**EE 219 Project 3 – Report**

**Collaborative Filtering**

Fangyao Liu 204945018

Xuan Hu 505031796

Yanzhe Xu 404946757

Zhechen Xu 805030074

**Content**

**Introduction ……………………………………………………………………………. 1**

**MovieLens dataset ………………………………………………………………………1**

Question 1 ……………………………………………………………………………......1

Question 2 ……………………………………………………………………………......1

Question 3 ……………………………………………………………………………......2

Question 4 ……………………………………………………………………………......2

Question 5 ……………………………………………………………………………......3

Question 6 ……………………………………………………………………………......3

**Neighborhood-based collaborative filtering …………………………………………..4**

Question 7 ……………………………………………………………………………......4

Question 8 ……………………………………………………………………………......4

Question 9 ……………………………………………………………………………......4

Question 10 ……………………………………………………………………………....5

Question 11 ……………………………………………………………………………....5

Question 12 ……………………………………………………………………………....6

Question 13 ……………………………………………………………………………....7

Question 14 ……………………………………………………………………………....8

Question 15 ……………………………………………………………………………....9

**Model-based collaborative filtering …………………………………………………..11**

Question 16 …………………………………………………………………………..... 11

Question 17 ……………………………………………………………………………..13

Question 18 ……………………………………………………………………………..13

Question 19 ……………………………………………………………………………..14

Question 20 ……………………………………………………………………………..15

Question 21 ……………………………………………………………………………..16

Question 22 ……………………………………………………………………………..17

Question 23 ……………………………………………………………………………..19

Question 24 ……………………………………………………………………………..20

Question 25 ……………………………………………………………………………..20

Question 26 ……………………………………………………………………………..21

Question 27 ……………………………………………………………………………..22

Question 28 ……………………………………………………………………………..23

Question 29 ……………………………………………………………………………..24

**Naïve collaborative filtering …………………………………………………………..26**

Question 30 ……………………………………………………………………………..26

Question 31 ……………………………………………………………………………..26

Question 32 ……………………………………………………………………………..26

Question 33 ……………………………………………………………………………..26

**Performance comparison ……………………………………………………………..27**

Question 34 ……………………………………………………………………………..27

**Ranking ……………………………………………………………………………......28**

Question 35 **……………………………………………………………………**………..29

Question 36 ……………………………………………………………………………..29

Question 37 ……………………………………………………………………………..30

Question 38 ……………………………………………………………………………..31

Question 39 ……………………………………………………………………………..32

**Introduction**

In this project, we built a recommender system for the MovieLens dataset. Based on multiple users, movies and the movie rating information, we are able to give out movies recommendation for users.

We built recommendation system using collaborative filtering methods. The basic idea of collaborative filtering methods is that these unspecified ratings can be imputed because the observed ratings are often highly correlated across various users and items. We will implement and analyze the performance of two types of collaborative filtering methods: Neighborhood-based collaborative filtering and Model-based collaborative filtering.

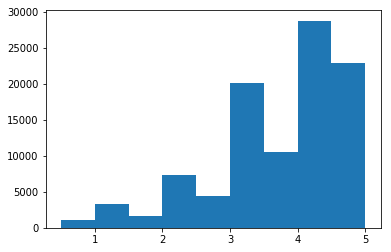
**MovieLens dataset**

In this project, we built a recommendation system to predict the ratings of the movies in the MovieLens dataset. We only used the movie rating list. We assume that the rating matrix is a m by n matrix. The (i, j) entry of the matrix is the rating of user i for movie j and is denoted by .

**Question1:**

The sparsity of the movie dataset is 0.0164391416087

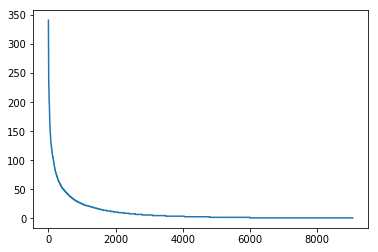
**Question2:**



*figure 1 Frequency of the rating values (rating values – count plot)*

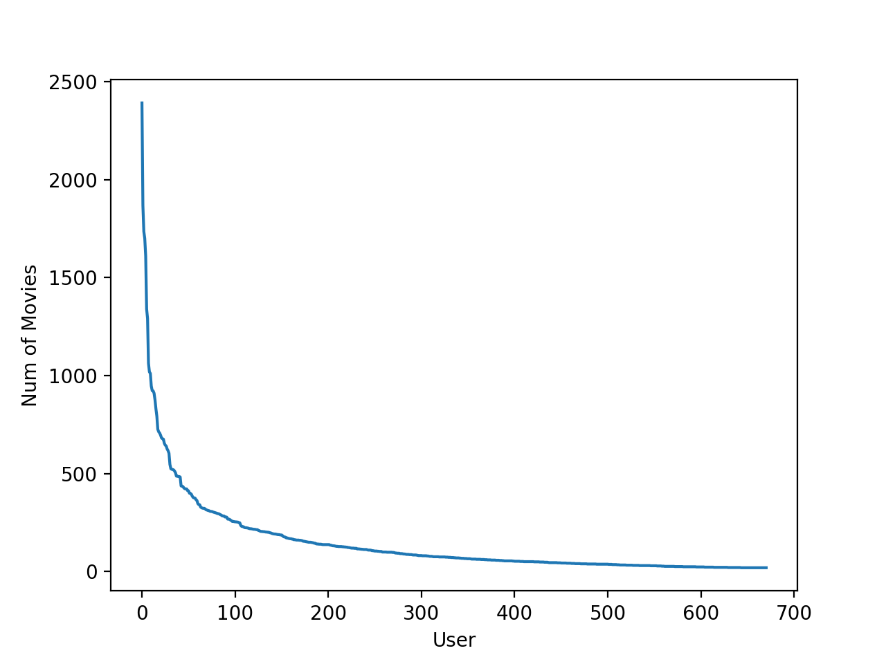
From figure 1, we can find out most movies (around 82%) are distributed between score 3 and score 5, which are not evenly distributed.

**Question3:**



*figure 2 Distribution of ratings among movies (movie index – number of ratings plot)*

**Question4:**



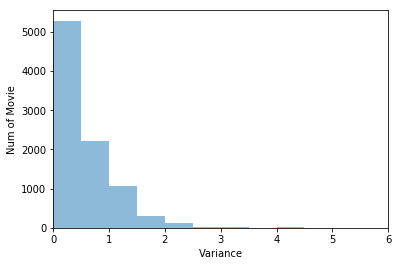
*figure 3 Distribution of ratings among users (user index – number of ratings)*

In this figure 3, the x axis does not mean the index of user. It will be total messy if applying user index to x axis so that we store the user index in an array and each point of the graph can use its x-coordinate to get its corresponding user ID.

**Question5:**

From question 3 we can see that about 500 movies in the front of x axis have many ratings. But other movies have only little evaluation. Thus, we can divide this question into two parts. On the one hand, the trend of curve moves from high to low. That means movies in the front are popular and attractive to most users. When we don’t know much about user’s information, the movies we recommended should be concentrated on the front part. On the other hand, based on enough users’ information we got, we should remove the front part which has high rating ratio in order to recommend specific movie to their lover. Since the movies in the front occupy too much proportion, we can’t extract useful details without cutting them.

**Question6:**



*figure 4 Variance – Number of movies plot*

From this figure 4, it is obvious that most movies have small variance of their ratings. It is easy to understand that since people always have very similar comment towards one specific movie. If it is a good one, people will give high rating because they think it’s awesome or not bad. Otherwise, they give bad feedback about it. Therefore, most people would rate based on the quality of the film and then their ratings would be similar.

**Neighborhood-based collaborative filtering**

Neighborhood-based methods is referring to use either user-user similarity or item-item similarity to make predictions from a ratings matrix. In this project, we only implemented user-based collaborative filtering. User-based neighborhoods are defined in order to identify similar users to the target user for whom the rating predictions are being computed. By computing the similarity function, we are able to determine the neighborhood of the target user. In this project, we will use Pearson-correlation coefficient to compute the similarity between users.

**Question7：**

*formula 1 Formula for in terms of and*

**Question8：**

The meaning of  union  is that the movies that have been watched by u and v. And  union  is possible to be empty when u and v haven't watched same movies

**Question 9:**

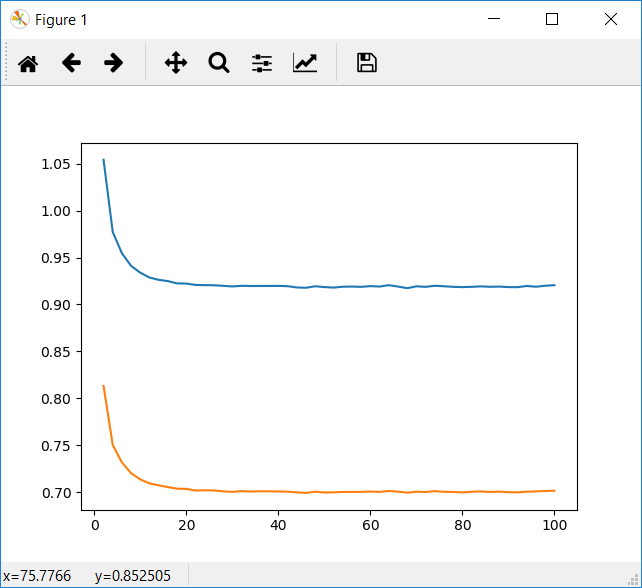
Considering some people could give high scores to all movies and someone give really low scores, we can divide scores into two parts: personal offset and movie rating. Similar raters would have similar movie rating but different personal offset. Therefore, mean-centering data removes personal offset but only deal with movie rating. Then, add back personal offset of people we are going to predict.

**K-NN collaborative filter**

k-Nearest neighbor of user u, denoted by Pu, is the set of k users with the highest Pearson-correlation coefficient with user u. In this part, we designed 10-fold validation to test the model designed earlier.

**Question 10**

In this question, we designed a k-NN collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated its performance using 10-fold cross validation. The blue line is k – average RMSE plot and the orange line is k – average MAE plot.



*figure 5 k – Average RMSE and k – Average MAE plot (k-NN collaborative filter)*

**Question 11**

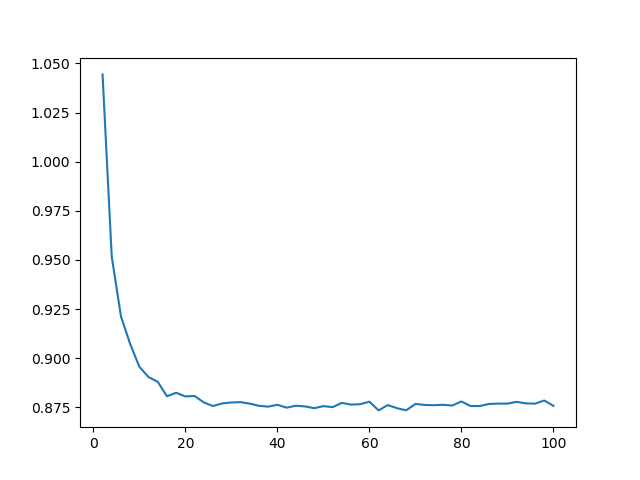
From the figure 5, minimum k should be around 20.

**Filtering performance on trimmmed test set**

In this part, we analyzed the performance of the k-NN collaborative filter in predicting the ratings of the movies in the trimmed test set. We considered popular movie trimming, unpopular movie trimming and high variance movie trimming in this project.

**Question 12**

In this problem, 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. In this trimming, the test set are trimmed to only contained movies that has receive more than 2 ratings.

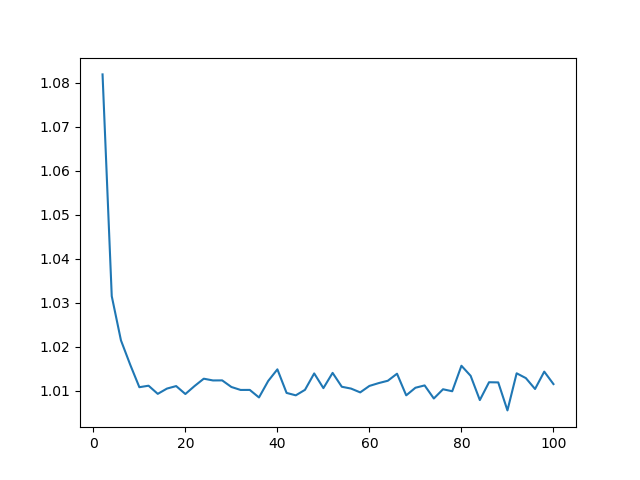


*figure 6 k – Average RMSE plot (popular movie trimmed)*

The minimum average RMSE is 0.873460277067.

**Question 13**

In this problem, 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. In this trimming, the test set are trimmed to only contained movies that has receive less or equal to 2 ratings.

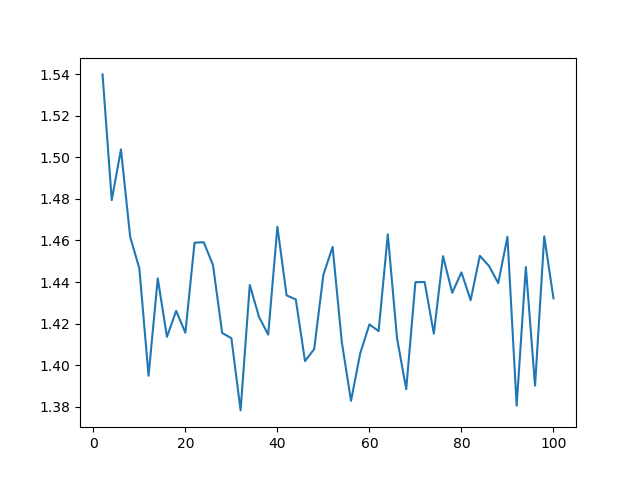


*figure 7 k – Average RMSE plot (unpopular movie trimmed)*

The minimum average RMSE is 1.00553781656.

**Question 14**

In this problem, we designed a k-NN collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluated its performance using 10-fold cross validation. In this trimming, the test set are trimmed to only contained movies that has variance of at least 2 and received less than 5 ratings.

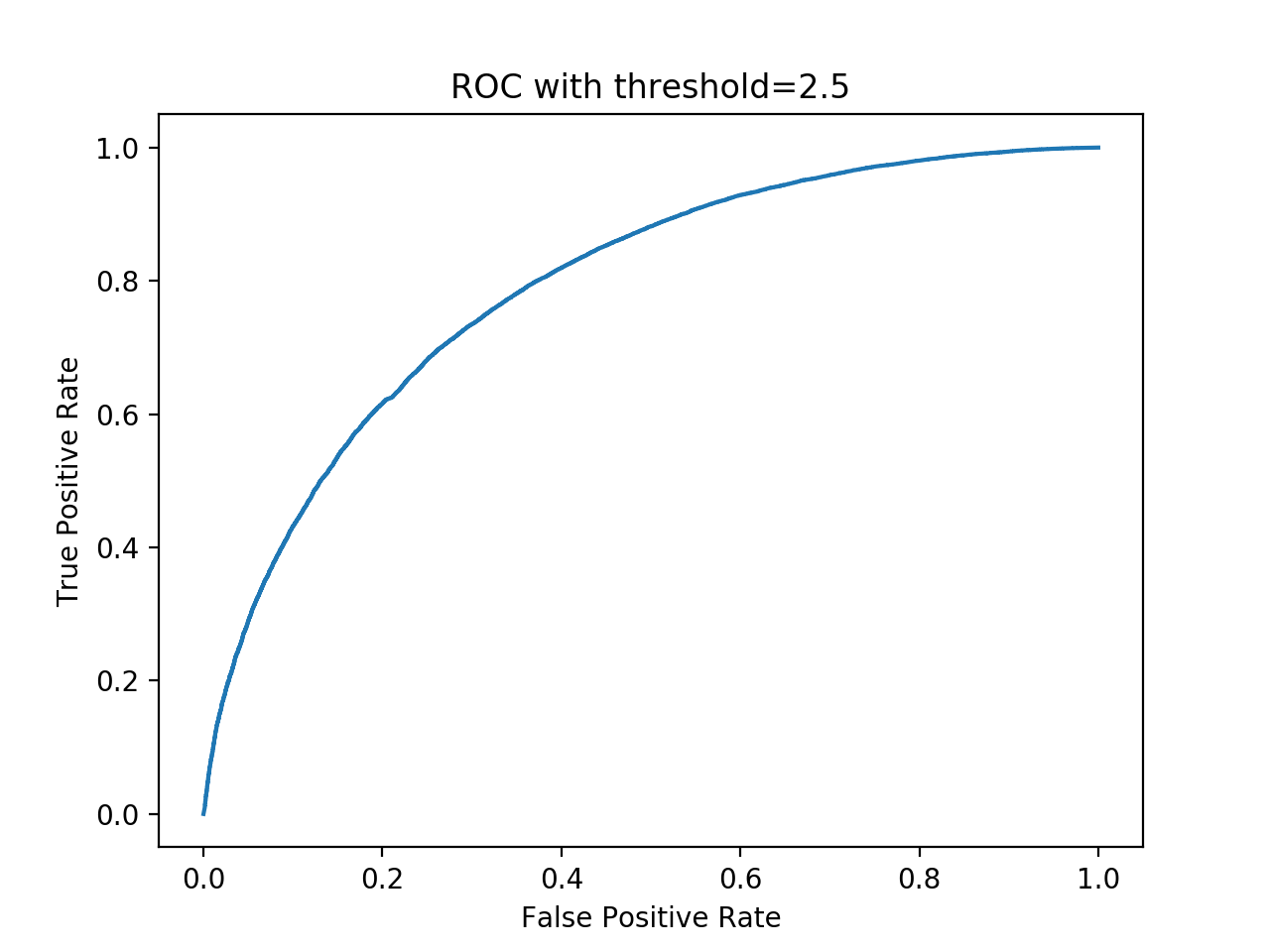


*figure 8 k – Average RMSE plot (high variance movie trimmed)*

The minimum average RMSE is 1.3783048837.

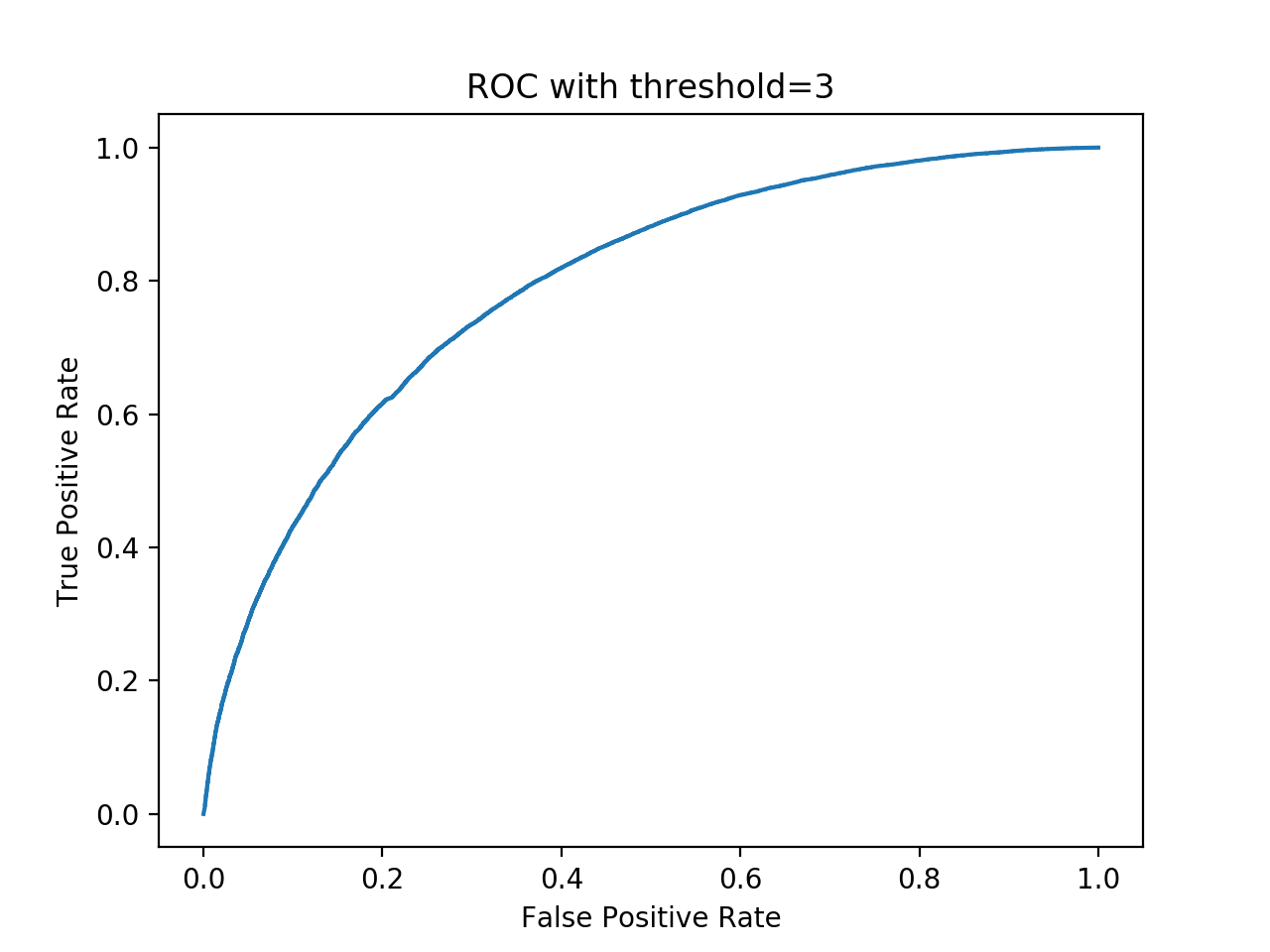
**Question 15**

k-NN collaborative filter ：



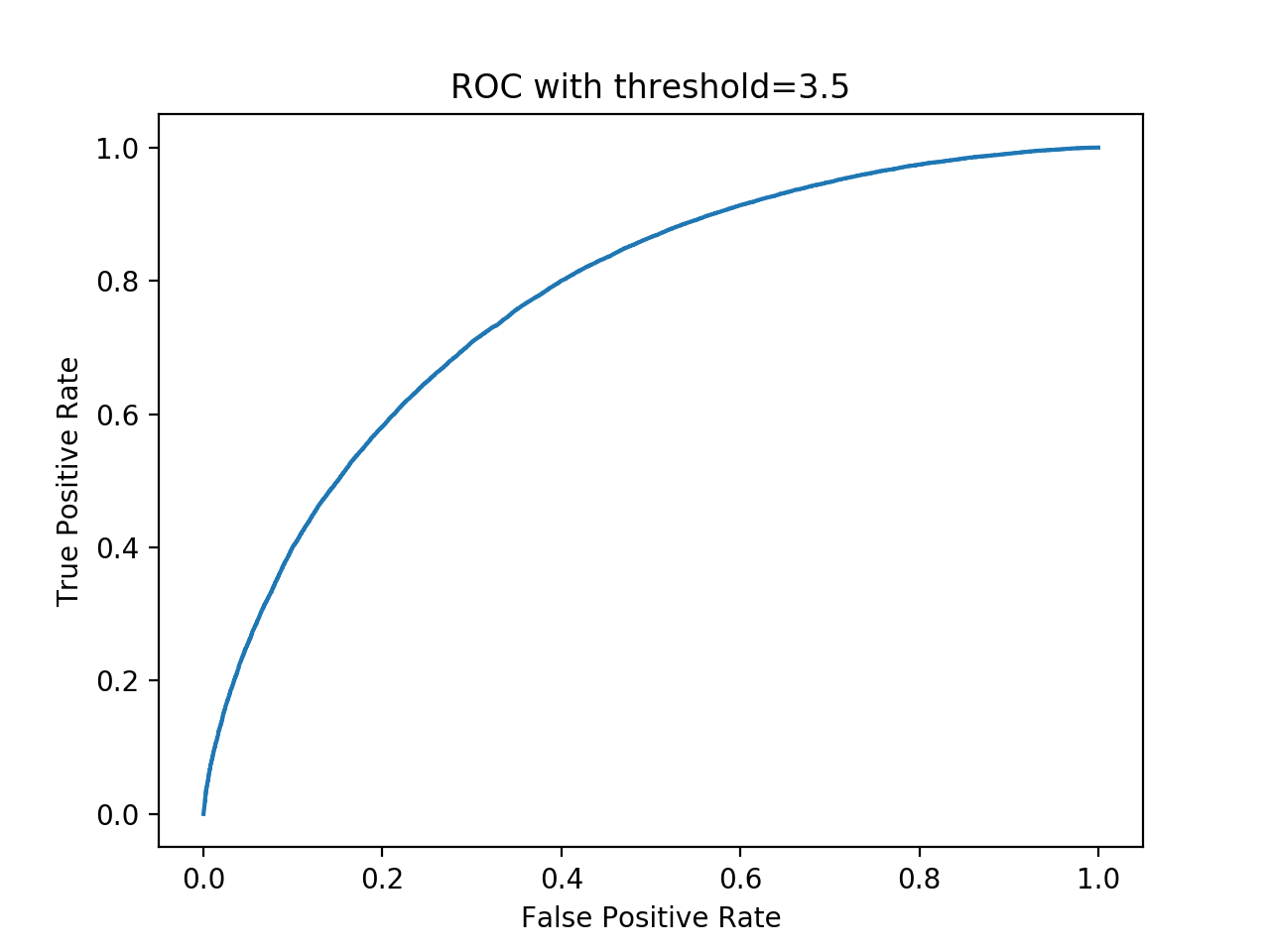
*figure 9 ROC with threshold = 2.5*

Area under curve value is 0.758763 with threshold=2.500000



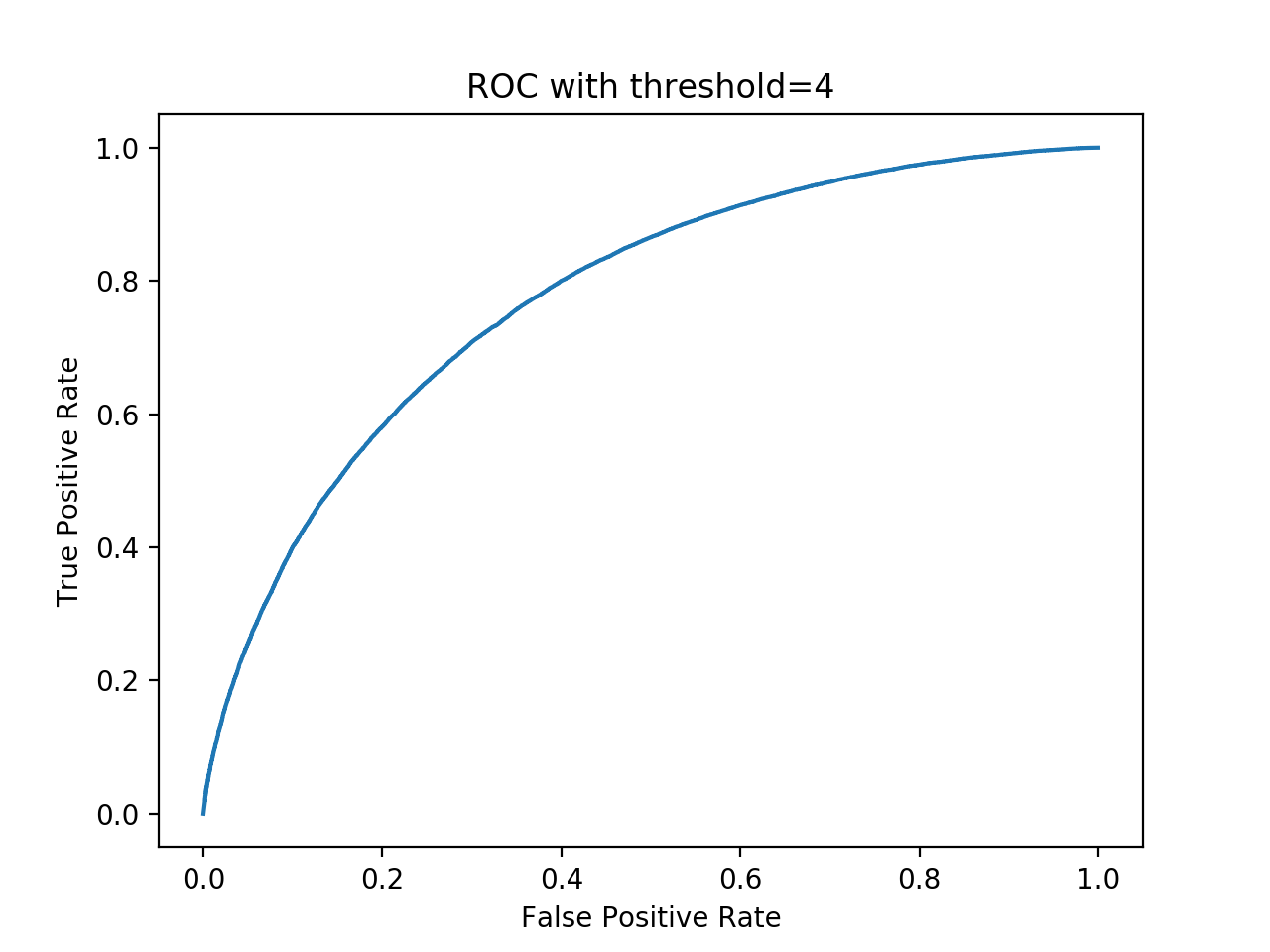
*figure 10 ROC with threshold = 3*

Area under curve value is 0.762489 with threshold=2.500000



*figure 11 ROC with threshold = 3.5*

Area under curve value is 0.746651 with threshold=3.500000



*figure 12 ROC with threshold = 4*

Area under curve value is 0.738854 with threshold=4.000000

**Model-based collaborative filtering**

In model-based collaborative filtering, models are developed using machine learning algorithms to predict users’ rating of unrated item. We implemented latent factor based models for collaborative filtering in this project. Because of the redundancies and highly correlated between rows and columns of the rating matrix R, we are able to find a low-rank matrix to estimate the missing entries, which is called matrix factorization.

Since the rating matrix R is sparse, it might cause overfitting. Thus, we used regularization to address this problem.

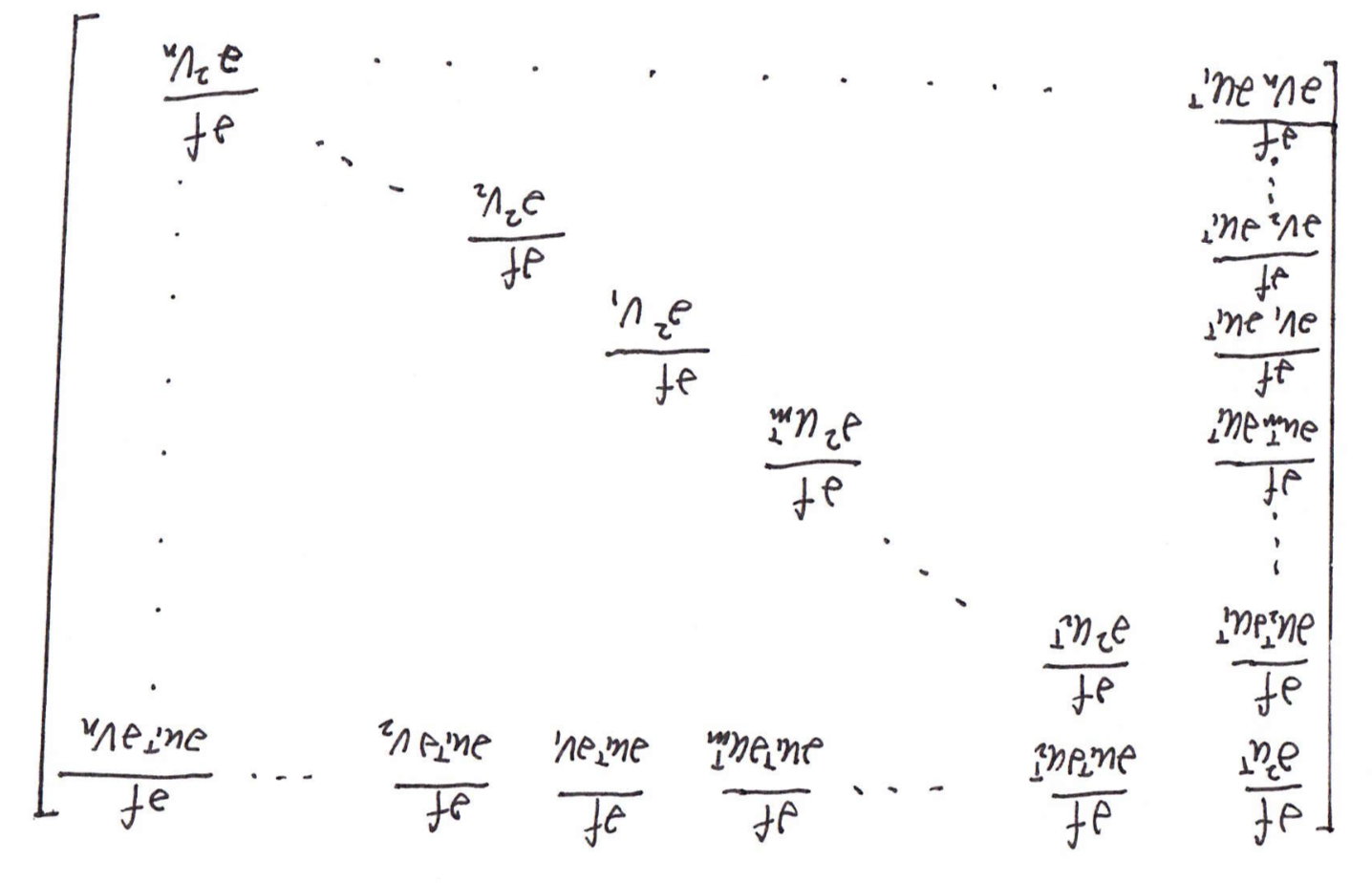
*formula 2 Unconstrained matrix factorization formulation*

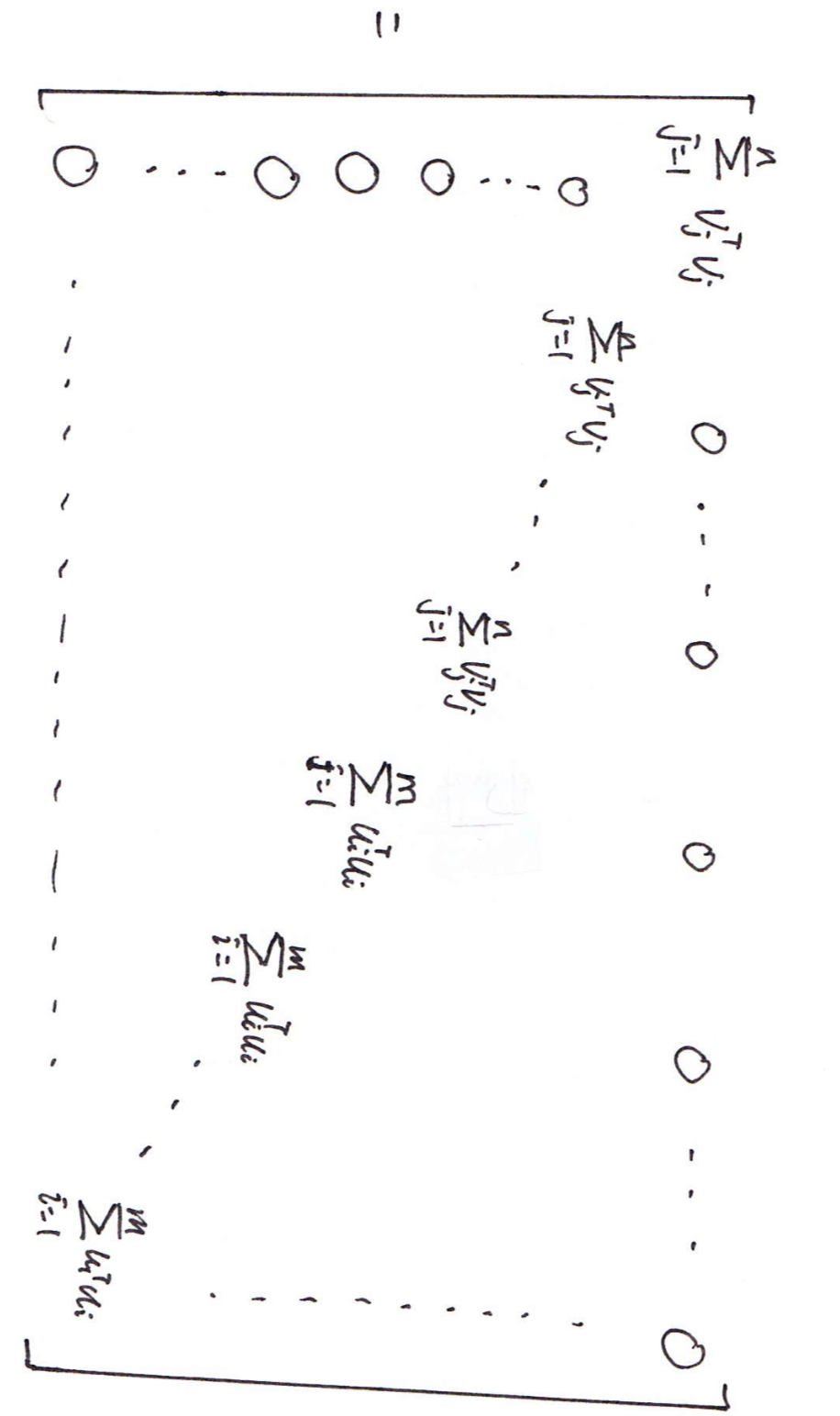
In this project, we will explore two variations to formula 2, which are Non-negative matrix factorization(NNMF) and Matrix factorization with bias (MF with bias)

**Question 16**

The optimization problem given by equation 5 is convex. The reason is that for variables U and V, the hessian matrix of second derivative is positive semidefinite.

*formula 3 Objection Function*

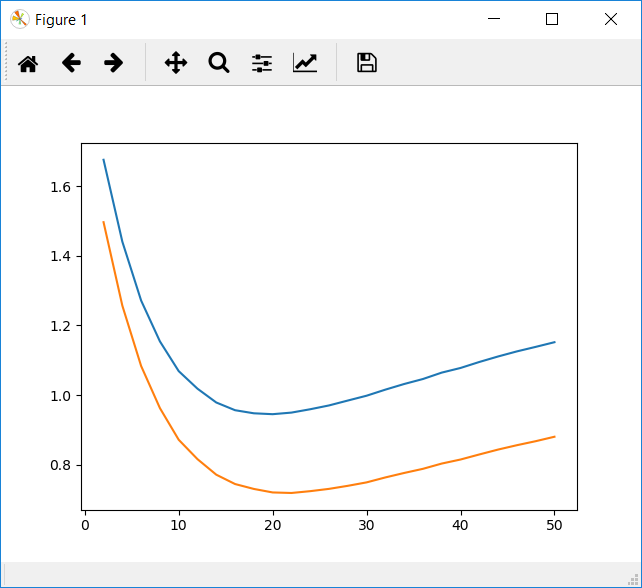




*formula 4 Hessian matrix*

**Question 17**

In this problem, we designed a NNMF-based collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated its performance using 10-fold cross-validation. The blue line is k – average RMSE plot and the orange line is k – average MAE plot.



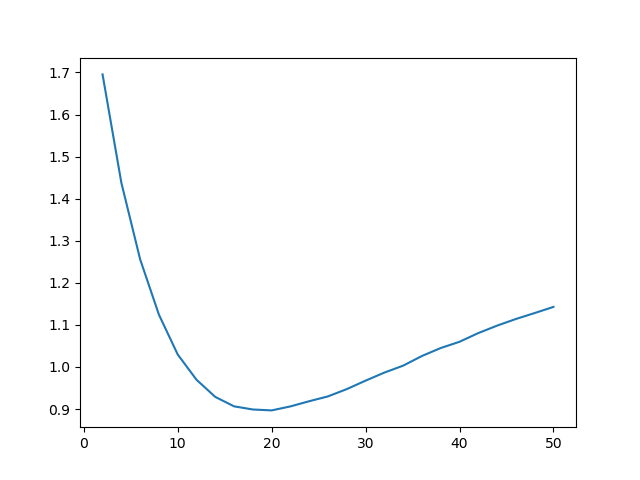
*figure 13 k – Average RMSE and k – MAE plot (NNMF-based collaborative filter)*

**Question 18**

The minimum k should be 20. The optimal latent factors are almost the same as the number of movie genres (19).

**Question 19**

In this problem, we designed a NNMF 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.

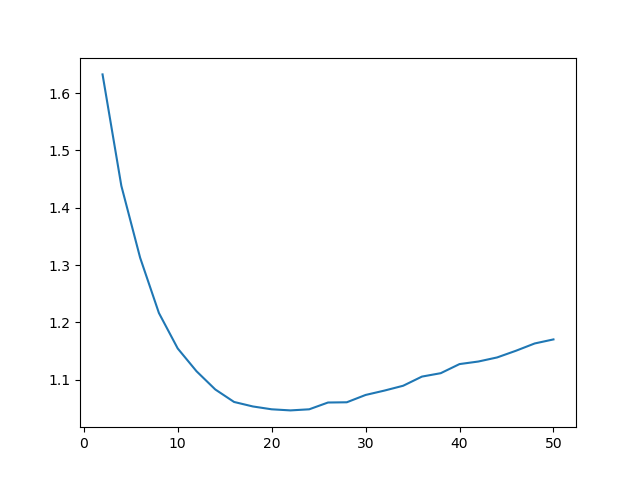


*figure 14 k – Average RMSE plot (popular movie trimmed)*

The minimum average RMSE is 0.89685481431.

**Question 20**

In this problem, we designed a NNMF 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.

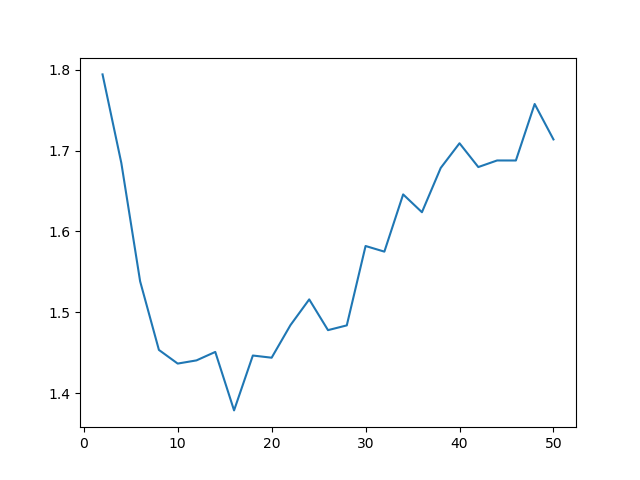


*figure 15 k – Average RMSE plot (unpopular movie trimmed)*

The minimum average RMSE is 1.04642297317.

**Question 21**

In this problem, we designed a NNMF collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluated its performance using 10-fold cross validation.

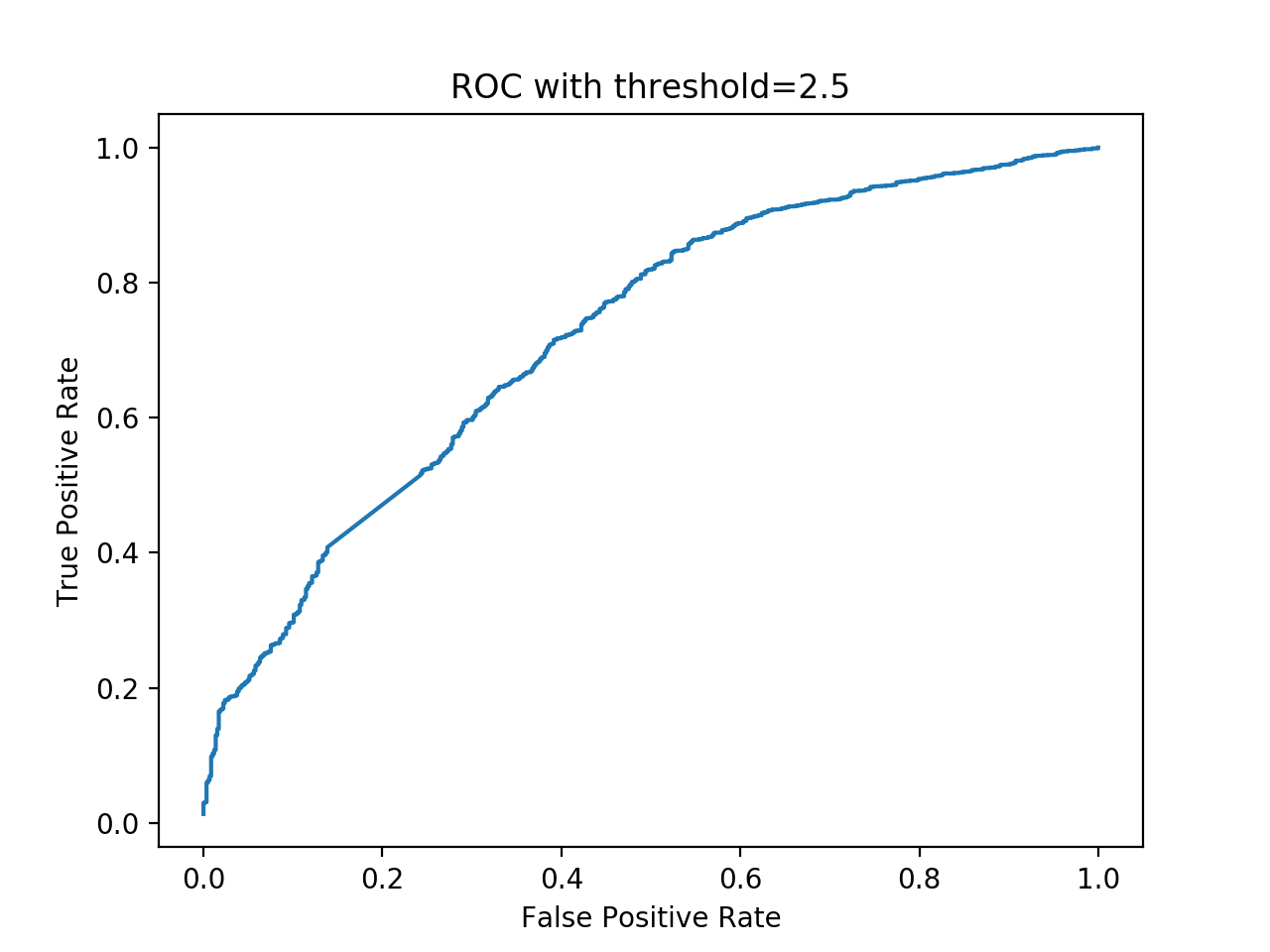


*figure 16 k – Average RMSE plot (high variance movie trimmed)*

The minimum RMSE average 1.37890711844.

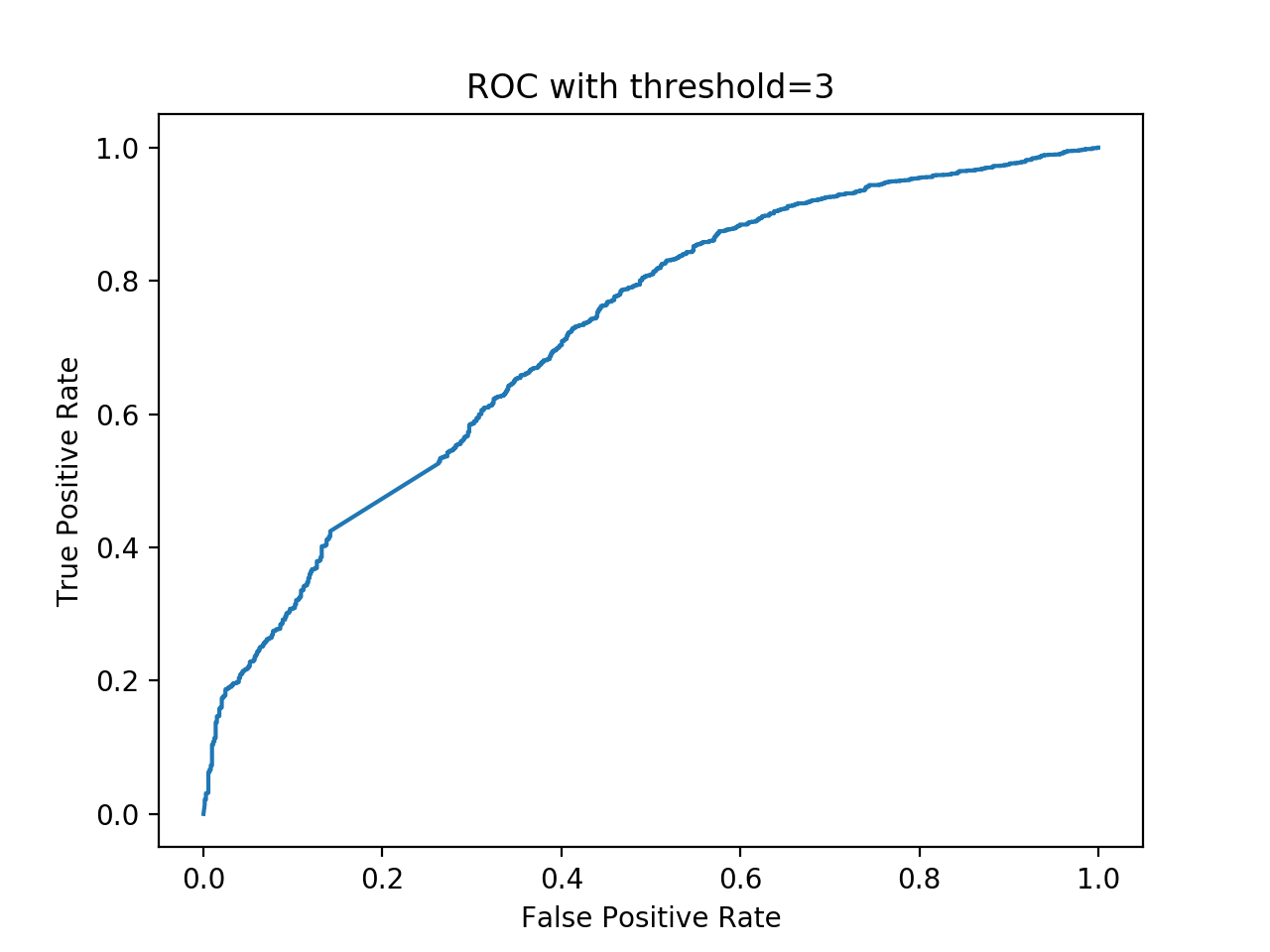
**Question 22**

NNMF-based collaborative filter:



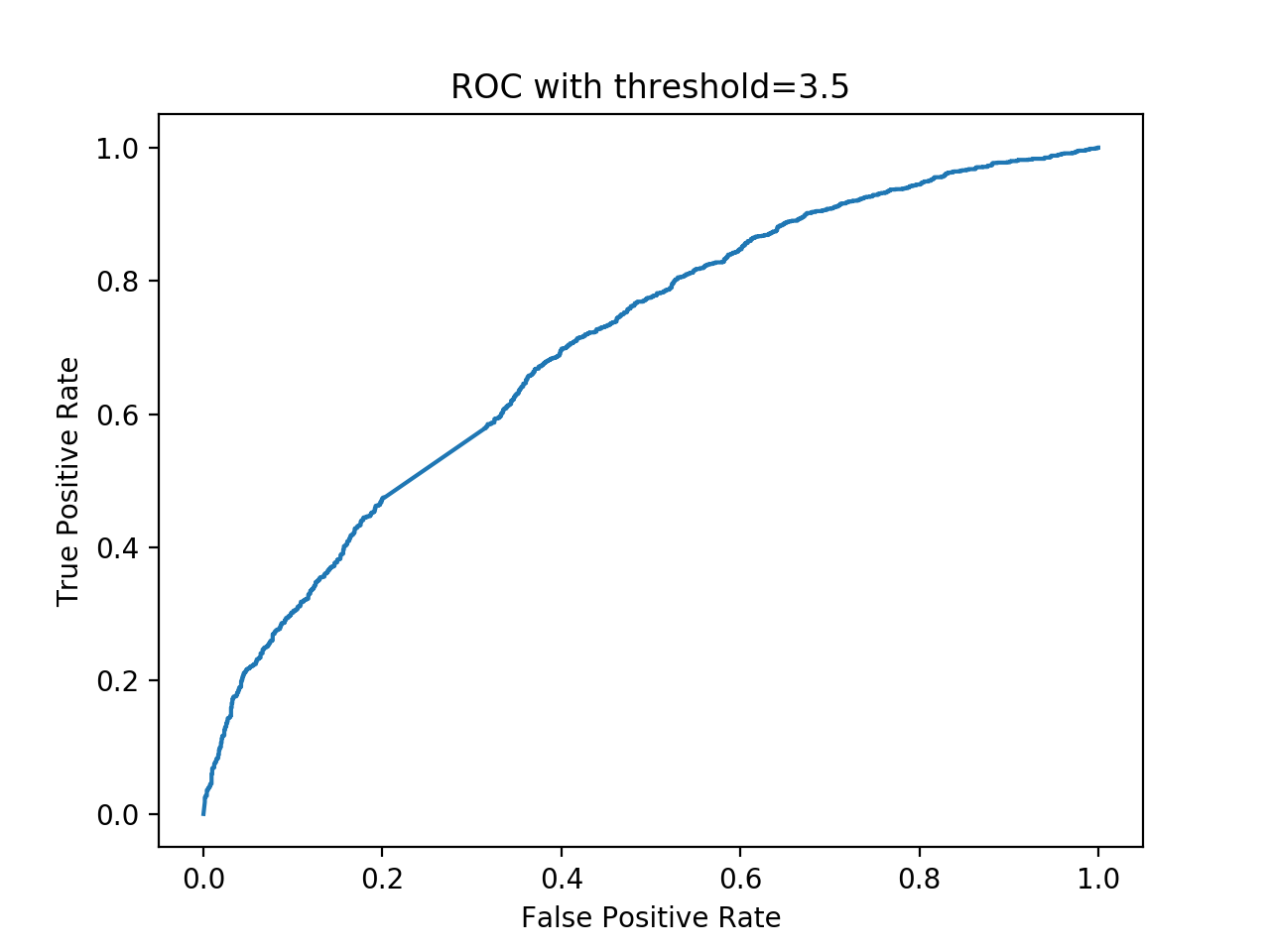
*figure 17 ROC with threshold = 2.5*

Area under curve value is 0.710183 with threshold=2.5



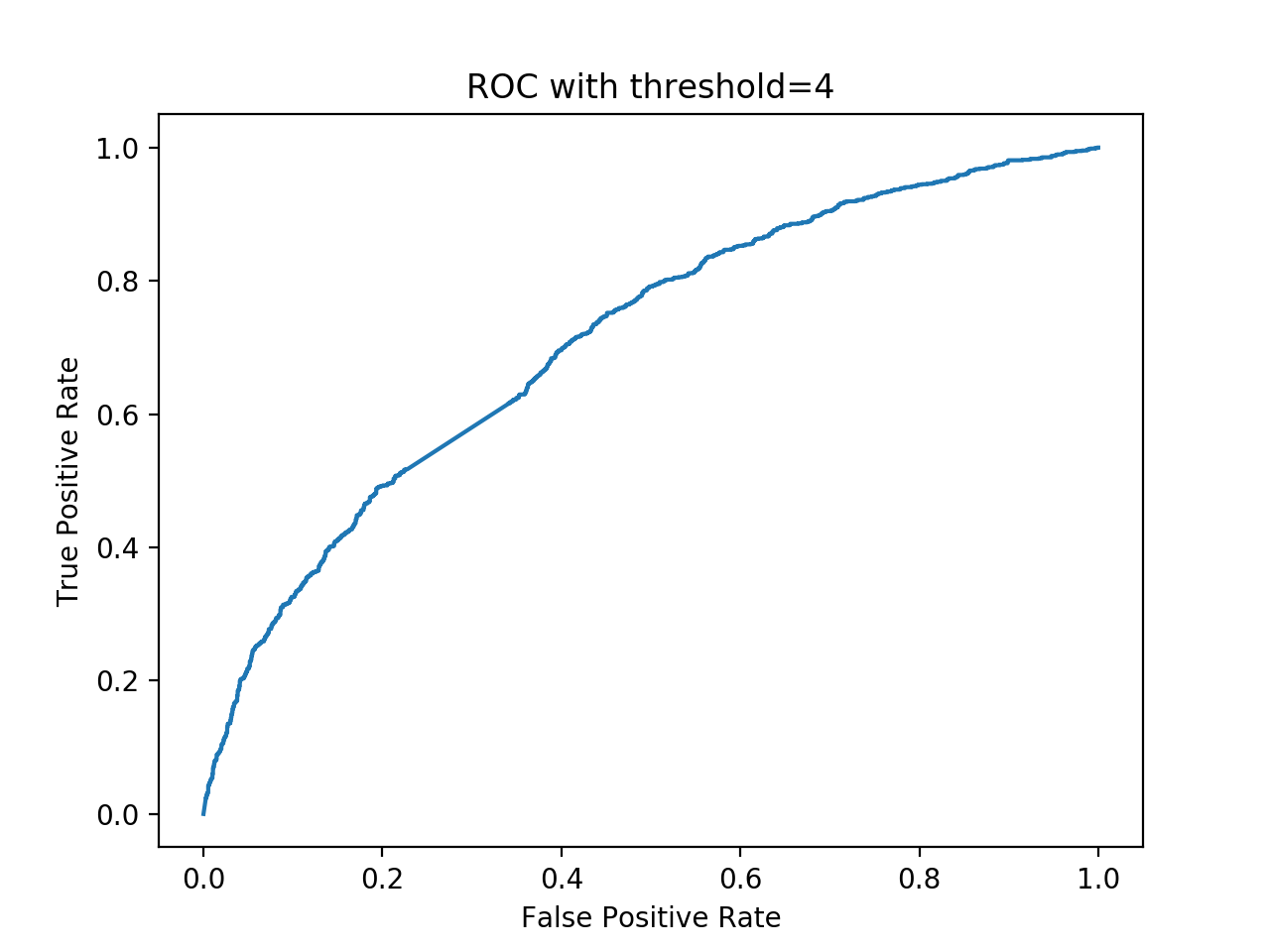
*figure 18 ROC with threshold = 3*

Area under curve value is 0.711119 with threshold=3



*figure 19 ROC with threshold = 3.5*

Area under curve value is 0.700139 with threshold=3.5



*figure 17 ROC with threshold = 4*

Area under curve value is 0.709567 with threshold=4

**Question 23**

After NMF factorization and sort the movies in descending order, the result seems to imply that each column represents a certain section of movies. First, almost very column’s top movies contain genre “drama”. We believe that this is because drama is a commonly-used tag for all the movies. Then, basically every group contains a certain topics, for example:

|  |  |
| --- | --- |
| **drama/comedy/thriller/horror** | |
| 91104 | Adventure|Drama|Fantasy|Romance |
| 6219 | Action|Drama|Thriller |
| 2287 | Horror|Sci-Fi|Thriller |
| 4520 | Comedy |
| 6450 | Drama|War |
| 704 | Action|Adventure |
| 1920 | Animation|Children|Fantasy|War |
| 4333 | Comedy|Crime |
| 1594 | Comedy|Drama |
| 2386 | Comedy|Drama |

*table 1 drama/comedy/thriller/horror*

|  |  |
| --- | --- |
| **action/sci-fi/adventure** | |
| 2285 | Drama |
| 1924 | Horror|Sci-Fi |
| 599 | Adventure|Western |
| 76293 | Action|Comedy|Romance |
| 2362 | Drama |
| 2365 | Action|Sci-Fi |
| 2041 | Action|Adventure|Children|Comedy |
| 7706 | Comedy|Musical |
| 3461 | Adventure|Drama|Thriller |
| 5269 | Drama |

*table 2 action/sci-fi/adventure*

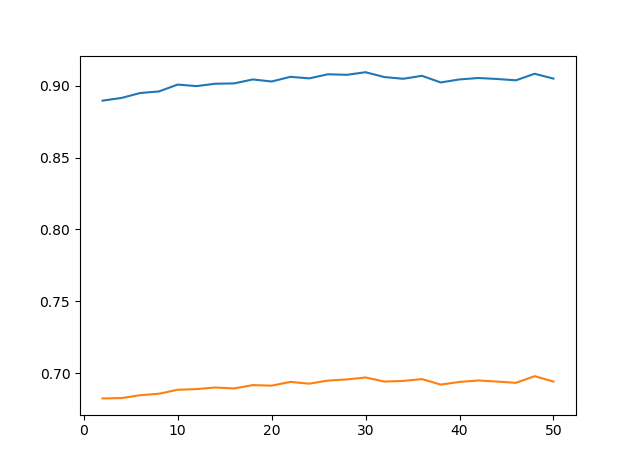
|  |  |
| --- | --- |
| **mystery/action** | |
| 72224 | Comedy |
| 190 | Thriller |
| 7505 | Drama|Horror|Mystery |
| 76175 | Action|Adventure|Drama|Fantasy |
| 4630 | Action |
| 5792 | Comedy|Drama |
| 78105 | Action|Adventure|Fantasy|Romance|IMAX |
| 8666 | Action|Crime|Fantasy |
| 3159 | Animation|Children|Musical|IMAX |
| 7369 | Action|Adventure|Children|Comedy|Mystery |

*table 3 mystery/action*

From the example above, we can see that each column represents some certain topics. However, the topic of each column is not very distinct to each other as we expect.

**Question 24**

In this problem, we designed a 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 blue line is k – average RMSE plot and the orange line is k – average MAE plot.



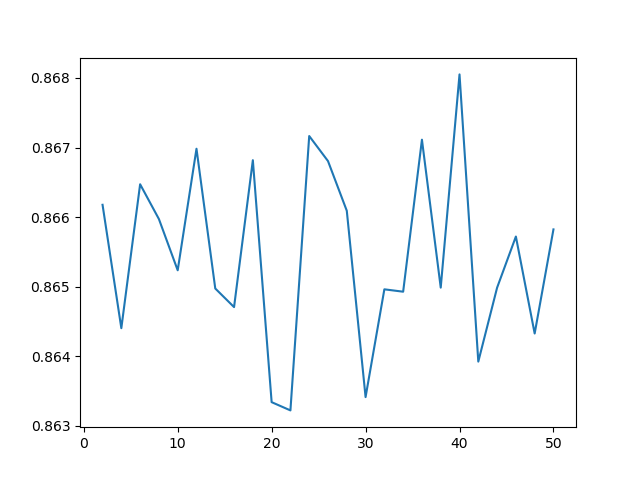
*figure 18 k – Average RMSE and k – Average MAE plot (MF with bias collaborative filter)*

**Question 25**

From figure 18, we are able to tell that minimum k should be 2.

**Question 26**

In this problem, we designed a 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.

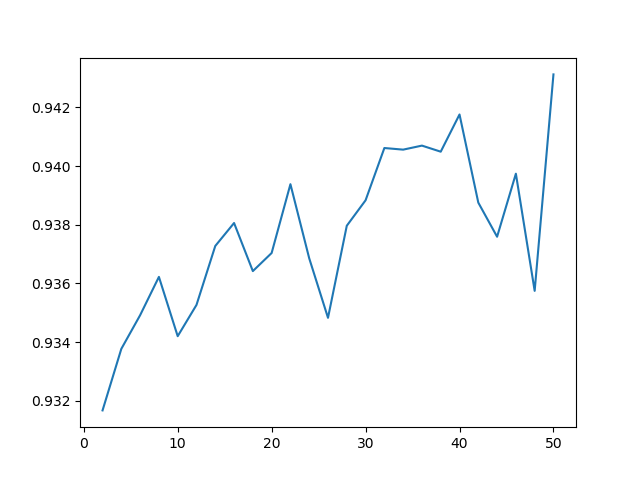


*figure 19 k – Average RMSE plot (popular movie trimmed)*

The minimum average RMSE is 0.863222696381.

**Question 27**

In this problem, we designed a MF with bias collaborative filter to predict the rat- ings of the movies in the unpopular movie trimmed test set and evaluated its performance using 10-fold cross validation.

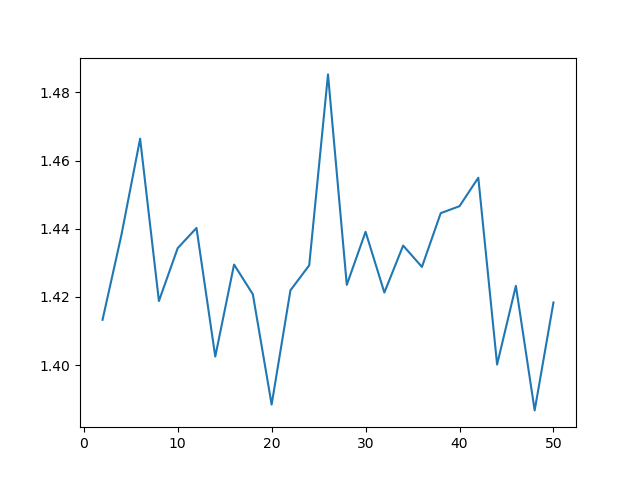


*figure 20 k – Average RMSE plot (unpopular movie trimmed)*

The minimum average RMSE is 0.93167517268.

**Question 28**

In this problem, we designed a MF with bias collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluated its performance using 10-fold cross validation.

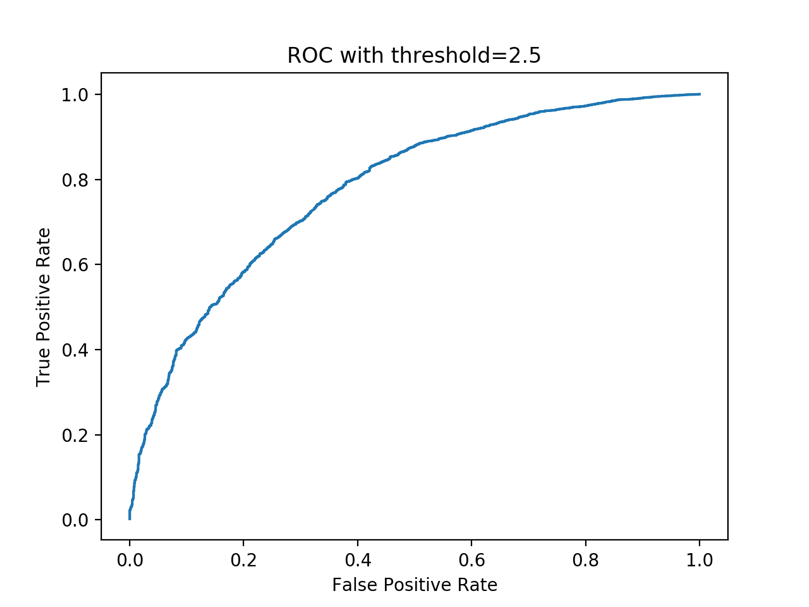


*figure 21 k – Average RMSE plot (high variance movie trimmed)*

The minimum average RMSE is 1.38669567917.

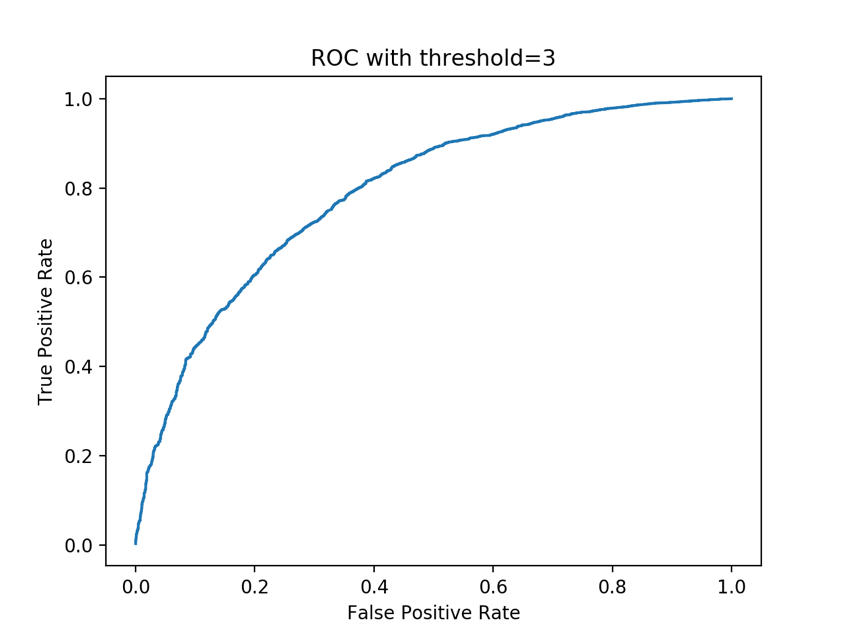
**Question 29**

MF with bias collaborative filter:



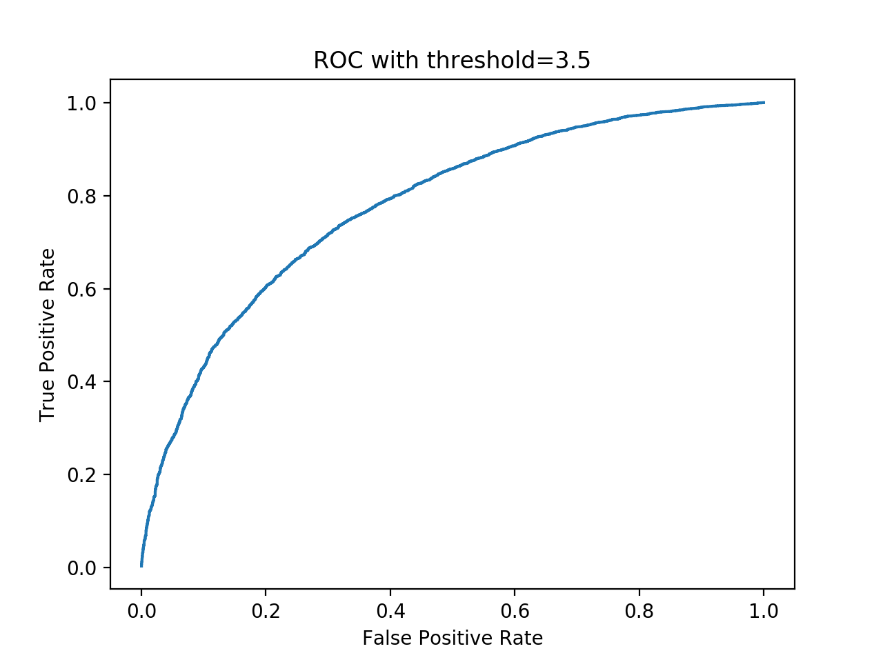
*figure 22 ROC with threshold = 2.5*

Area under curve value is 0.779444 with threshold=2.5.



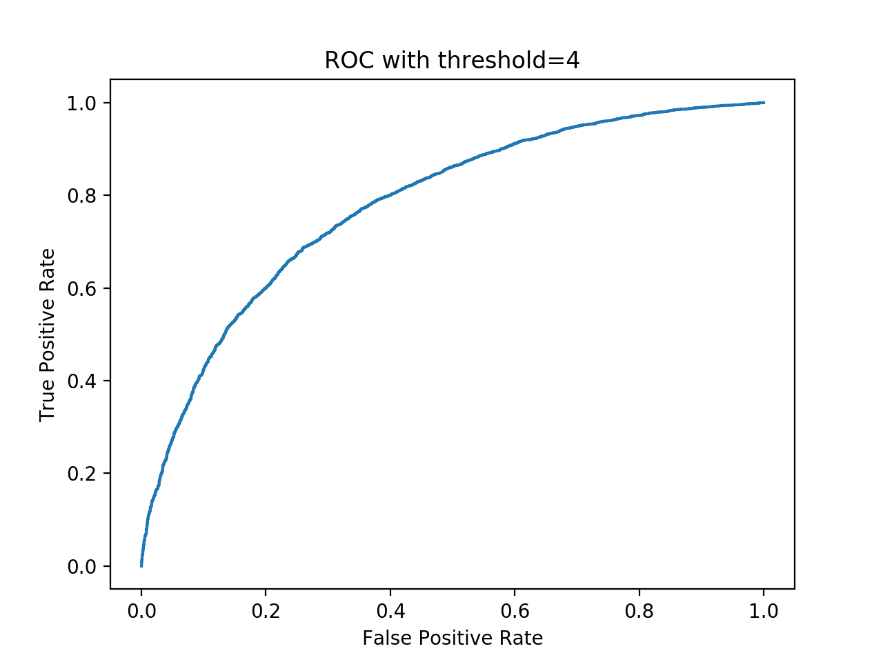
*figure 23 ROC with threshold = 3*

Area under curve value is 0.789535 with threshold=3.



*figure 24 ROC with threshold = 3.5*

Area under curve value is 0.778914 with threshold=3.5.



*figure 25 ROC with threshold = 4*

Area under curve value is 0.780036 with threshold=4.

**Naïve collaborative filtering**

In this part of the project, we implemented a naive collaborative filter to predict the ratings of the movies in the MovieLens dataset, which returns the mean rating of the user as it’s predicted rating for an item.

**Question 30**

In this problem, we designed a naive collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated its performance using 10-fold cross validation. The average RMSE is 3.65819094585.

**Question 31**

In this problem, we designed a naive 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 average RMSE is 3.3367489615.

**Question 32**

In this problem, we designed a naive 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 average RMSE is 3.46278215666.

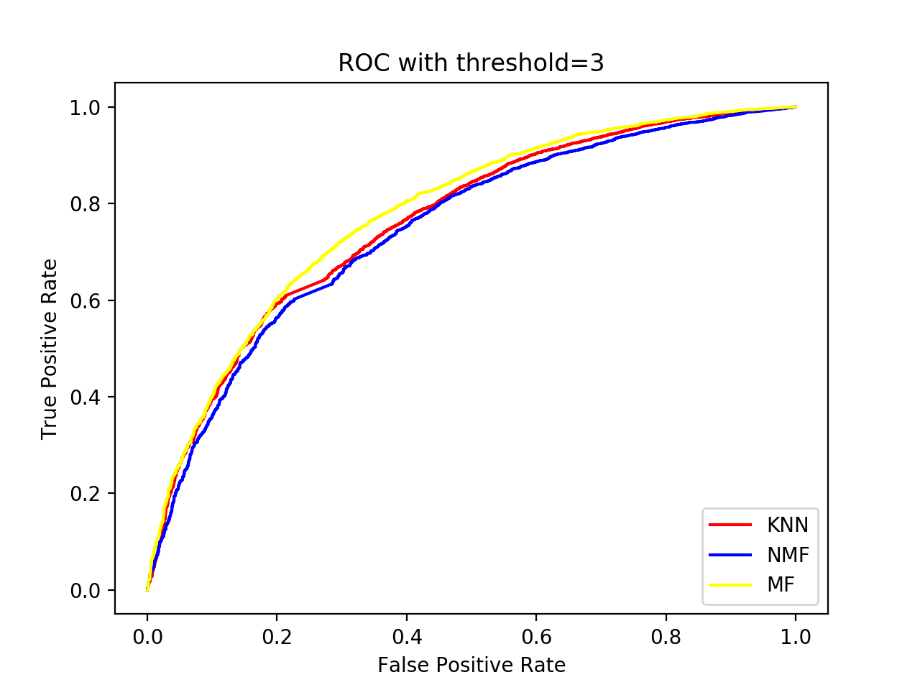
**Question 33**

In this problem, we designed a naive collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluated its performance using 10-fold cross validation. The average RMSE is 2.90080102758.

**Performance comparison**

In this section, we compared the performance of the k-NN, NNMF and MF with collaborative filters predicting the ratings of the movies in the MovieLens dataset.

**Question 34**

****

*figure 26 ROC with threshold = 3 (k-NN, NNMF, MF)*

From figure 26, we can find out that MF with bias based collaborative filter has the best performance among three collaborative filter.

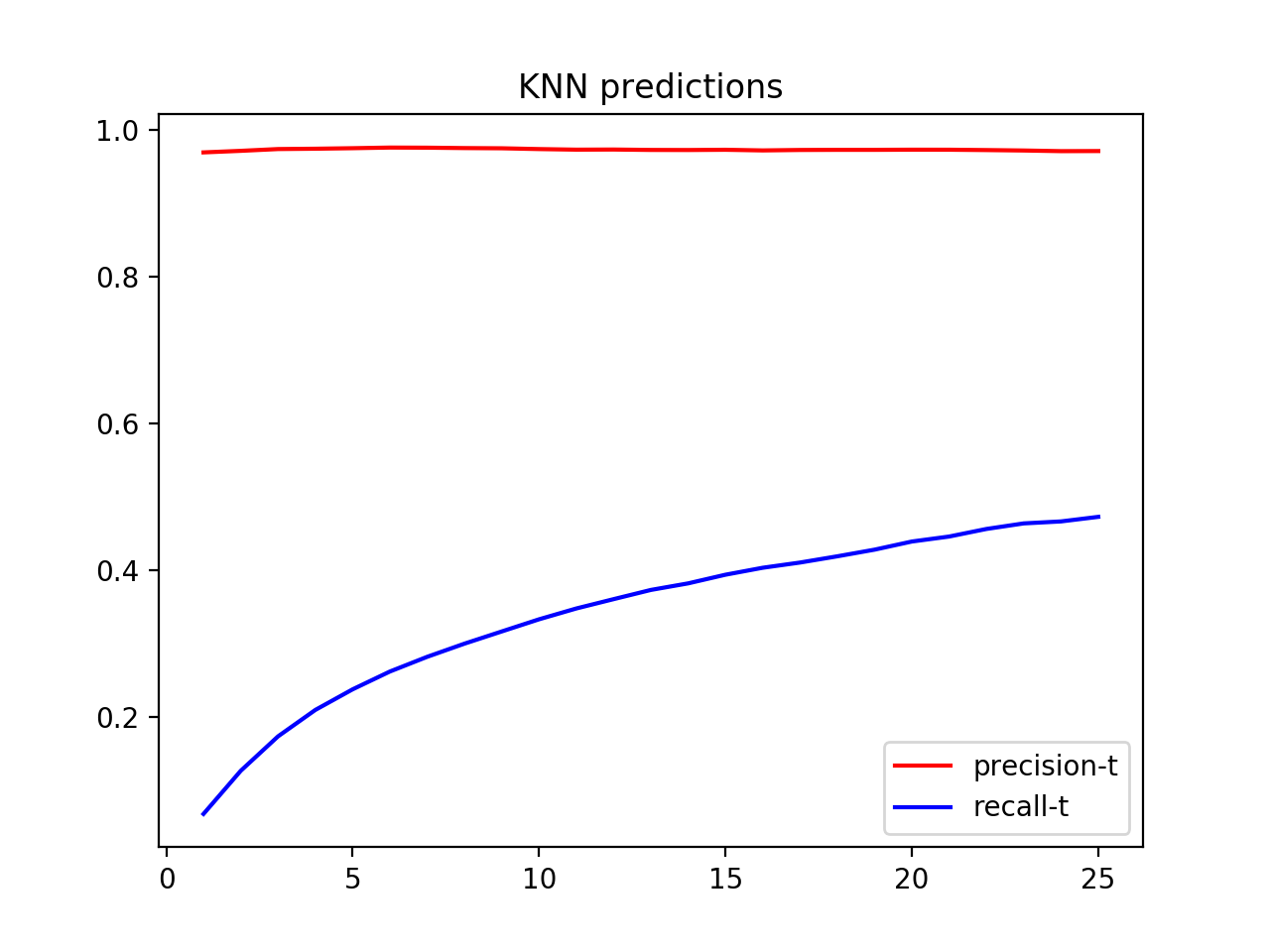
**Ranking**

In this part, we implemented techniques for solving the ranking version of the problem. We solved the prediction problem and then ranked the predictions.

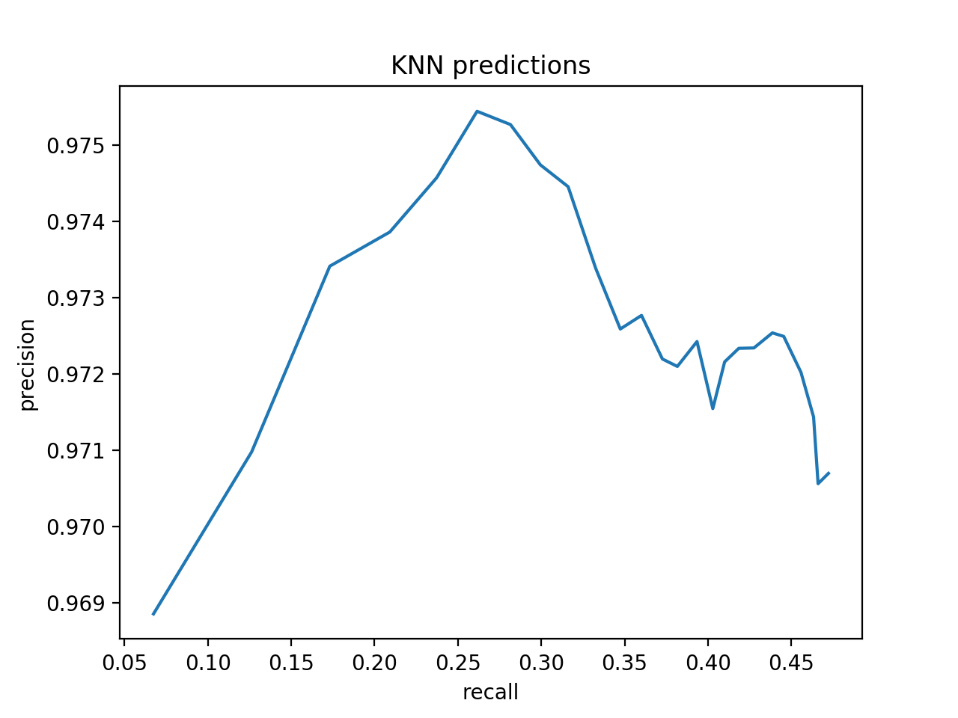
**Question 35**

Precision is referring to the accuracy of the results from our recommended system, which is the proportion of user’s favorite movies; recall is referring to the proportion of user’s favorite movies in the recommended movie results.

**Question 36**

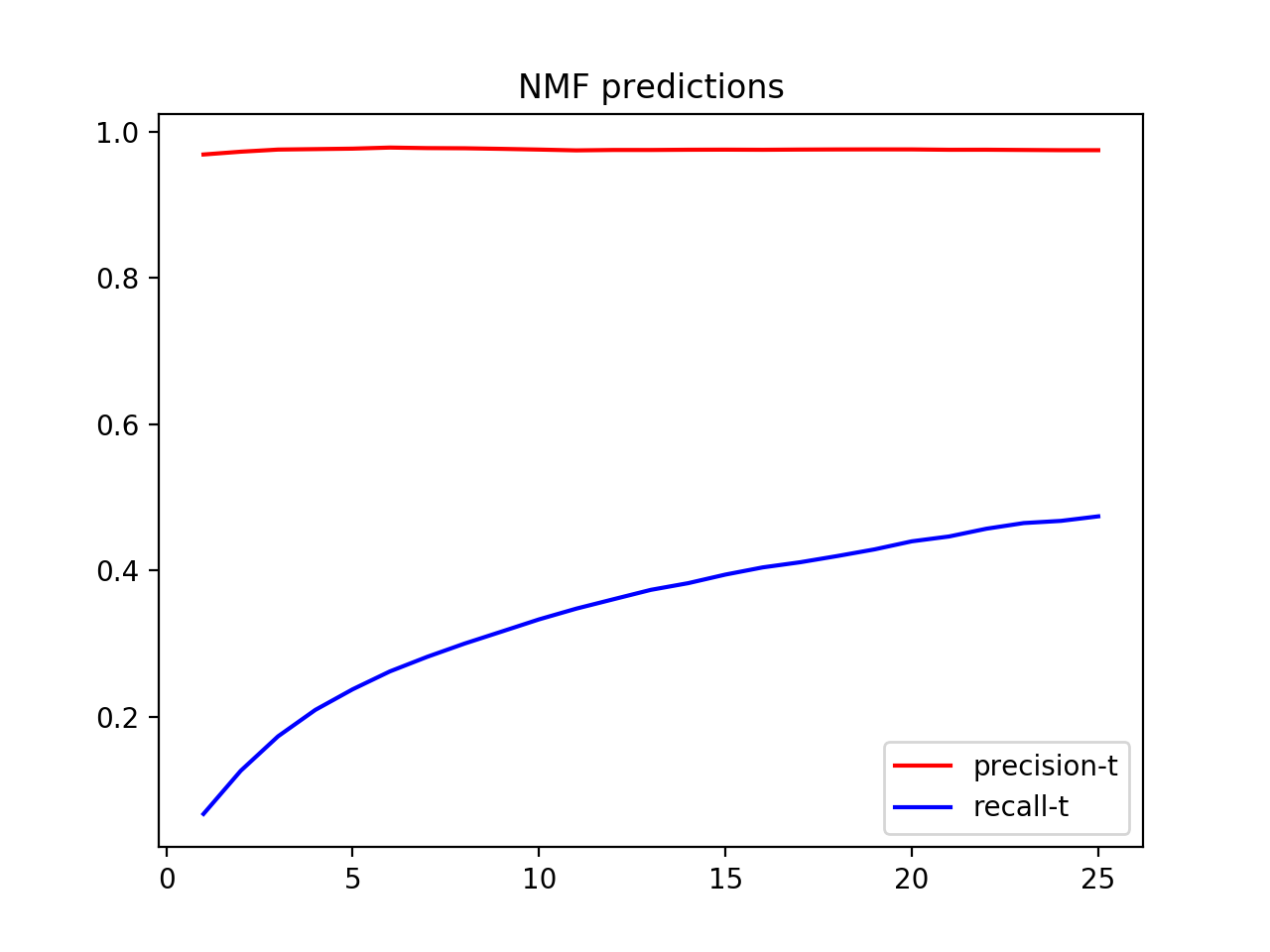
****

*figure 27 t – average precision and t – average recall plot (k-NN prediction)*

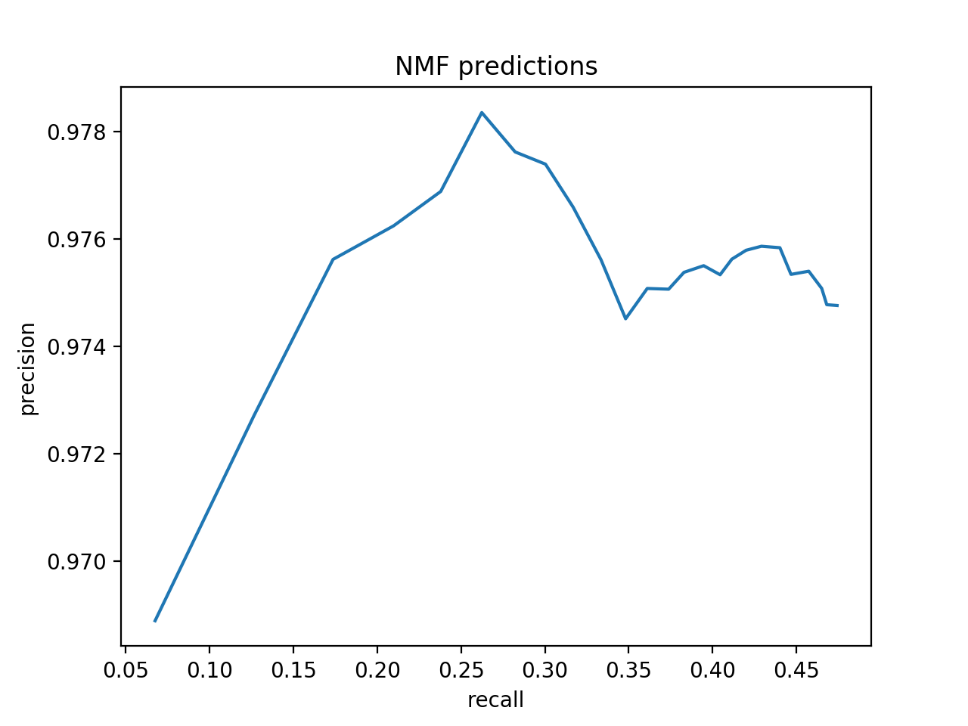
****

*figure 28 average recall – average precision plot (k-NN prediction)*

**Question 37**

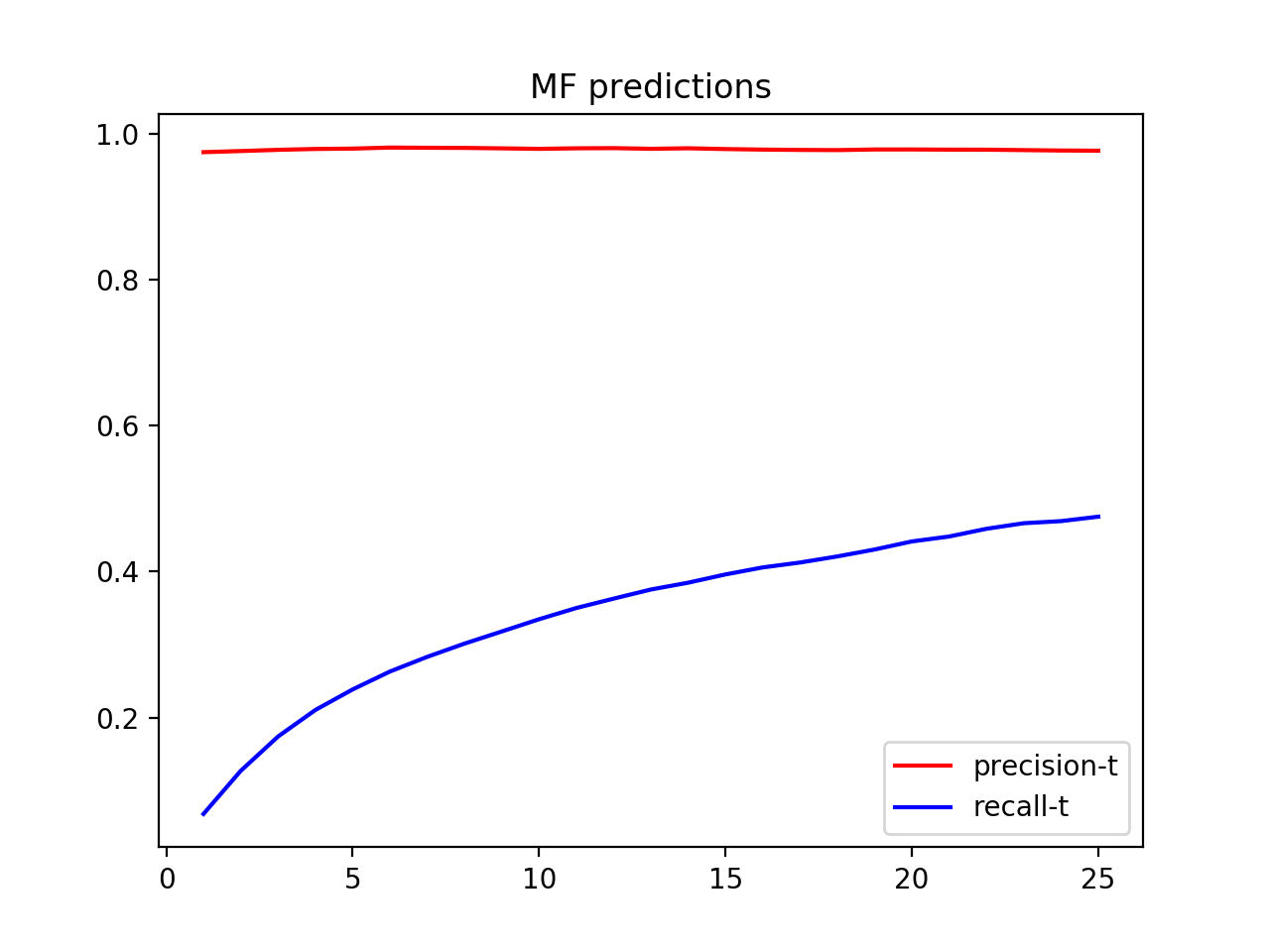
**

*figure 29 t – average precision and t – average recall plot (NNMF prediction)*

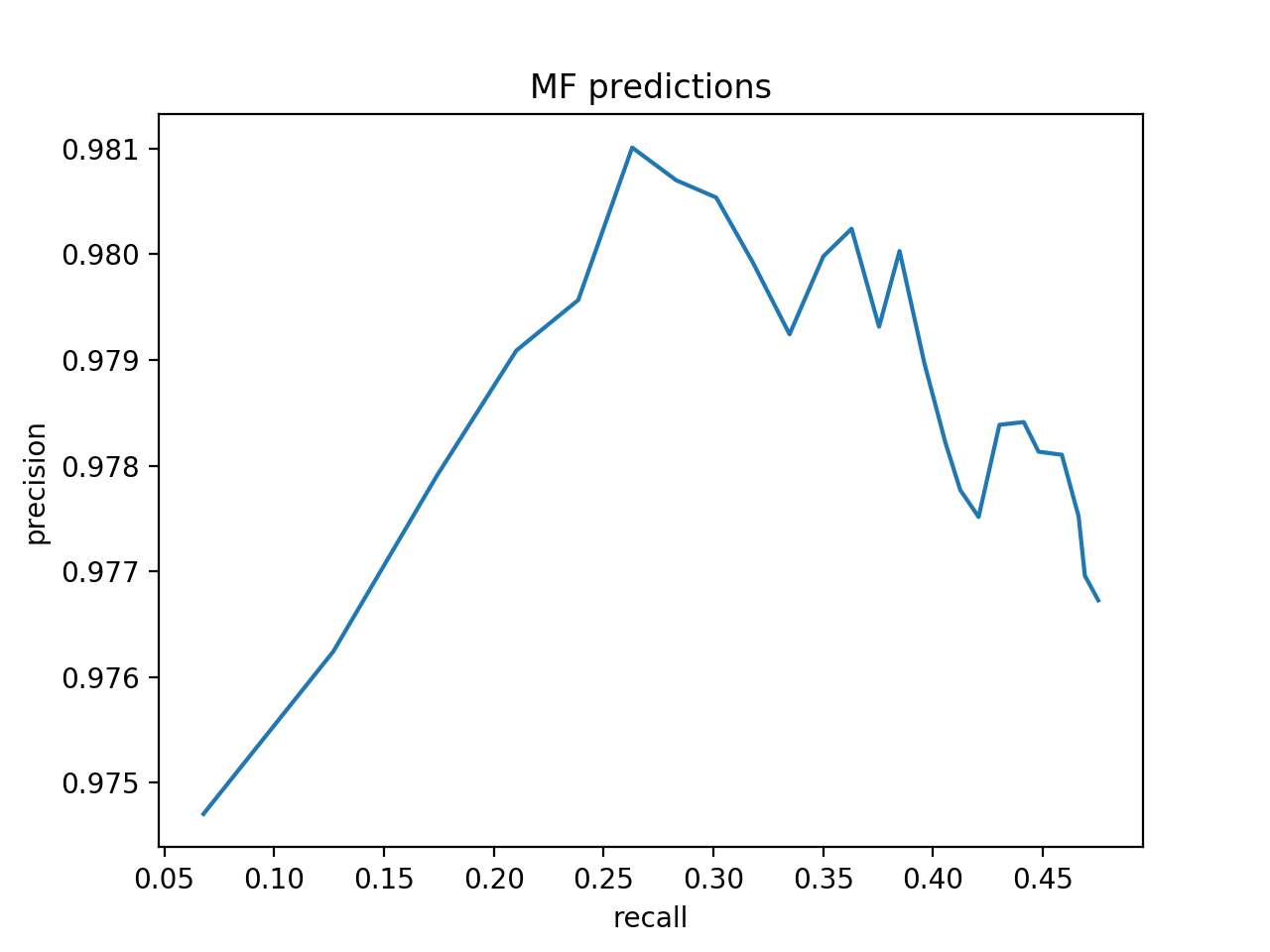
**

*figure 30 average recall – average precision plot (NNMF prediction)*

**Question 38**

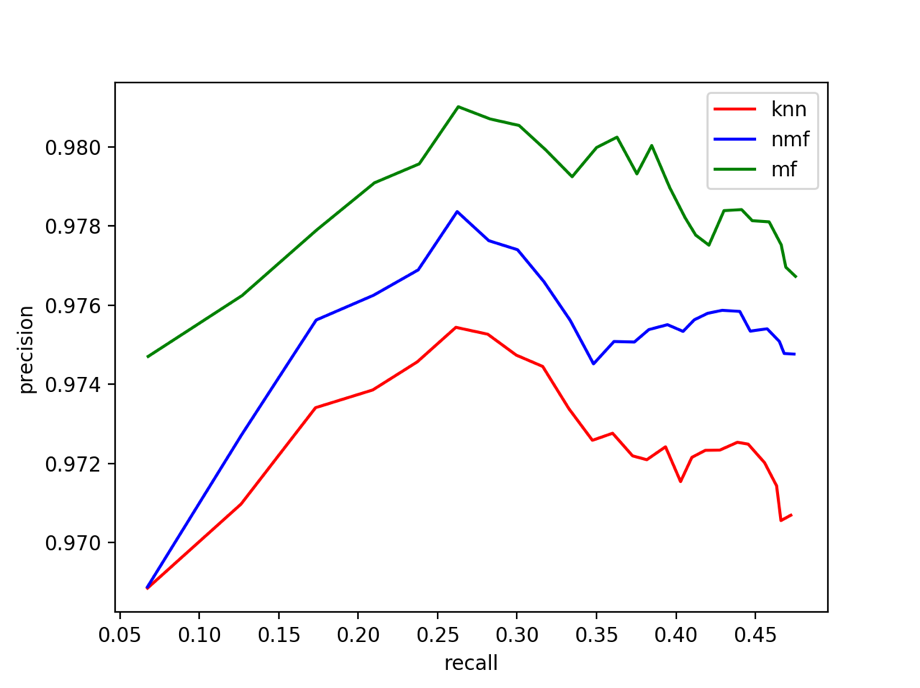
**

*figure 31 t – average precision and t – average recall plot (MF prediction)*

**

*figure 32 average recall – average precision plot (MF prediction)*

**Question 39**



*figure 33 precision – recall curve (k-NN, NNMF, MF with bias)*