EE 219 Project 3

Yifan Shu, Chengshun Zhang, Xuan Yang

Introduction

The basic idea of recommender systems is to utilize user data to infer customer interests. 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.

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

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

Models using first type of data are referred to as collaborative filtering methods, whereas models that use second type of data are referred to as content based methods. In our project, we will build recommendation system using collaborative filtering methods.

One main challenge in designing collaborative filtering method is that the underlying rating matrix is sparse. So the basic idea is that these unspecified ratings can be imputed because the observed ratings are often highly correlated across various users and items. Most of the collaborative filtering methods focus on either inter-item correlation or inter-user correlation for the prediction process.

In this project, we implemented and analyzed the performance of two types of collaborative filtering methods:

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

We build a recommendation system to predict the ratings of the movies in the MovieLens dataset. For the subsequent discussion, we assume that the ratings matrix is denoted by R, and it is an $m \times n$ matrix containing m users (rows) and n movies (columns). The (i, j) entry of the matrix is the rating of user i for movie j and is denoted by r_{ij} .

Question 1

By the definition of Sparsity Matrix

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

We calculated it with the following result.

$$Sparsity = 0.9864135824507121$$

Question 2

To plot the histogram showing the frequency of the rating values, we binned the rating values into intervals of width 0.5 and used the binned values as the horizontal axis. We then counted the number of entries in the ratings matrix R with rating values in the intervals and use this count as the vertical axis. The result is shown in Fig 1.

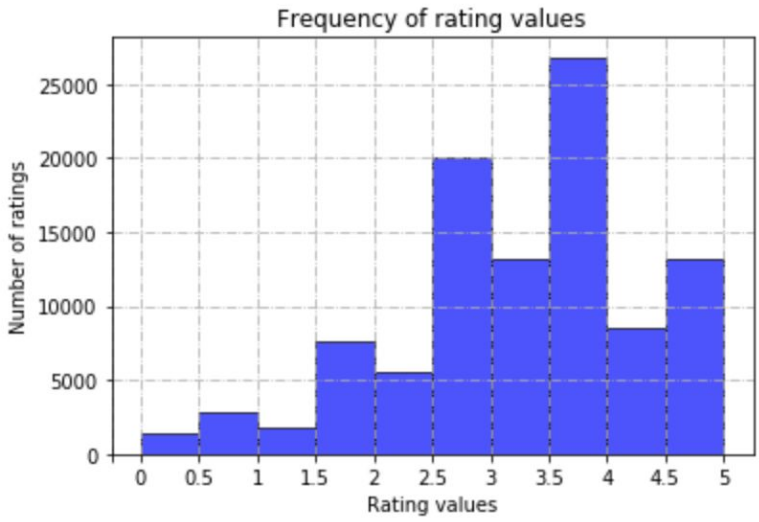

Figure 1. Frequency of Rating Values in Each Interval

It is shown that the rating values in each interval are not distributed evenly. Ratings between 2.5 and 4 account for more than half of total ratings.

Question 3

To plot the distribution of the number of ratings received each movie, we first ordered the movies by decreasing frequency, i.e., the movie with largest number of ratings has index 1, and used the index as X-axis. The Y-axis is simply the number of ratings for each movie. The monotonically decreasing curve is shown in Fig 2.

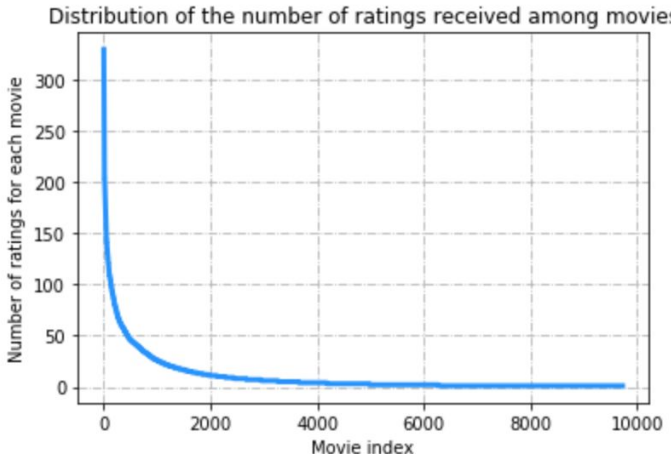

Figure 2. Number of Ratings for Each Movie

Question 4

We plotted the distribution of ratings among users, with the X-axis being the user index ordered by decreasing frequency and the Y-axis being the number of movies the user have rated. The monotonically decreasing curve is shown in Fig 3.

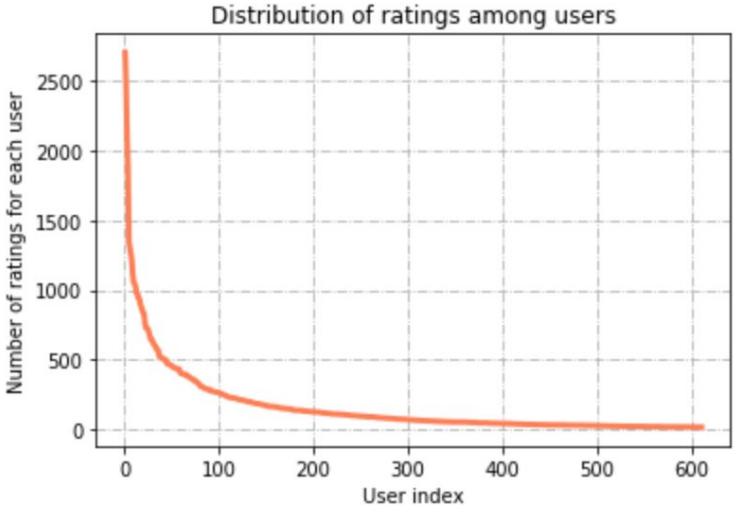

Figure 3. Number of Ratings By Each User

Question 5

From Fig 3, we can see that only an extremely small fraction of movies have received enough ratings (larger than 100) and a large number of movies hardly have any ratings. As a result, the recommendation system will always recommend a subset of movies of those that have received many ratings. The majority of movies in the dataset will just be ignored as they almost receive no ratings.

Question 6

We computed the variance of the rating values received by each movie and binned the variance values into intervals of width 0.5 and used the binned variance values as the horizontal axis. We counted the number of movies with variance values in each binned interval and used this count as the vertical axis. The result is shown in Fig 4.

It is shown in Fig 4 that more than half of the movies receive consistent ratings, i.e., the difference of their ratings by different users are within 0.5. Almost no movies receive ratings that differ more than 2.5.

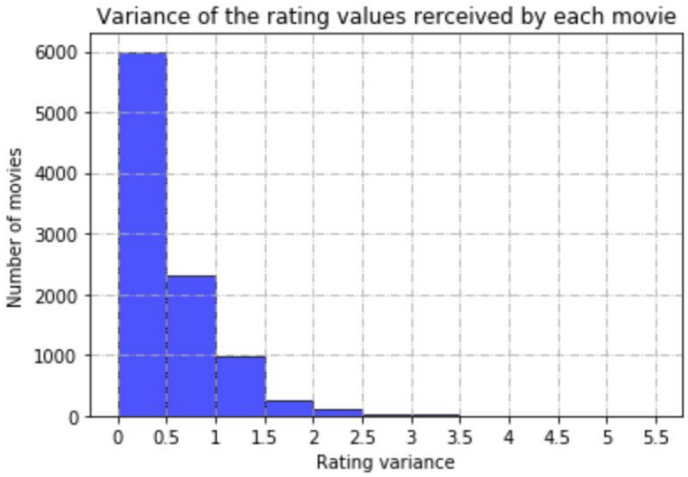

Figure 4. Number of Movies in Each Variance Interval

Question 7

The notations to be used are defined as below.

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.

From the definition, we wrote the formula for μ_u in terms of I_u and r_{uk} as following.

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

Question 8

$I_u \cap I_v$ means the set of item indices, movie indices in our project, that both user u and user v have specified the ratings. $I_u \cap I_v$ can be empty because our ratings matrix is sparse, it is highly likely that there are no movies get rated by both user u and user v.

Question 9

Since each user may rate differently, some users may rate things universally higher while other users may rate things universally lower. For example, rating as 4 may mean not good enough for some users while for other users, it may be the highest rating they've ever rated. So in the light of this, we need $(r_{vj} - \mu_v)$ to balance the ratings.

Question 10

We designed a k-NN collaborative filter to predict the ratings of the movies in MovieLens dataset and evaluated its performance using 10-fold cross validation. We used Pearson-correlation function as the similarity metric and swept k (number of neighbors) from 2 to 100 in step size of 2. For each k we computed the average RMSE and average MAE by averaging the RMSE and MAE across all 10 folds. At last we plotted average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis) as shown in Fig 5 and Fig 6 respectively.

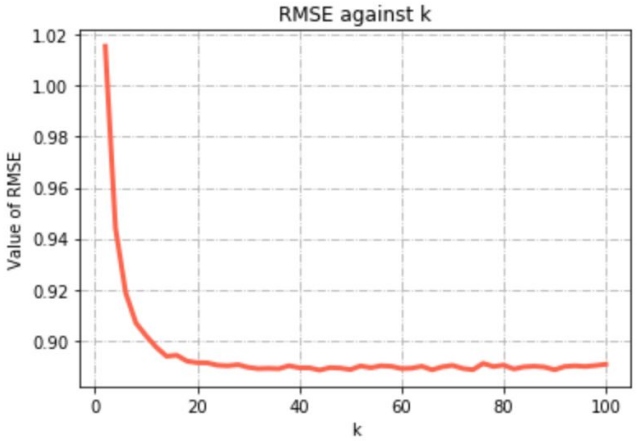

Figure 5. RMSE Against k from k-NN filter

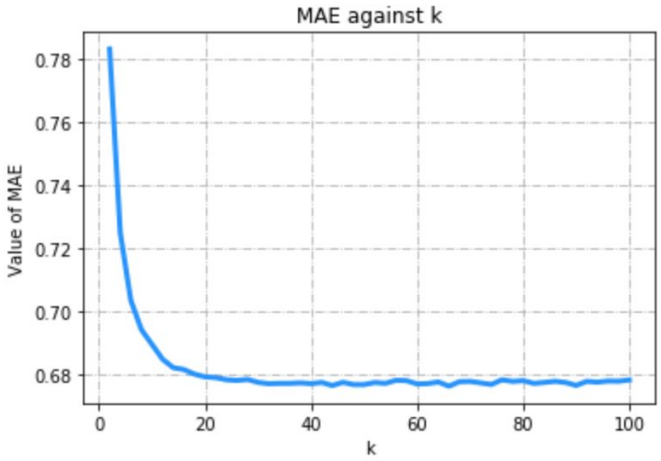

Figure 6. MAE against k from k-NN filter

Question 11

To find minimum k, we zoomed into the Fig 5 and Fig 6 as shown in Fig 7 and Fig 8. It shows that average RMSE and MAE converges to a steady value at $k = 20$. Thus the minimum k and the corresponding values of average RMSE and MAE are:

$$k = 20$$

$$RMSE = 0.8960893680190407$$

$$MAE = 0.683257956536476$$

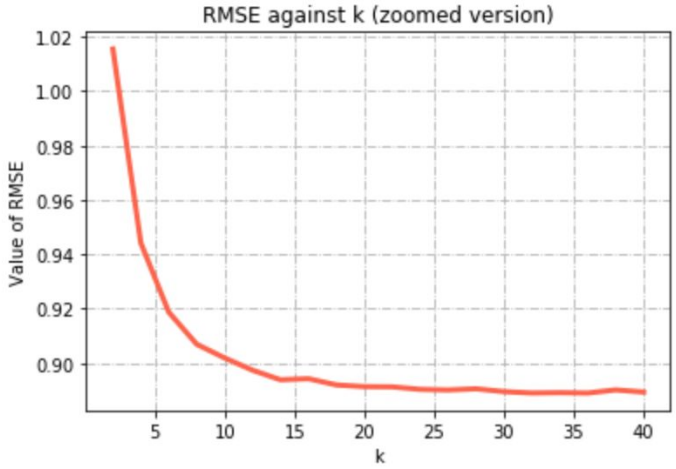

Figure 7. Zoomed in of RMSE Against k

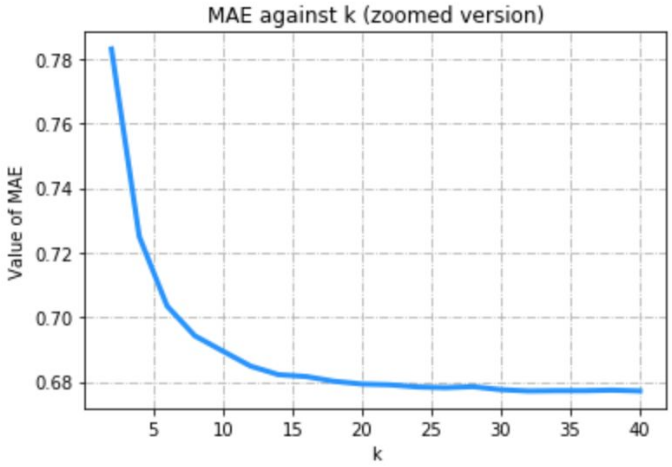

Figure 8. Zoomed in of MAE Against k

Question 12

We designed a k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluated its performance using 10-fold cross validation. We swept k (number of neighbors)

from 2 to 100 in step size of 2. For each k we computed the average RMSE by averaging the RMSE across all 10 folds. At last we plotted average RMSE (Y-axis) against k (X-axis) as shown in Fig 9. The minimum average RMSE is

$$RMSE = 0.8548322418127625$$

The detailed process is shown below:

- For each value of k, split the dataset into 10 pairs of training and test sets (trainset 1, testset 1), (trainset 2, testset 2), …, (trainset 10, testset 10).
- For each pair of (trainset, testset):
  § Train the collaborative filter on the trainset
  § Call the trimming function written ahead that takes as input the test set and outputs a trimmed test set
  § Predict the ratings of movies in the trimmed test set using the trained collaborative filter
  § Compute the RMSE of the predictions in the trimmed test set
- Compute the average RMSE by averaging across all 10 folds

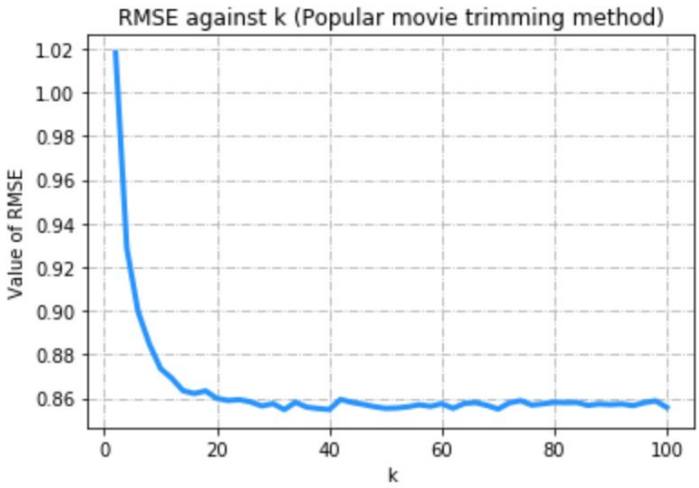

Figure 9. RMSE Against k with Popular Movie Trimming

Question 13

We designed a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluated its performance using 10-fold cross validation. We swept k (number of neighbors) from 2 to 100 in step size of 2. For each k we computed the average RMSE by averaging the RMSE across all 10 folds. The detailed process is the same as that in Question 12. At last we plotted average RMSE (Y-axis) against k (X-axis) as shown in Fig 10. The minimum average RMSE is

$$RMSE = 0.951260104160724$$

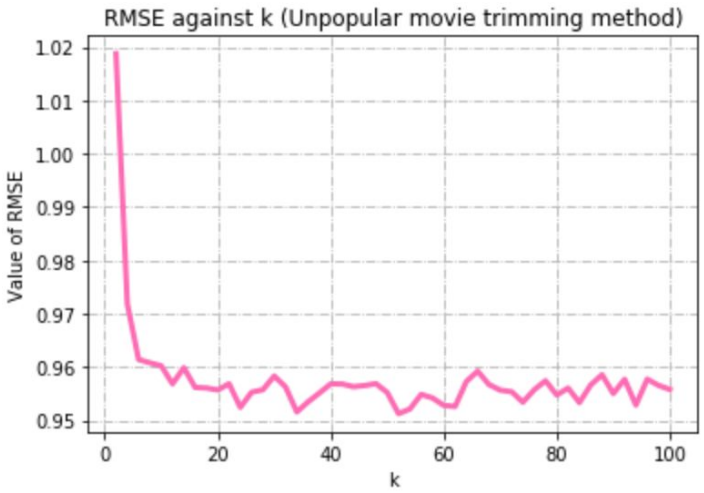

Figure 10. RMSE Against k with Unpopular Movie Trimming

Question 14

We designed a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluated its performance using 10-fold cross validation. We swept k (number of neighbors) from 2 to 100 in step size of 2. For each k we computed the average RMSE by averaging the RMSE across all 10 folds. The detailed process is the same as that in Question 12. At last we plotted average RMSE (Y-axis) against k (X-axis) as shown in Fig 11. The minimum average RMSE is

$$RMSE = 0.7683069443497439$$

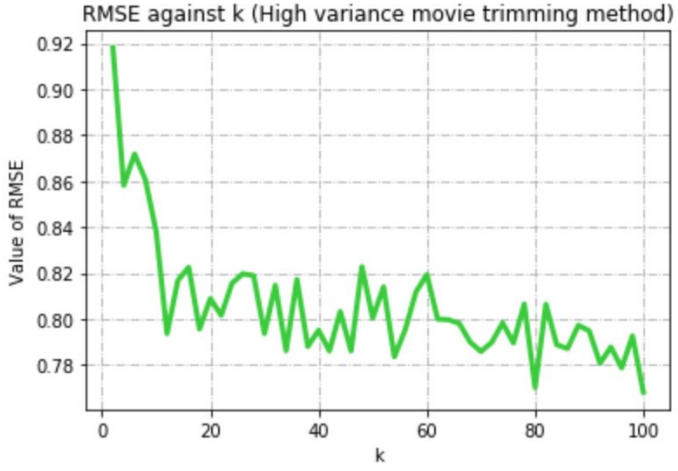

Figure 11. RMSE Against k with High Variance Movie Trimming

Question 15

We plot the ROC curve for k-NN collaborative filter designed in Question 10 for threshold values [2.5, 3, 3.5, 4]. For the plotting, we split the dataset into 90% for training and 10% for testing. We used the k found in Question 11, i.e., $k = 20$. Here we assumed that if the user likes the movie, i.e., the rating is

larger than threshold, the label is 1. And if the value of the rating is equal to the threshold, we assumed that the user still likes the movie, in which the label is also 1. And so as all the following parts. The results are shown in Fig 12, 13, 14, 15 respectively. For each threshold, the area under the curve (AUC) is:

$$AUC(threshold = 2.5) = 0.7839$$

$$AUC(threshold = 3.0) = 0.7842$$

$$AUC(threshold = 3.5) = 0.7788$$

$$AUC(threshold = 4,0) = 0.7723$$

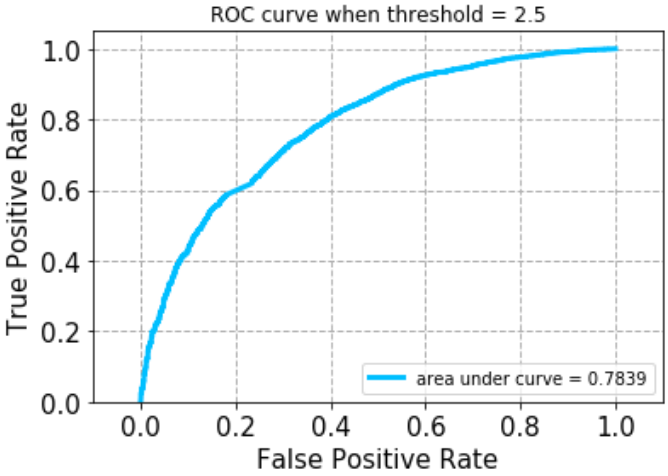

Fig 12. ROC Curve with Threshold 2.5

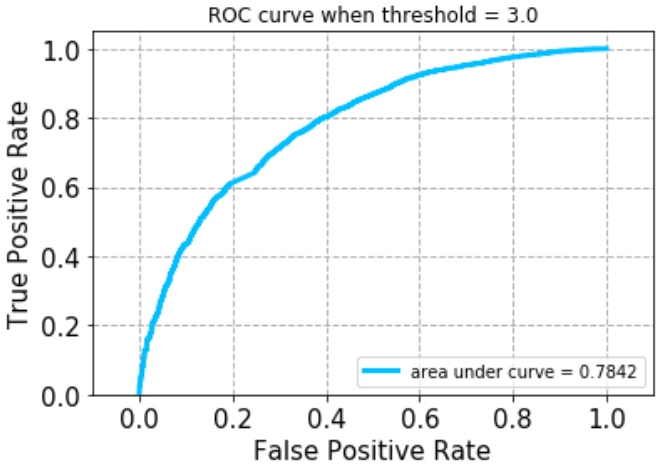

Fig 13. ROC Curve with Threshold 3.0

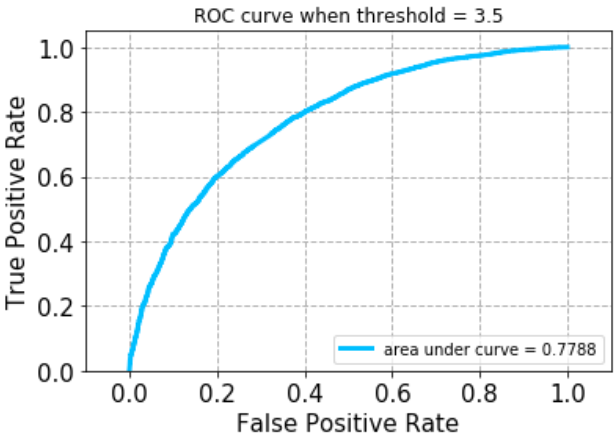

Fig 14. ROC Curve with Threshold 3.5

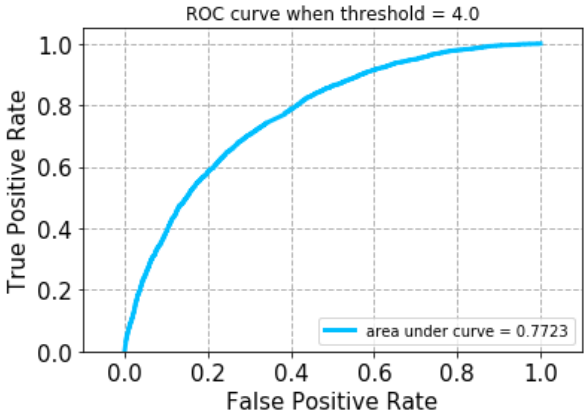

Fig 15. ROC Curve with Threshold 4.0

Question 16

The optimization problem specified by the formula is convex.

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

With U fixed and for a fixed i, it is

$$minimize_V \sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2$$

$$= minimize_V W_i(R_i - U_i V_i^T)^2$$

Where W_i is the ith row of W, R_i is the ith row of R, U_i and V_i are ith row of U and V respectively And this is a least square problem, which is a convex optimization problem. The original problem is just the addition over i of these least square problems, thus, is also convex.

Question 17

In this part, we used Non-negative matrix factorization (NNMF) based collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated its performance using 10-fold cross

validation. And k (number of latent factors) was swept from 2 to 50 in step sizes of 2. For each k, we also computed the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. The following figures are average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k respectively.

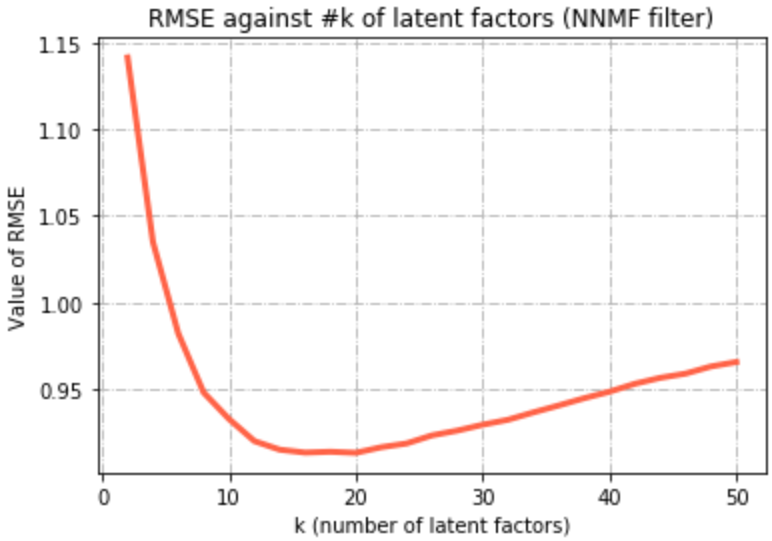

Figure 16. Average RMSE of NNMF filter from 10-fold cross-validation against k

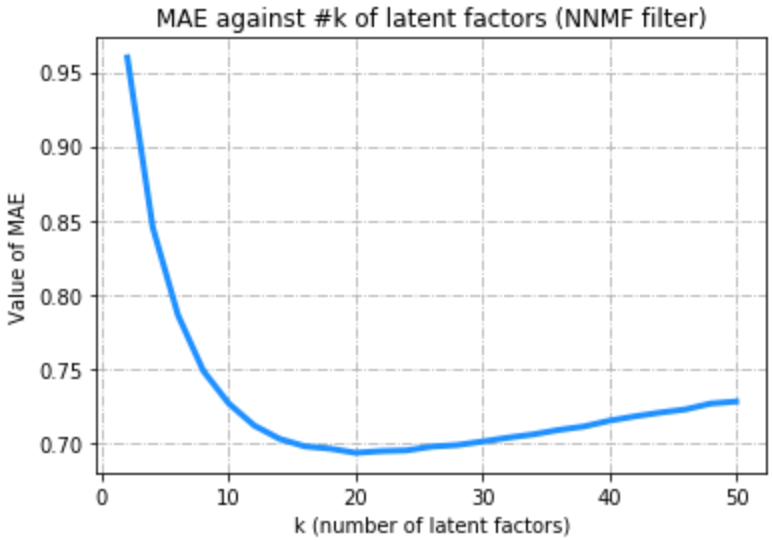

Figure 17. Average MAE of NNMF filter from 10-fold cross-validation against k

Question 18

As we could see from question 17, we could found out the optimal number of latent factors, which attains the minimum MAE or RMSE value.

	RMSE	MAE
Optimal k	20	20
Minimum Average	0.9131984342180456	0.6932127173998259

Table 1. The optimal number of latent factors and minimum average value from RMSE and MAE using NNMF

For the optimal number of latent factor from RMSE is 20, which is the same as the number of genres 20. And the optimal number of latent factor from MAE is 20, which is also the same number of genres 20.

Question 19

In this part, we predicted the ratings of the movies using NNMF collaborative filter on popular movie trimmed test set. Then its performance was evaluated using 10-fold cross validation. The number of latent factors k was swept from 2 to 50 in step size of 2, and for each k the average RMSE was computed. The figure below shows the result of average RMSE (Y-axis) against k (X-axis).

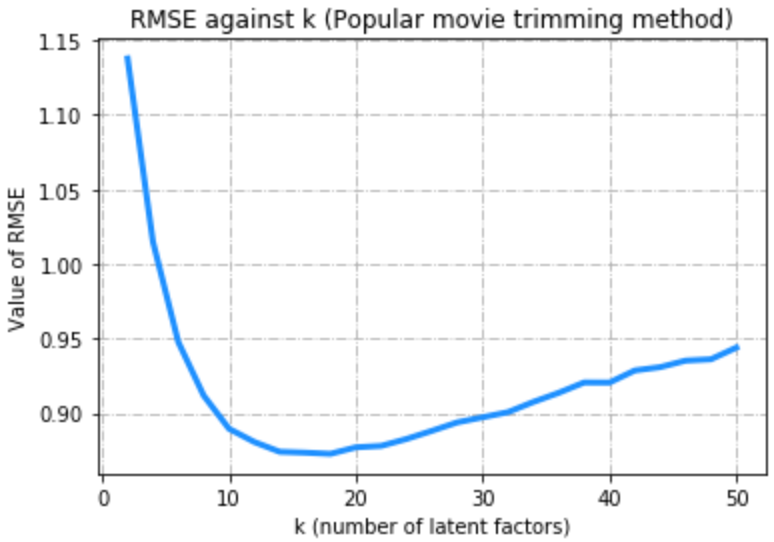

Figure 18. Average RMSE of NNMF filter from 10-fold cross-validation against k on popular movie trimmed test set

And the minimum average RMSE value is 0.8694672637780329.

Question 20

In this part, we predicted the ratings of the movies using NNMF collaborative filter on unpopular movie trimmed test set. Then its performance was evaluated using 10-fold cross validation. The number of latent factors k was swept from 2 to 50 in step size of 2, and for each k the average RMSE was computed. The figure below shows the result of average RMSE (Y-axis) against k (X-axis).

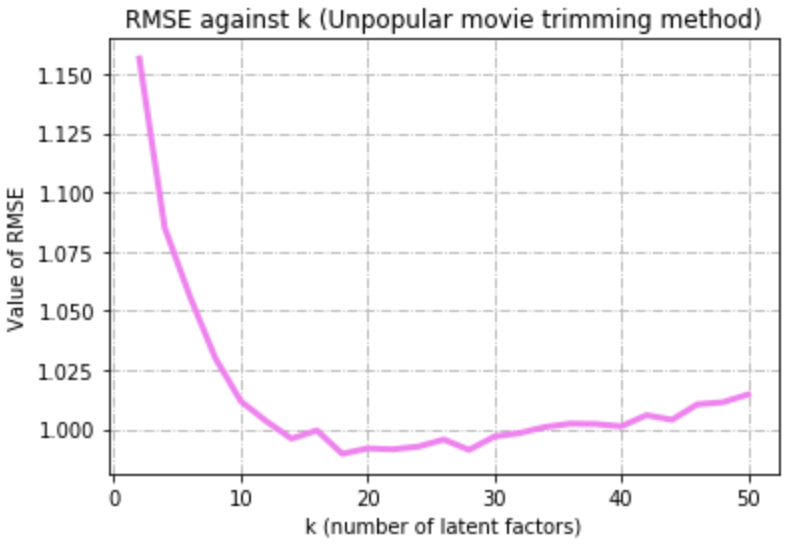

Figure 19. Average RMSE of NNMF filter from 10-fold cross-validation against k on unpopular movie trimmed test set

And the minimum average value of RMSE is 0.9881696966401068.

Question 21

In this part, we predicted the ratings of the movies using NNMF collaborative filter on high variance movie trimmed test set. Then its performance was evaluated using 10-fold cross validation. The number of latent factors k was swept from 2 to 50 in step size of 2, and for each k the average RMSE was computed. The figure below shows the result of average RMSE (Y-axis) against k (X-axis).

Figure 20. Average RMSE of NNMF filter from 10-fold cross-validation against k on high variance movie trimmed test set

And the minimum average value of RMSE is 0.7585844024224345.

Question 22

In this part, the ROC curve was plotted using NNMF collaborative filter designed in question 17 for threshold in [2.5, 3, 3.5, 4]. And we used the optimal number of latent factors found in question 18, which is 20. And for each of the plots, the area under the curve (AUG) value was also reported in the plot.

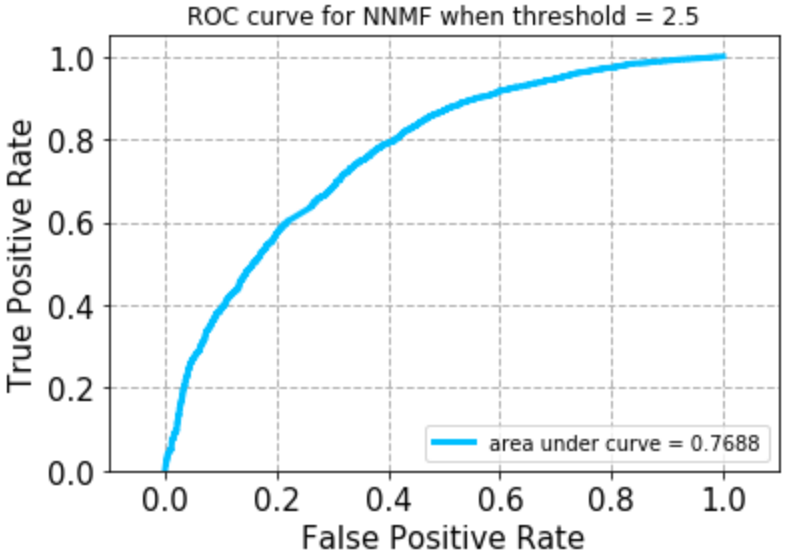

Figure 21. ROC curve for NNMF filter when threshold = 2.5

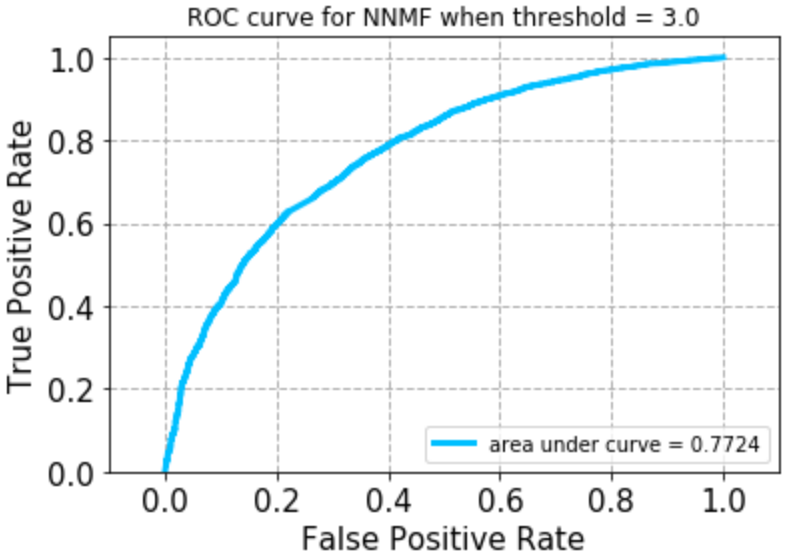

Figure 22. ROC curve for NNMF filter when threshold = 3.0

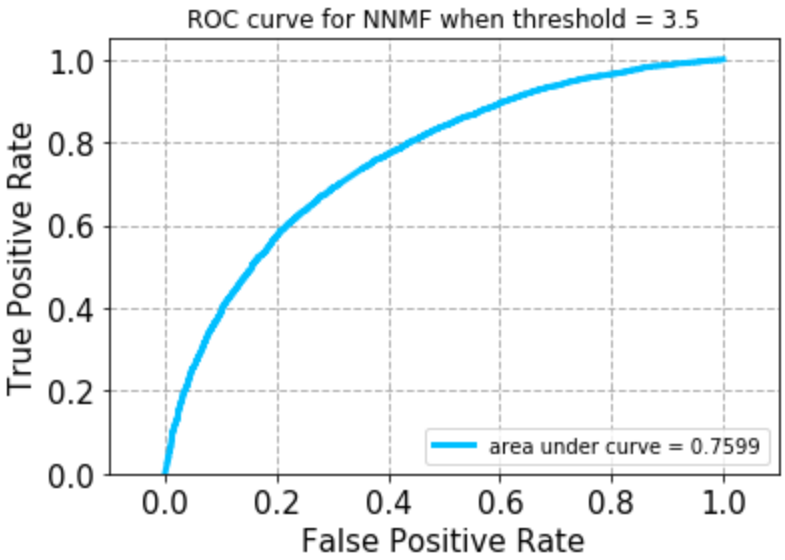

Figure 23. ROC curve for NNMF filter when threshold = 3.5

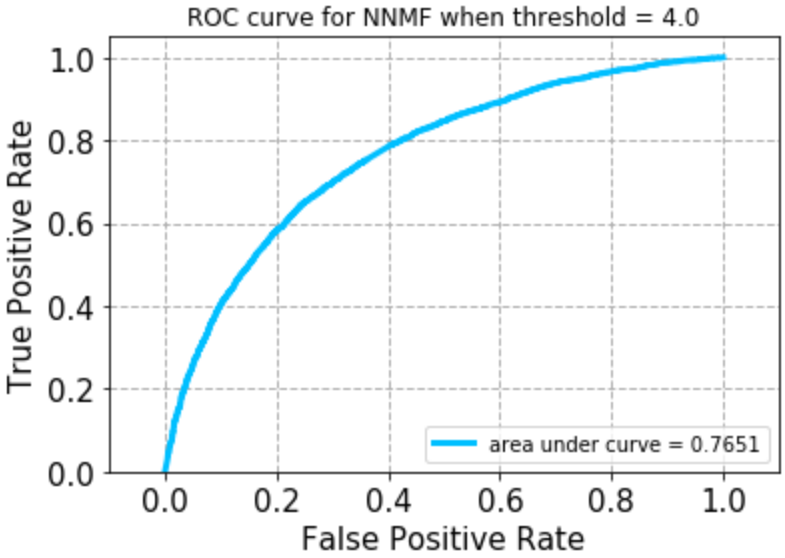

Figure 24. ROC curve for NNMF filter when threshold = 4.0

Question 23

In this part, the Non-negative matrix factorization on the ratings matrix R was performed 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, the top 10 movies in each genre are shown as below:

	Top 100 movies' ID in each genre

Genre 1	33834, 1999, 73042, 89904, 61350, 1966, 74754, 26249, 171, 77688
Genre 2	213, 70994, 7564, 100714, 95182, 7116, 3682, 179819, 49932, 1014
Genre 3	1194, 1468, 69644, 3990, 619, 68073, 121097, 1483, 67695, 104863
Genre 4	3837, 4678, 7842, 430, 4863, 4850, 104879, 3018, 4233, 3841
Genre 5	6140, 7650, 8035, 6464, 61350, 2488, 5427, 3799, 7115, 1658
Genre 6	998, 4721, 70687, 3446, 5222, 5522, 3494, 166461, 82, 4703
Genre 7	4733, 4450, 139642, 165549, 160565, 96004, 3089, 2070, 932, 522
Genre 8	130634, 3404, 6022, 611, 158783, 1241, 27912, 1772, 86320, 2693
Genre 9	74754, 27251, 191, 85354, 93270, 2290, 6464, 461, 99764, 4289
Genre 10	2068, 1627, 113829, 3214, 3030, 4381, 86347, 166534, 1606, 1416
Genre 11	89118, 51931, 53974, 158783, 3925, 63436, 613, 46572, 4474, 70946
Genre 12	25850, 446, 5034, 165549, 3525, 7700, 5485, 2772, 1295, 98279
Genre 13	901, 54648, 3223, 159858, 33725, 120635, 4052, 3616, 5466, 3837
Genre 14	484, 3223, 8042, 1211, 3272, 2693, 26258, 4056, 49286, 3706
Genre 15	5048, 8372, 6686, 156609, 61729, 105355, 49274, 5034, 128968, 1804
Genre 16	34338, 185029, 25850, 7318, 4821, 3538, 52712, 8138, 1128, 3537
Genre 17	4735, 58554, 3061, 2587, 72167, 42018, 137857,

	52712, 4617, 90866
Genre 18	7116, 26409, 7883, 8482, 8521, 7991, 45880, 70946, 5919, 5222
Genre 19	2807, 72171, 4663, 3225, 79251, 2568, 58347, 1334, 85367, 2007
Genre 20	112804, 30, 8199, 5666, 57532, 148881, 143367, 6669, 25825, 4289

Table 2. Top 10 movies in each 20 genre

There is a connection between the latent factors and the movie genres. The number of latent factors decides the number of genres, so that different number of latent factors makes top 10 movies in each genre different.

Question 24

In this part we designed an Matrix Factorization (MF) with bias collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated its performance using 10-fold cross-validation. The regularization parameter was set to default here. The number of latent factor k was also swept from 2 to 50 in the step size of 2, and for each k the average RMSE and average MAE were computed across all 10 folds. The plots of average MAE and RMSE against k are shown as below:

Figure 25. Average RMSE using MF with bias against k

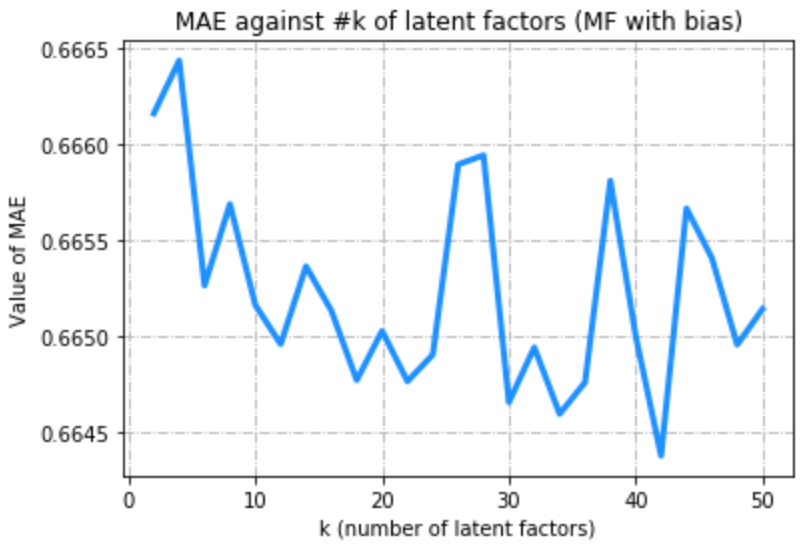

Figure 26. Average MAE using MF with bias against k

Question 25

Based on the plots in question 24, the optimal number of latent factors, which is the value of k that gives the minimum average RMSE or MAE, were shown as below:

	RMSE	MAE
Optimal k	50	30
Minimum Average	0.8650408150286013	0.6644484105171823

Table 3. The optimal number of latent factors and minimum average value from RMSE and MAE using MF with bias

Here we assumed that 30 is the optimal number for k.

Question 26

We designed an MF with bias collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluated its performance using 10-fold cross validation. The number of latent factors k was swept from 2 to 50 in the step sizes of 2, and for each k the average RMSE was computed across all 10 folds. The plot of the average RMSE (Y-axis) against k (X-axis) is shown as below:

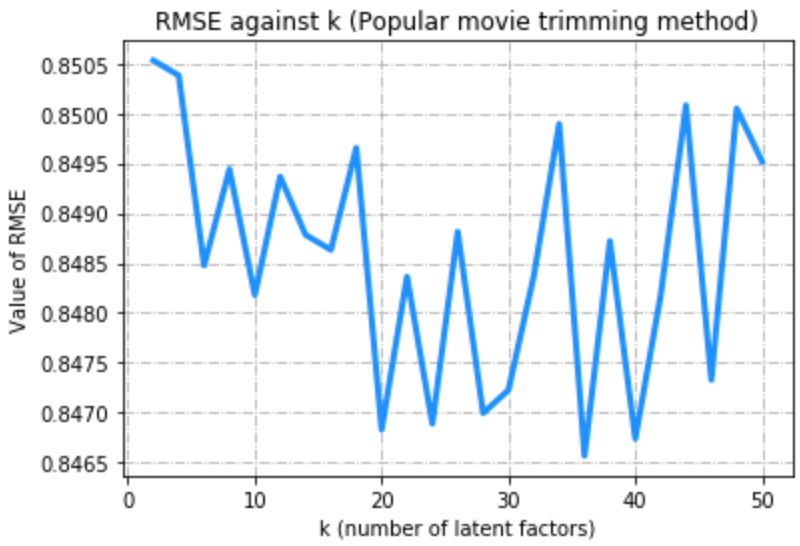

Figure 27. Average RMSE using MF with bias against k on popular movie trimmed test set

And the minimum average value of RMSE is: 0.8465675329963434.

Question 27

We designed an MF with bias collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluated its performance using 10-fold cross validation. The number of latent factors k was swept from 2 to 50 in the step sizes of 2, and for each k the average RMSE was computed across all 10 folds. The plot of the average RMSE (Y-axis) against k (X-axis) is shown as below:

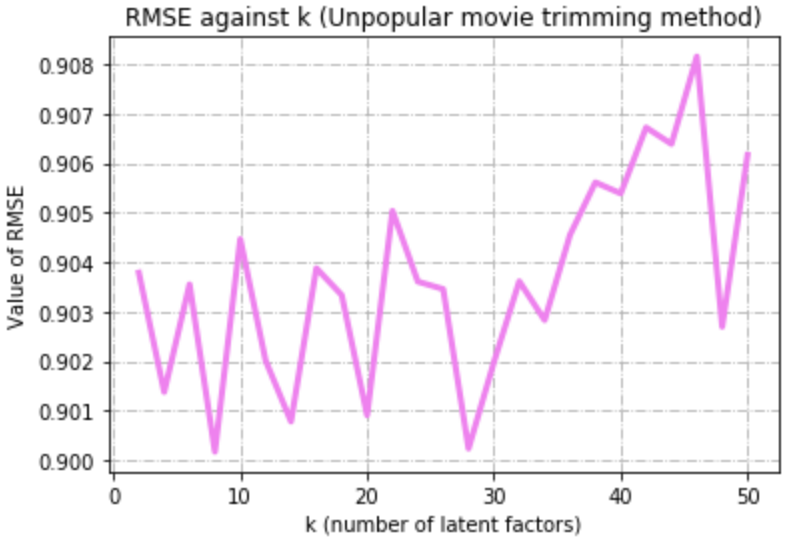

Figure 28. Average RMSE using MF with bias against k on unpopular movie trimmed test set

And the minimum average value of RMSE is: 0.9001695506555052.

Question 28

We designed an MF with bias collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluated its performance using 10-fold cross validation. The number of latent factors k was swept from 2 to 50 in the step sizes of 2, and for each k the average RMSE was computed across all 10 folds. The plot of the average RMSE (Y-axis) against k (X-axis) is shown as below:

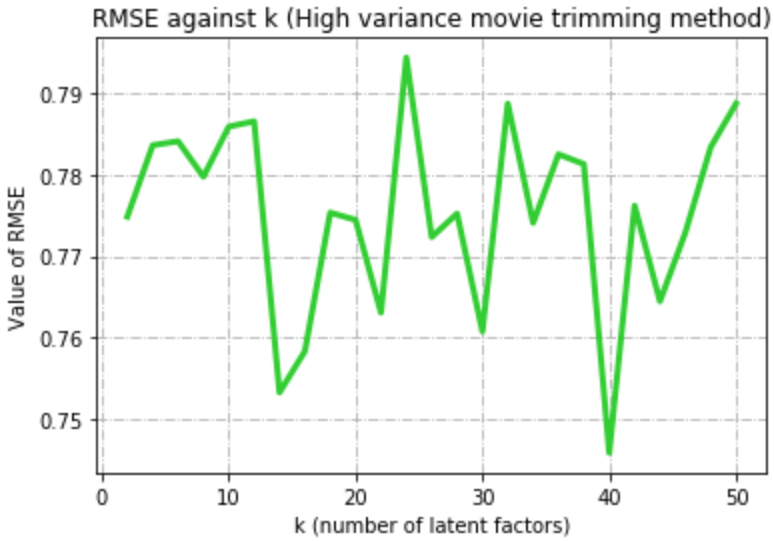

Figure 29. Average RMSE using MF with bias against k on high variance movie trimmed test set

And the minimum average value of RMSE is: 0.7459481762000919.

Question 29

The ROC curve was plotted for the MF with bias collaborative filter designed in question 24 for threshold values [2.5, 3, 3.5, 4] here. And the optimal number of latent factors found in question 25 was used here, which is k = 30. For each of the plot, the area under the curve (AUG) was also reported in the legend of the plot as below:

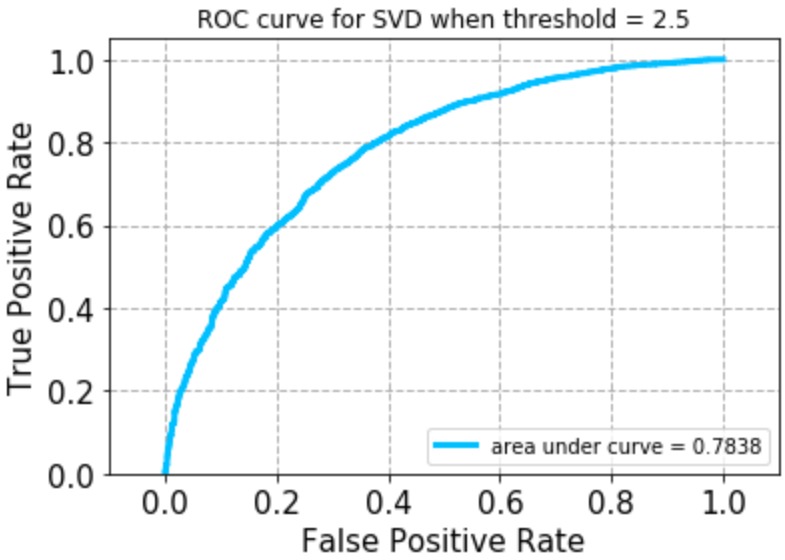

Figure 30. ROC curve for MF filter with bias when threshold = 2.5

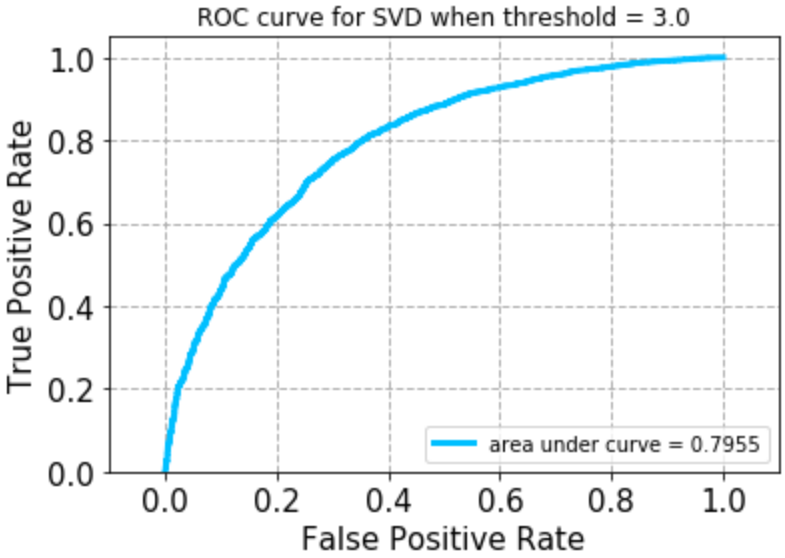

Figure 31. ROC curve for MF filter with bias when threshold = 3.0

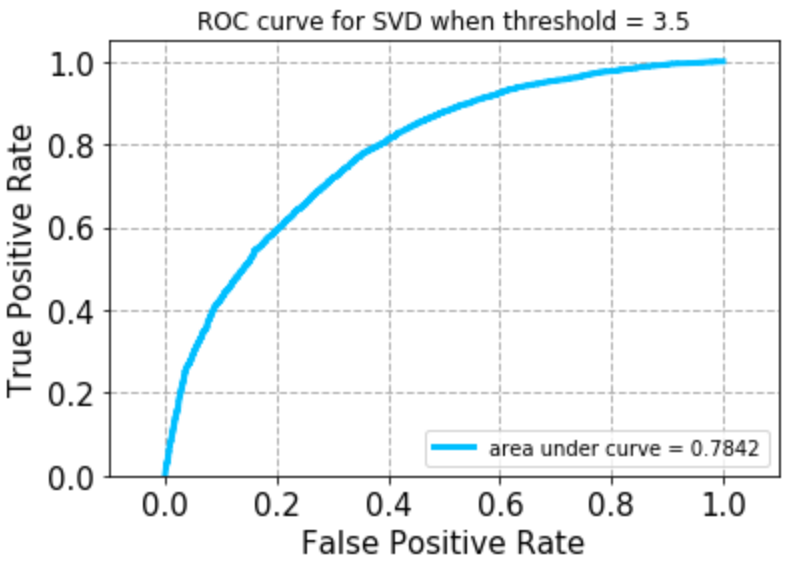

Figure 32. ROC curve for MF filter with bias when threshold = 3.5

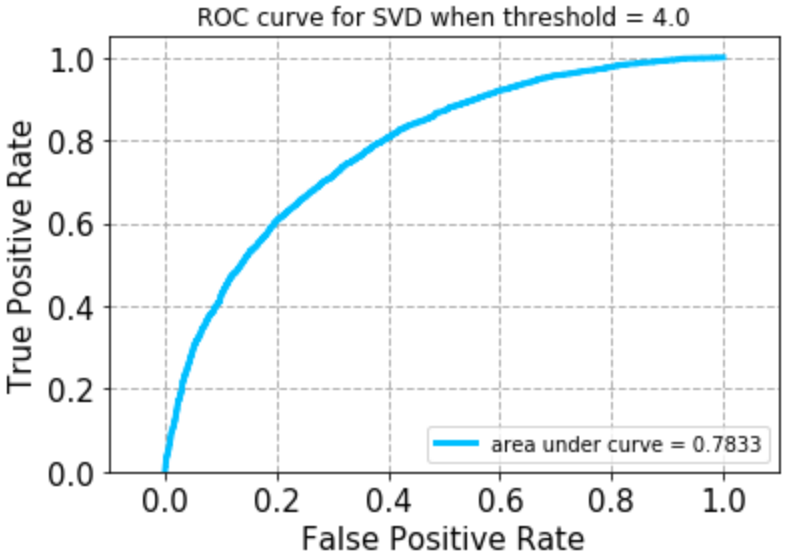

Figure 33. ROC curve for MF filter with bias when threshold = 4.0

Question 30

We designed a naive collaborative filter to predict the ratings of movies in the MovieLens dataset and evaluated its performance using 10 fold cross validation. The predicted rating of user *i* for item *j*, denoted by $\hat{r}_{ij}$ is given by equation below. μ_i is the mean rating of user *i*.

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

We used a single set of μ_i's calculated on the entire dataset and computed the average RMSE by averaging the RMSE across all 10 folds. The result is:

$$Avg(RMSE) = 0.9410964549697685$$

Question 31

We designed a naive collaborative filter to predict the ratings of movies in the popular movie trimmed test set and evaluated its performance using 10-fold cross validation. The average RMSE is:

$$Avg(RMSE) = 0.9303045438426659$$

Question 32

We designed a naive collaborative filter to predict the ratings of movies in the unpopular movie trimmed test set and evaluated its performance using 10-fold cross validation. The average RMSE is:

$$Avg(RMSE) = 0.9609220387549572$$

Question 33

We designed a naive collaborative filter to predict the ratings of movies in the high variance movie trimmed test set and evaluated its performance using 10-fold cross validation. The average RMSE is:

$$Avg(RMSE) = 0.8446559041009639$$

Question 34

We set the threshold to be 3 and plotted the ROC curves for the k-NN, NNMF and MF with bias based collaborative filters in the same figure, as shown in Fig 34.

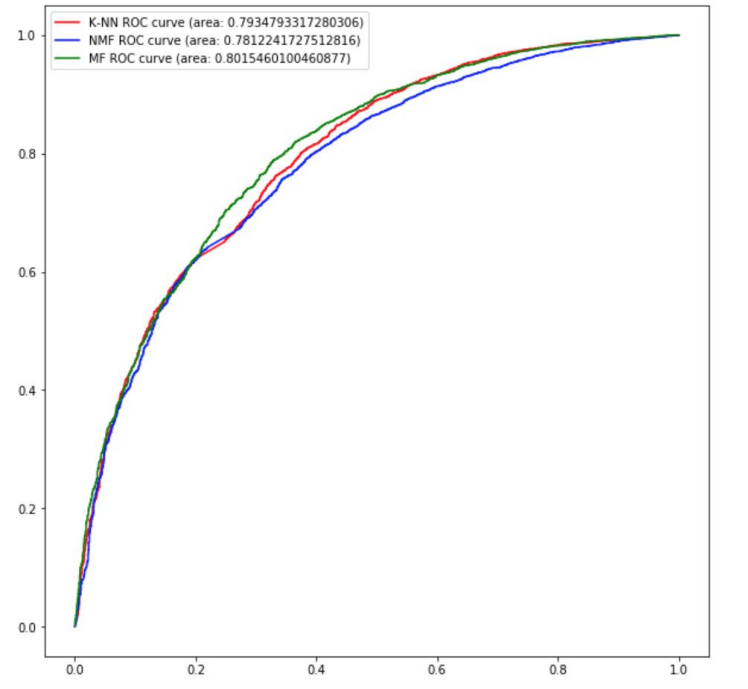

Figure 34. ROC Curves for k-NN, NNMF and MF with Threshold = 3

As is shown in Fig 34, the AUCs for k-NN, NNMF and MF are

$$AUC(k-NN) = 0.7934793317280306$$
$$AUC(NNMF) = 0.7812241727512816$$
$$AUC(MF) = 0.8015460100460877$$

Therefore, the performance of MF filter is the best in predicting the ratings of the movies.

Question 35

The mathematical definitions of precision and recall are given below respectively:

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

Where $S(t)$ means the set of items of size t recommended to the user and G represents the set of items liked by the user.

Based on the definition, precision means the percentage of items recommended to the user that the user actually likes, over the total number of recommended items. Whereas recall means the percentage of items recommended to the user that the user actually likes, over the total number of user-liked items.

Question 36

We plotted average precision (Y-axis) against t (X-axis), average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis) for the ranking obtained using k-NN collaborative filter predictions with k = 20, found in Question 11 and swept t from 1 to 25 with step size 1. The results are shown in Fig 35, 36, 37.

It shows that the average precision decreases as t increases, whereas the average recall increases as t increases. Also, the average precision decreases as the average recall increases.

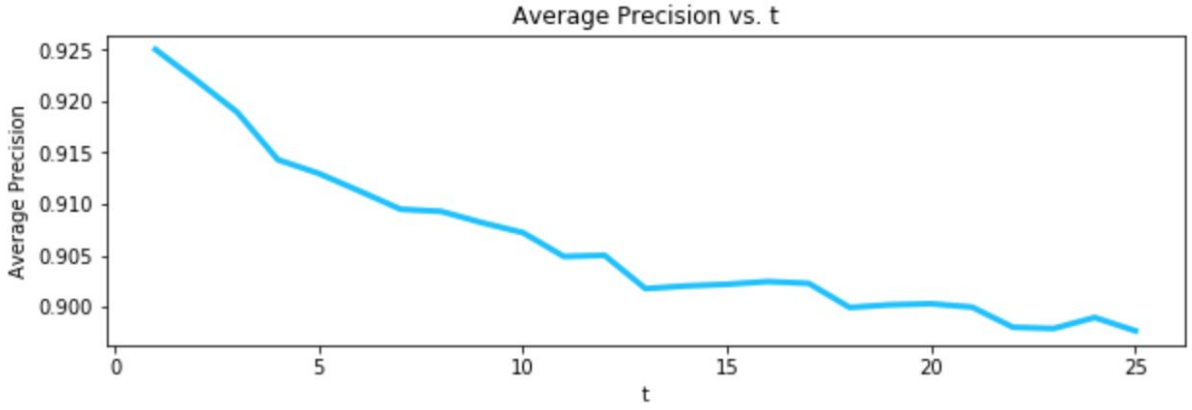

Figure 35. Average Precision Against t

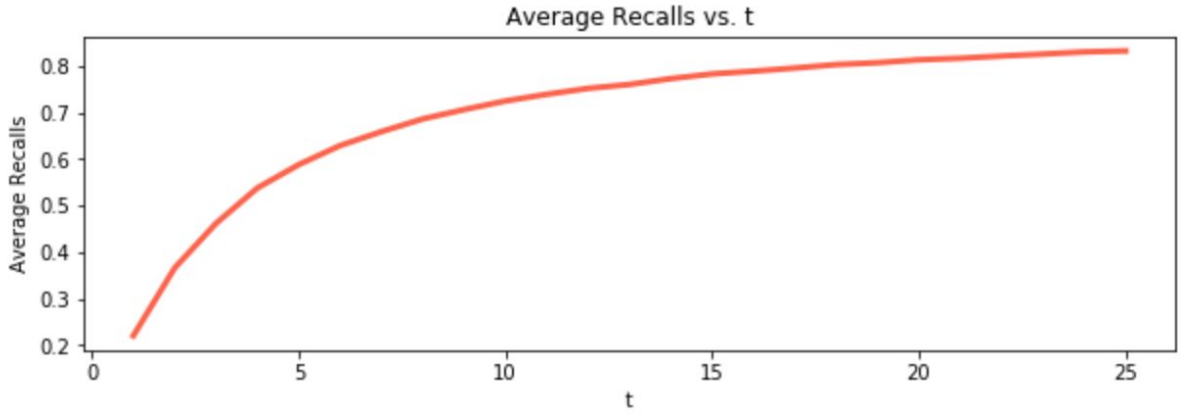

Figure 36. Average Recall Against t

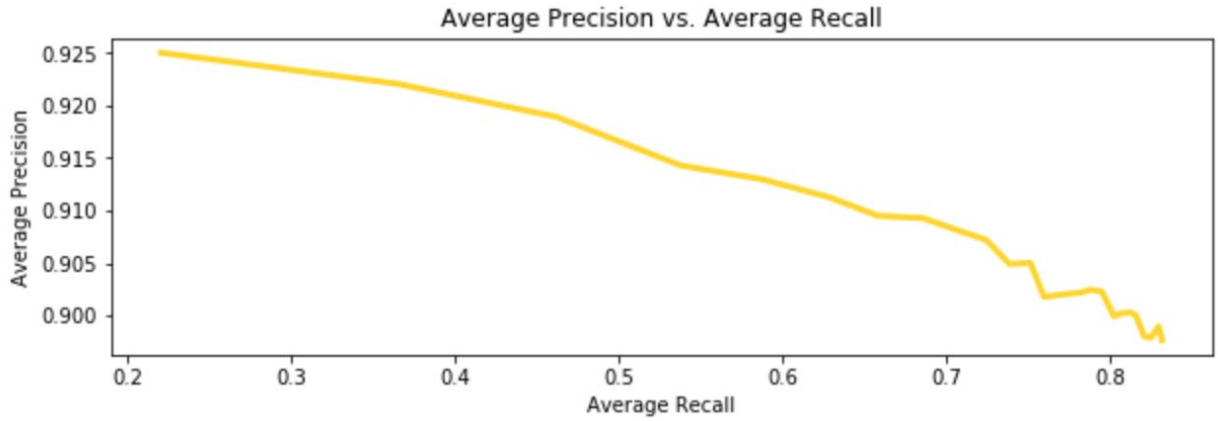

Figure 37. Average Precision Against Average Recall

Question 37

We plotted average precision (Y-axis) against t (X-axis), average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis) for the ranking obtained using NNMF-based collaborative filter predictions with optimal number of latent factors found in Question 18, 20 and swept t from 1 to 25 with step size 1. The results are shown in Fig 38, 39, 40.

It shows that the trend of precision and recall versus t is the same: average precision decreases with t increasing and average recall increases with t increasing. But it is not exactly monotonic as that is in k-NN collaborative filter predictions.

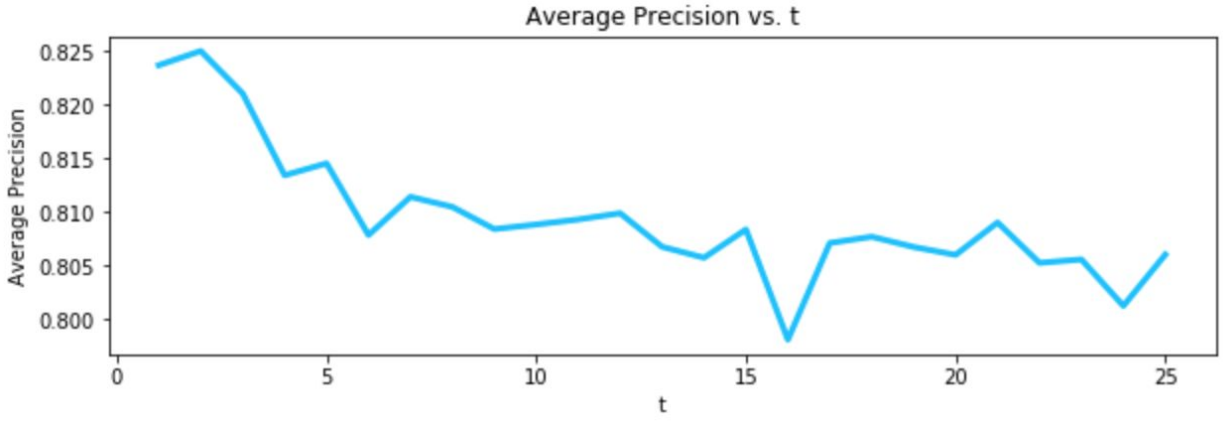

Figure 38. Average Precision Against t

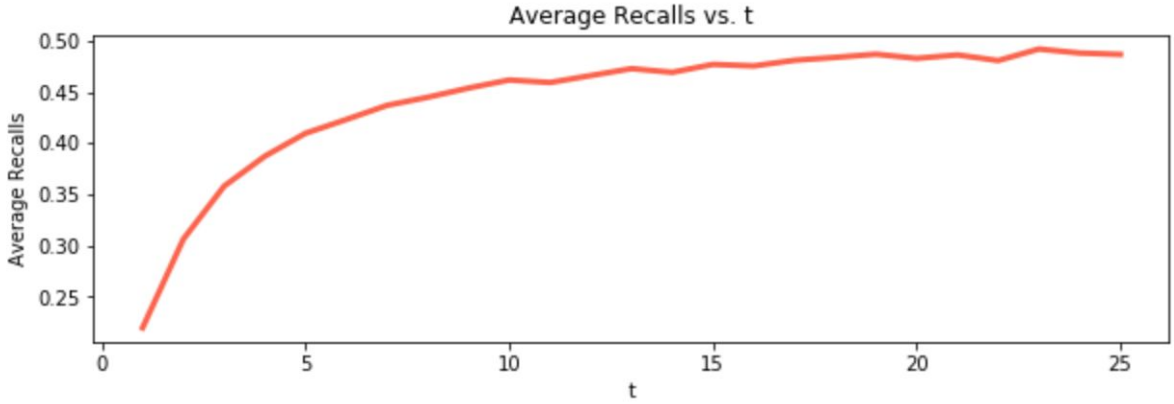

Figure 39. Average Recall Against t

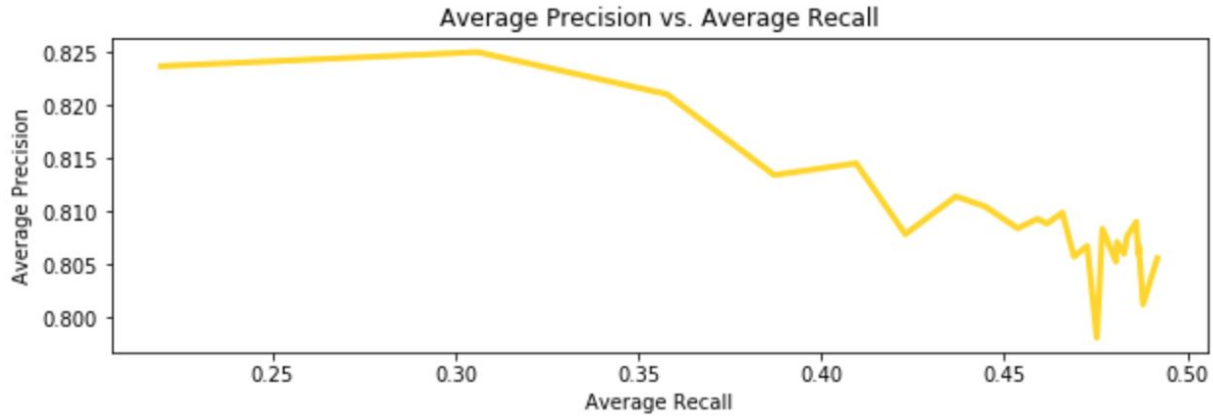

Figure 40. Average Precision Against Average Recall

Question 38

We plotted average precision (Y-axis) against t (X-axis), average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis) for the ranking obtained using MF with bias-based collaborative filter predictions with optimal number of latent factors found in Question 25, 30 and swept t from 1 to 25 with step size 1. The results are shown in Fig 41, 42, 43.

It shows that the trend of precision and recall versus t is the same: average precision decreases with t increasing and average recall increases with t increasing. But it is not exactly monotonic as that is in k-NN collaborative filter predictions.

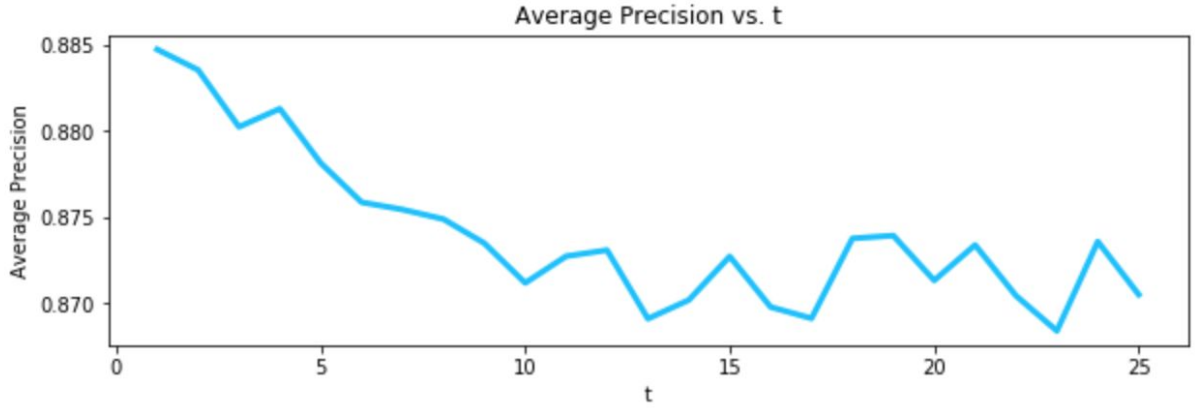

Figure 41. Average Precision Against t

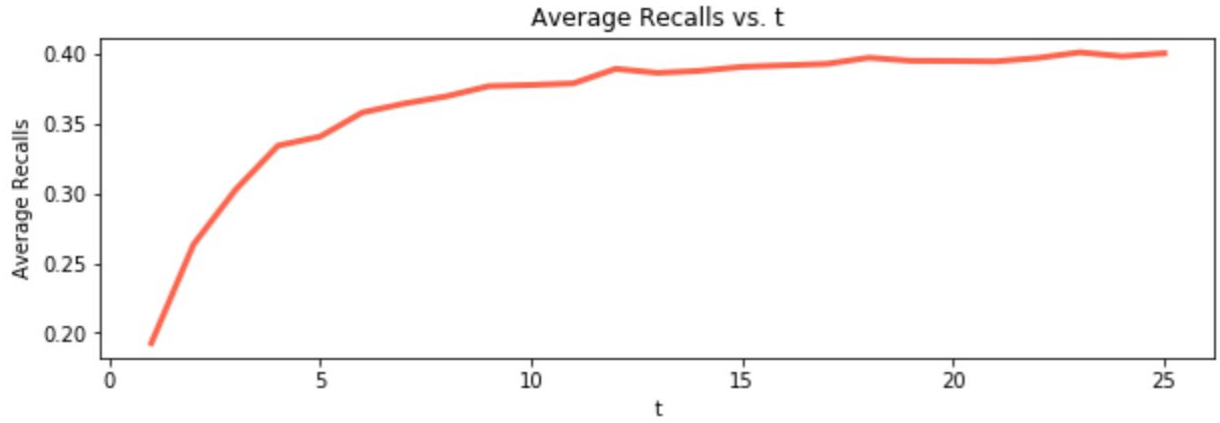

Figure 42. Average Recall Against t

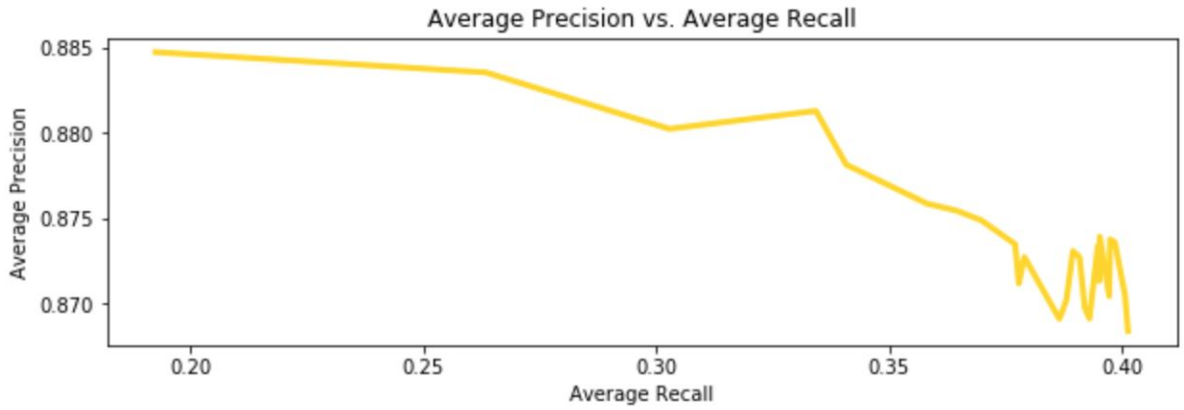

Figure 43. Average Precision Against Average Recall

Question 39

We plotted the precision-recall curve obtained in Question 36, 37, 38 in the same figure, shown in Fig 44. It shows in Fig 44 that the result with k-NN collaborative filter predictions is the best.

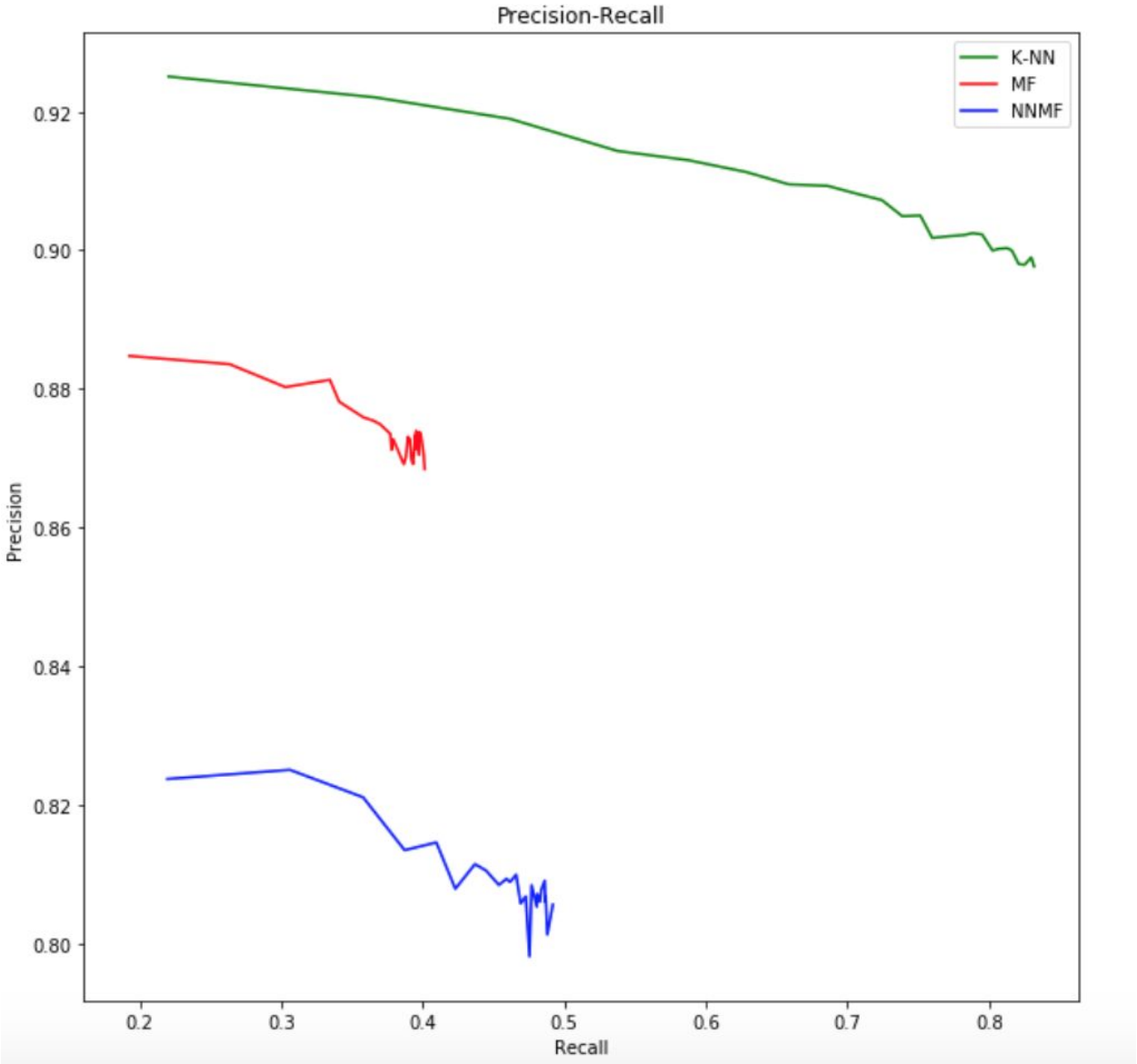

Figure 44. Precision Against Recall