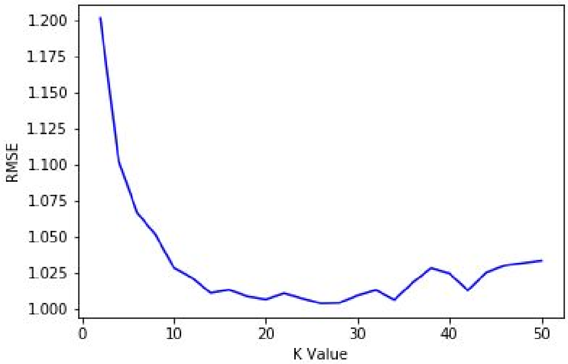


Figure 12: Plot for RMSE against K with unpopular trimmed testset

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.

In this part, the minimum RMSE is reached when the latent factor is 34. The minimum RMSE value is 1.364

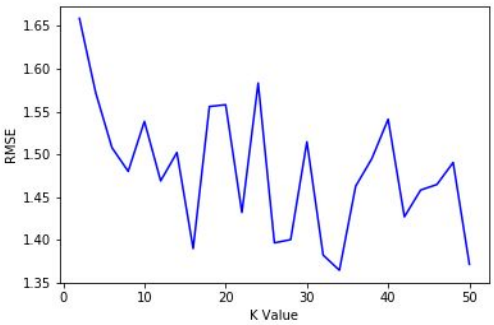


Figure 13: Plot for RMSE against K with high variance trimmed testset

Question 22: Plot the ROC curves for the NNMF-based collaborative filter designed in question 17 for threshold values [2.5, 3, 3,5, 4]. For the ROC plotting use the optimal number of latent factors found in question 18. For each of the plots, also report the area under the curve (AUC) value.

The ROC curves and AUC values are shown below. We can discover that when the threshold is set to 2.5, the ROC curve has the highest AUC values which means the best performance. We can actually use 2.5 as the threshold to define whether a user like a movie or not.

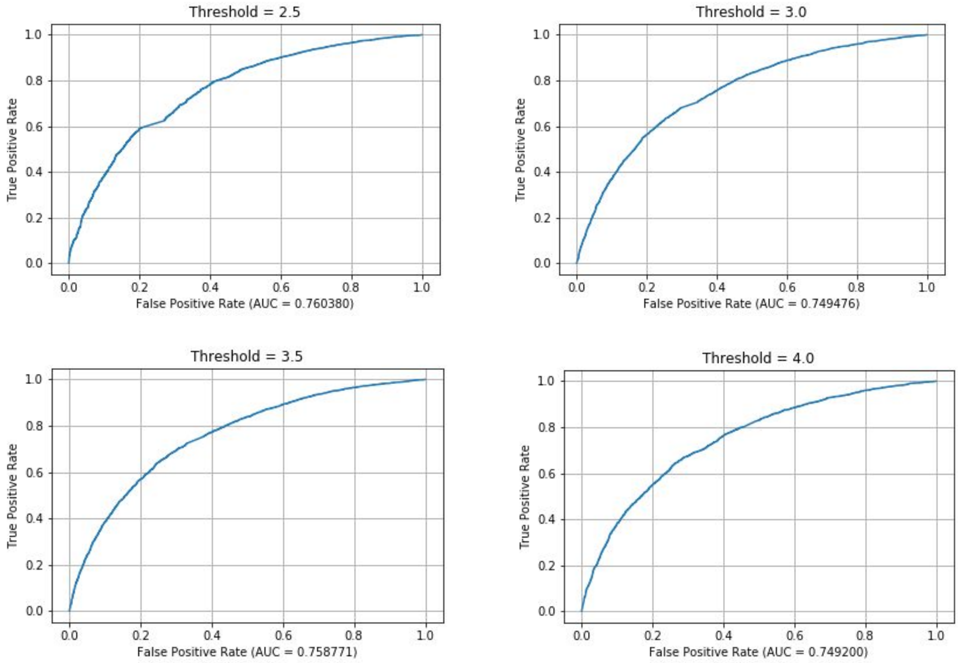


Figure 14: Plots for ROC curves using threshold 2.5, 3, 3.5 and 4

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). 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?

We extract three sample column of V. The genres of the top 10 movies in the three samples are listed as follows:

```
top 10 list (sample 1):
Drama
Drama
Adventure|Drama|Western
Action|Adventure|Thriller
Comedy|Drama|Romance
Drama|Romance|Thriller
Crime|Drama|Thriller
Action|Crime|Drama
Thriller
Crime|Film-Noir|Mystery|Thriller

top 10 list (sample 2):
Drama|Romance
Drama|Mystery
Film-Noir|Mystery
Drama
Action|Adventure|Mystery|Romance|
Thriller
Drama
Comedy|Drama|Romance
Drama
Comedy|Romance
Comedy|War

top 10 list (sample 3):
Action|Adventure|Comedy|Fantasy|Horror
Action|Adventure|Sci-Fi|Thriller
Action|Adventure|Comedy|Sci-Fi
Horror|Sci-Fi
Action|Adventure|Horror|Sci-Fi
Mystery|Sci-Fi|Thriller
Action|Sci-Fi|Thriller
Action|Crime|Drama|Sci-Fi|Thriller
Action|Adventure|Sci-Fi|Thriller
Action|Sci-Fi|Thriller
```

It is easy to notice that most of top 10 movies in the same column belong to a particular genre. In the sample 1, 7 of the top 10 movies are categorized into the genre of drama. In the sample 2, 6 of the top 10 movies are categorized into the genre of drama. In the sample 3, 8 of the top 10 movies are categorized into the genre of action. The non-negative matrix factorization is used to reduce the dimensionality of the dataset. A good dimension reduction method can map the original dataset to dimensions, where data points with high correlation can group together, and data points with low correlation are separately distributed. Therefore, in one of the dimensions of V, the movies with close scores must have somewhat similar properties, like genre. Thus, the observation of the top 10 movies is explainable.

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. 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.

For this part, we use the SVD function in surprise kit and set parameter biased to TRUE. The figure results are shown as following:

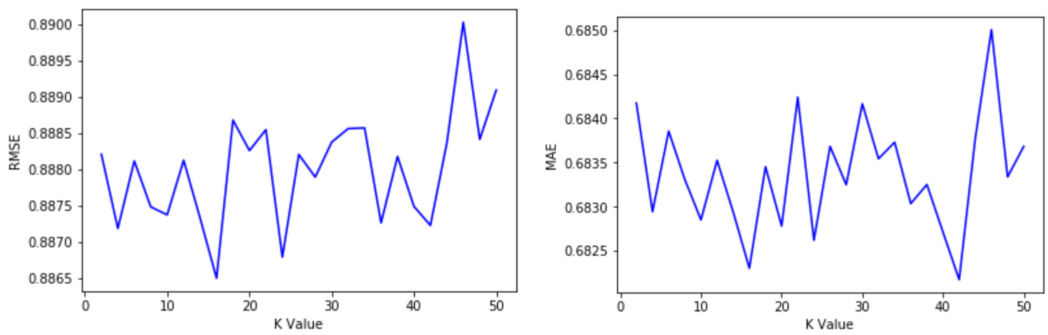


Figure 15: Plots for RMSE and MAE against K

Notice that for both of the plots, the variability is pretty small, especially compared to the NNMF. Also, the line seems go up and down randomly. This is probably because of that the MF with bias filter will learn the use and item bias. This contributes to a good performance even when k is pretty small. As a result, both of RMSE and MAE for MF with bias filter is smaller than the NNMF.

Question 25: <u>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.</u>

For RMSE, the optimal number of k is 16 and for MAE is 42. The minimum of RMSE is 0.8865 and the minimum of MAE is 0.6822. However, as we discussed above, for the MF with bias filter, the performance will not increase significantly if we increase the k value.

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

In this part, the minimum RMSE is reached when the latent factor is 10. The minimum RMSE value is 0.8634.

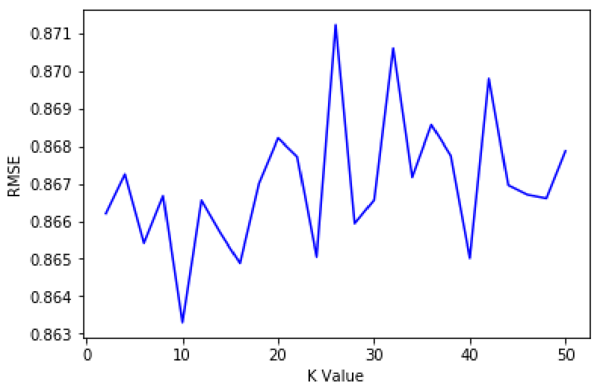


Figure 16: Plot for RMSE against K with popular trimmed testset

Question 27: <u>Design a MF with bias collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k</u>

compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE

In this part, the minimum RMSE is reached when the latent factor is 14. The minimum RMSE value is 0.9218.

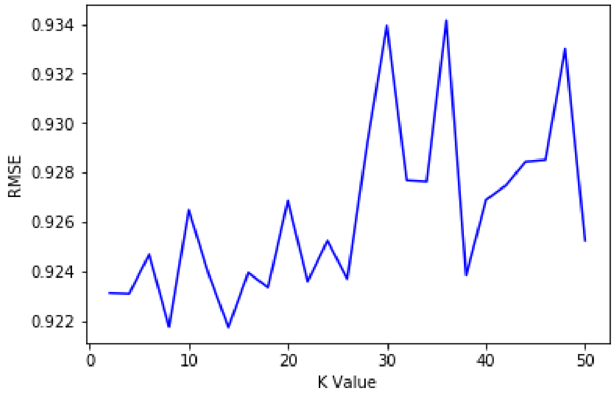


Figure 17: Plot for RMSE against K with unpopular trimmed testset

Question 28: Design a MF with bias collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE

In this part, the minimum RMSE is reached when the latent factor is 48. The minimum RMSE value is 1.504.

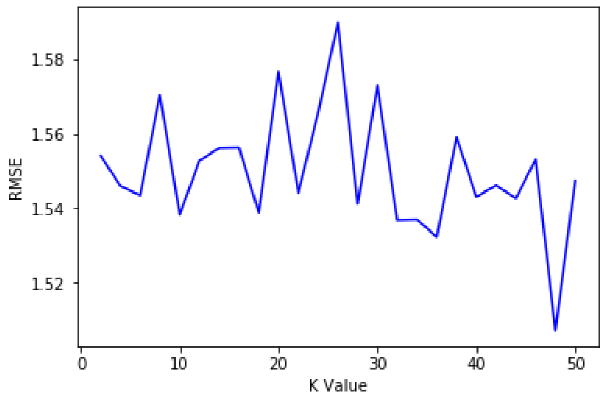


Figure 18: Plot for RMSE against K with high variance trimmed testset

As we can see here, the MF with bias filter gives a significantly worse result for this variance trimmed testset. This is probably due to the dataset itself. The movies in the this trimmed set is kind of controversial since different people gives different ratings. As a result, it's kind of difficult for the algorithm to predict the rating.

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.

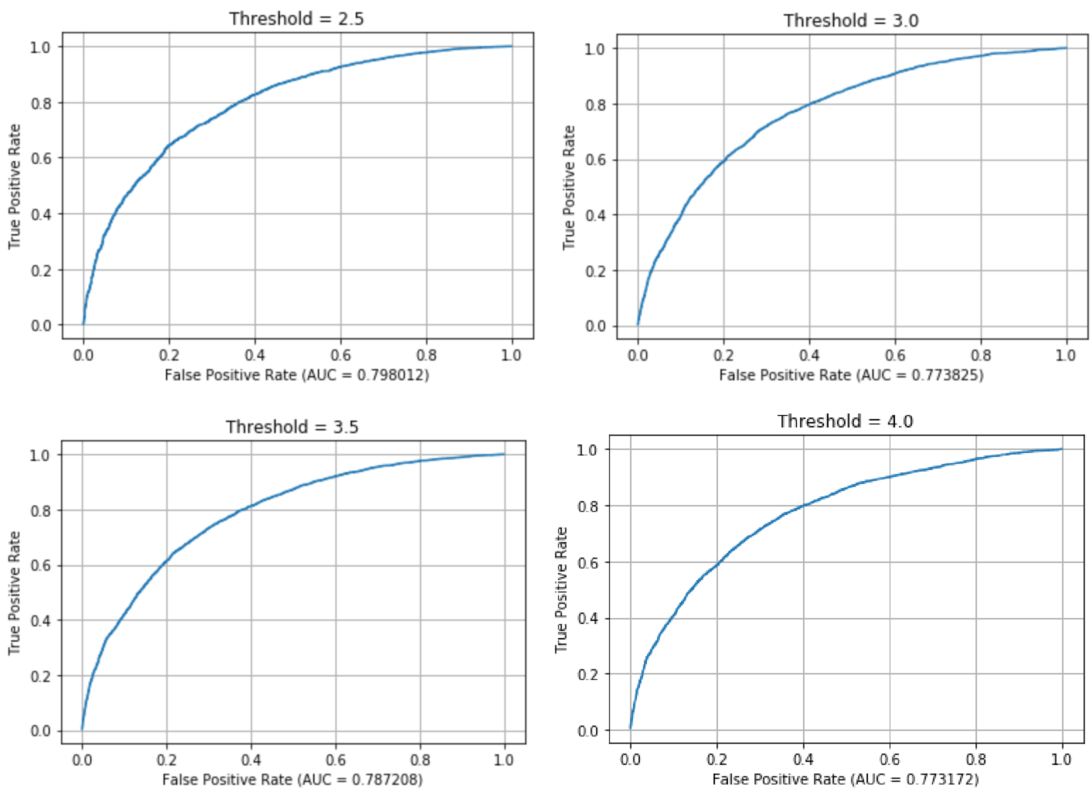


Figure 19: Plots for ROC curves using threshold 2.5, 3, 3.5 and 4

The ROC curves and AUC values are shown above. We can discover that when the threshold is set to 2.5, the ROC curve has the highest AUC values which means the best performance. We can actually use 2.5 as the threshold to define whether a user like a movie or not.

6. Naive collaborative filtering

For this part, we will implement a naive collaborative filter. Notice that one key different thing for the naive collaborative filter is that there is no actual training step. Another thing that is worth to mention is that during the prediction step, the prediction will use the entire dataset to compute the predicted function, including the test set itself. Thus, while dealing with the trimmed dataset, we can first use a data strimmer that finds all the trimmed data in the entire dataset we want and save them to a new csv file. This procedure will give an exactly same result when we use the traditional cross validation where we trim the test set only at each iteration.

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 is: 0.954963227285

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 is: 0.950837119126

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 is: 0.892263514189

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 is: 1.00482465229

For the naive collaborative filter, it gives similar results for all the different test sets. It uses a rather simple computation to compute the predicted results. This method is kind of conservative, which will give a neither very good nor too bad performance.

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.

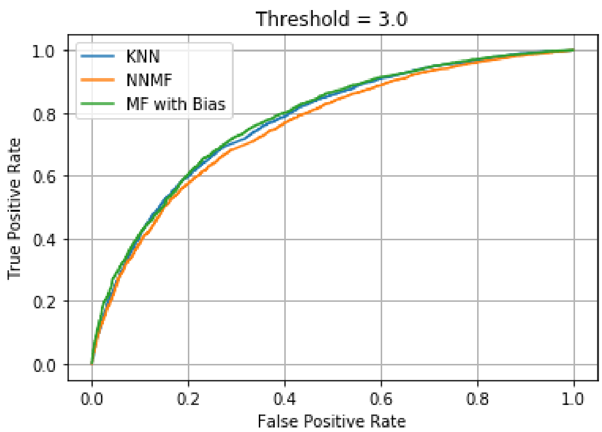


Figure 20: Comparison of the three filters

We plot these three filters in the same figure. We use the optimal k values for each filter so that they tend to give their best performance respectively. As we can see in the figure, the MF with Bias gives a slightly better performance than k-NN, while NNMF gives the worst performance. For this dataset and this recommendation task, k-NN is proved to be the best model among these three models by experiment.

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.

The equations of precision and recall are given as follows:

$$Precision(t) = \frac{|S(t) \cap G|}{|S(t)|}$$

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

S(t), here, represents the set of items of size t recommended to the user. G represents the set of rated items liked by the user. So, $|S(t) \cap G|$ represents the number of items which is recommended to the user as well as known to be liked by the user. In other words, $|S(t) \cap G|$ stands for number of true positive. $|S(t)|$ represents number of predicted condition positive, while $|G|$ represents number of condition positive.
The precision is positive predictive value, which is number of true positive over number of predicted condition positive. In the case, it is the number of recommended movies, which are also liked by users over the number of recommended movies.
The recall is true positive rate, which is number of true positive over number of condition positive. In the case, it is the number of recommended movies, which are also liked by users, over the number of movies liked by users.

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.

The k found in question 11 is 24. This means we use the 24-NN collaborative filter to predict the ratings of movies in test sets. The plot of average precision over t, the plot of average recall over t, and the plot of average precision over average recall are shown in Figure 21. From the plot of average precision against t, we can see that the average precision drops just a little, as t gets larger. The plot of average recall against t shows that the average recall decreasingly increases as t becomes larger. The average recall tends to be stabilized in a small range around 0.65. In the precision-recall plot, the curve is tilted downwards at a small angle. This means that using top-t ranking, where t varies from 1 to 25, in the recommendation problem, can always result in high precision and t-proportional recall. High precision means that most of recommended movies by the system would be liked by the users.

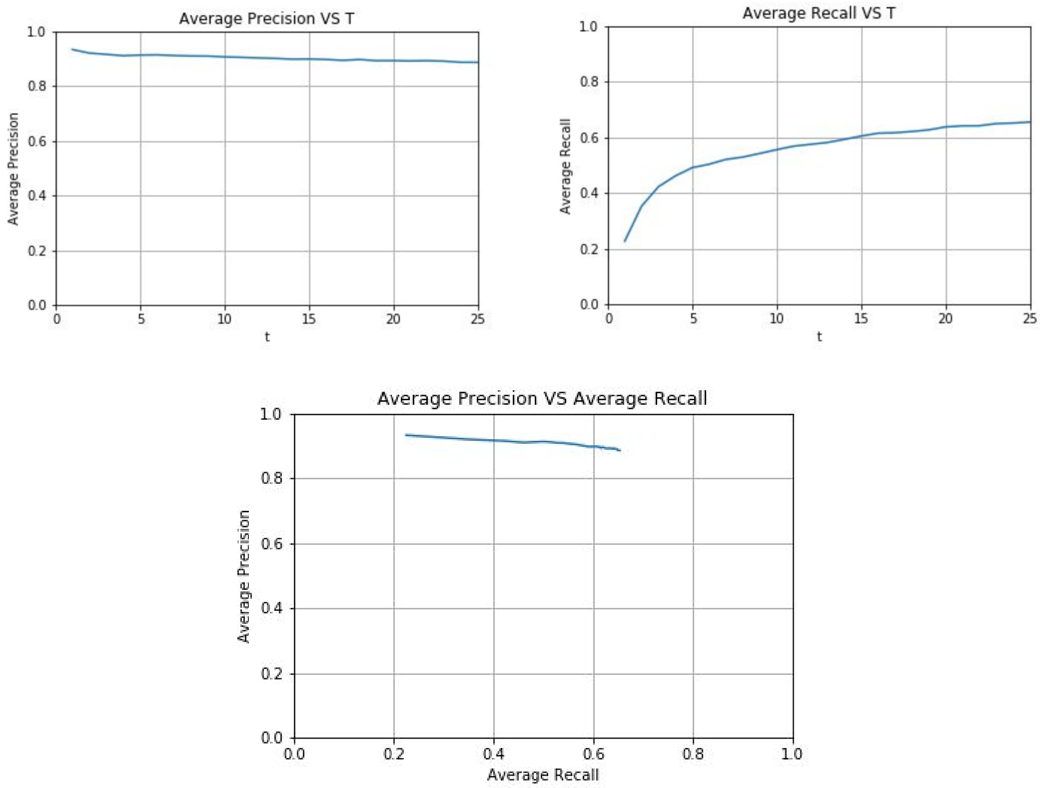


Figure 21: Plots for average precision against t, average recall against t, and average precision against average recall.(k-NN collaborative filter)

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.

The optimal number of latent factor found in question 11 is 18. The plot of average precision over t, the plot of average recall over t, and the plot of average precision over average recall are shown in Figure 22. From the plot of average precision against t, we can see that the average precision drops just a little, as t gets larger. The plot of average recall against t shows that the average recall decreasingly increases as t becomes larger. The average recall tends to be stabilized in a small range around 0.65. In the precision-recall plot, the curve is tilted downwards at a small angle. This means that using top-t ranking, where t varies from 1 to 25, in the recommendation problem, can always result in high precision and t-proportional recall. High precision means that the users would like most of recommended movies by the system. It is totally the same case as that for k-NN filter.

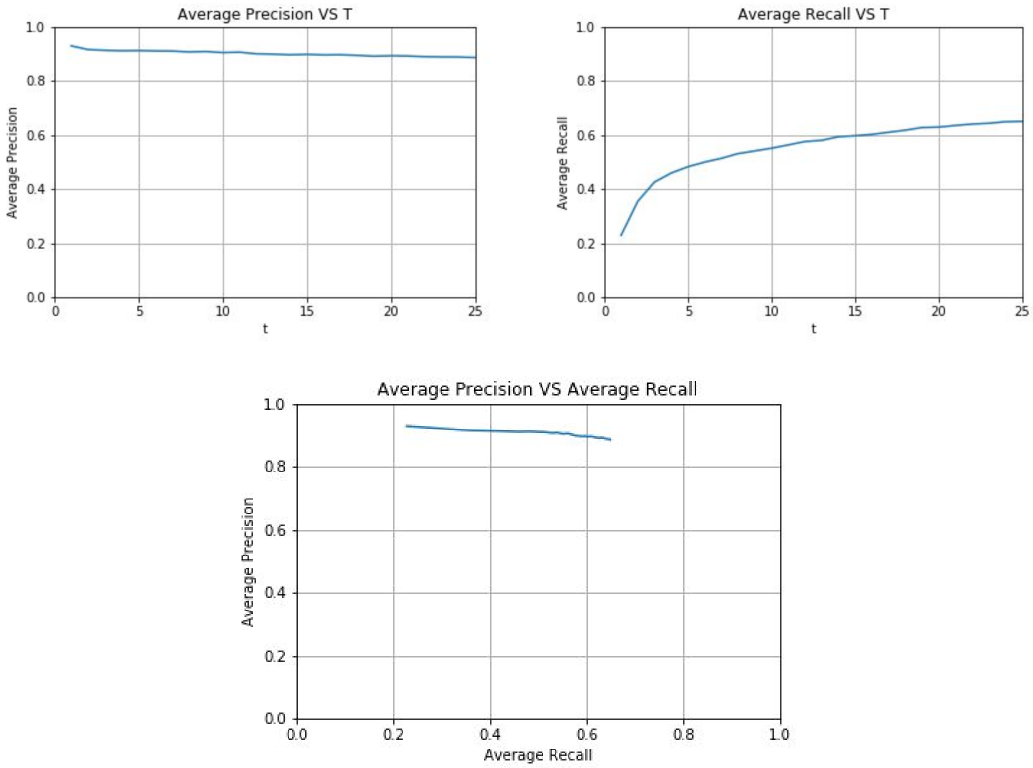


Figure 22: Plots for average precision against t, average recall against t, and average precision against average recall.(NNMF-based collaborative filter)

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.

The optimal number of latent factor found in question 11 is 16. The plot of average precision over t, the plot of average recall over t, and the plot of average precision over average recall are shown in Figure 23. From the plot of average precision against t, we can see that the average precision drops just a little, as t gets larger. The plot of average recall against t shows that the average recall decreasingly increases as t becomes larger. The average recall tends to be stabilized in a small range around 0.65. In the precision-recall plot, the curve is tilted downwards at a small angle. This means that using top-t ranking, where t varies from 1 to 25, in the recommendation problem, can always result in high precision and t-proportional recall. High precision means that most of recommended movies by the system would be liked by the users. It is totally the same case as those for both k-NN filter and NNMF filter.

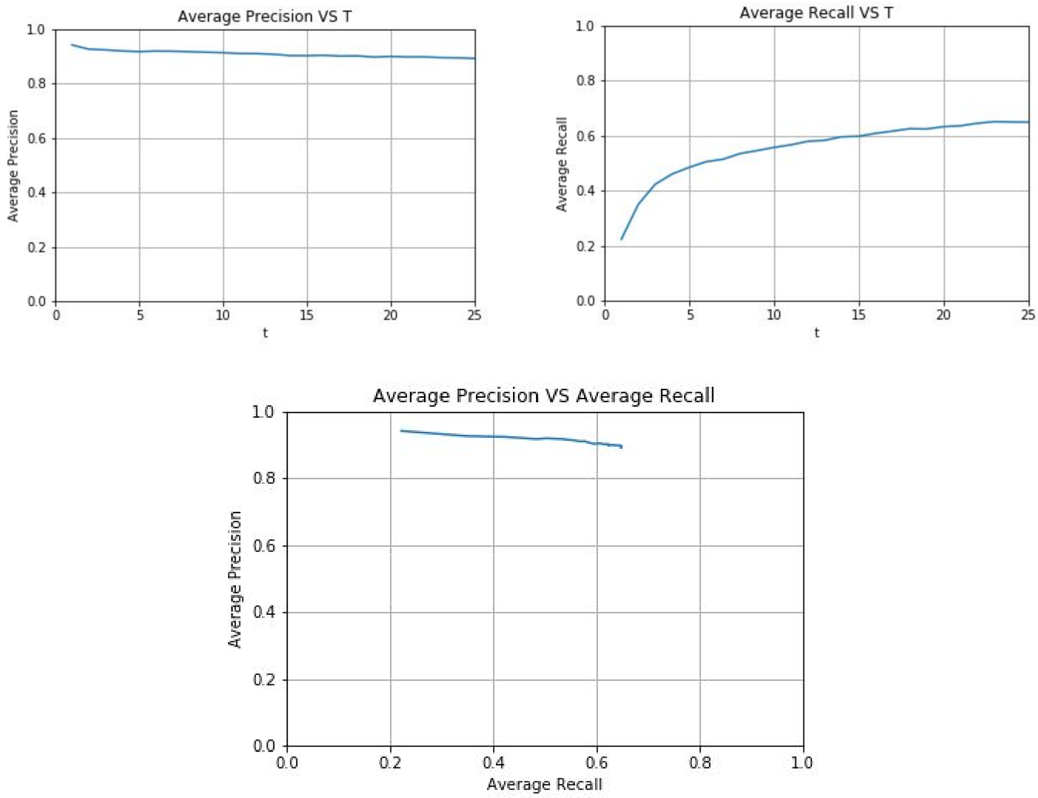


Figure 23: Plots for average precision against t, average recall against t, and average precision against average recall.(MF with bias-based collaborative filter)

Question 39: <u>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.</u>

We add the other two curves in question 36 and 37 to the plot in question 38. The result is shown in Figure 24. It shows that three curves are almost overlapping with one another. This means the relevance of recommendation list generated by either using k-NN, NNMF, or MF with bias is nearly equivalent. In other words, it doesn't matter which type of filter is used in recommendation system. In conclusion, three types of filters in ranking can generate predictions which in general have high precision but insufficient recall. The lists of recommended movies would be very likely to be accepted by the users, although the list is not likely to cover all favorite movies of the users. After all, the recommended movies are unlikely to be annoying to the users.

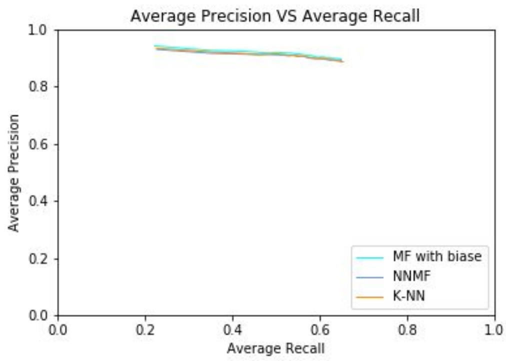


Figure 24: Plot of average precision against average recall for three different collaborative filters.