ECE 219 Project 3: Collaborative Filtering

Chuang Yu (305629107), Hongyi Chen (705186099), Zichun Chai (505625207)

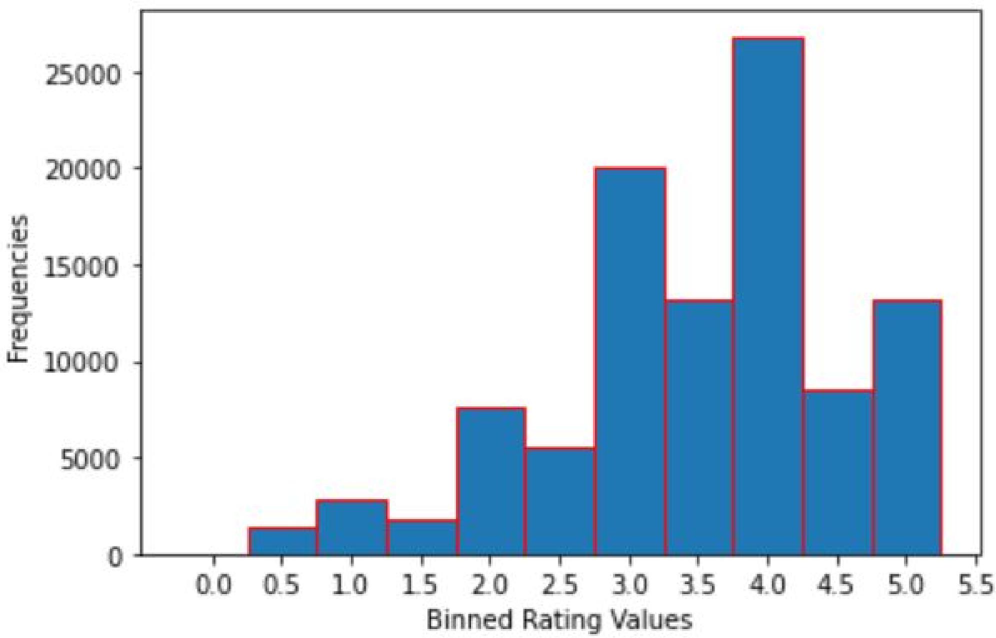
Question 1:

In this project, we built a recommendation system which can predict the ratings of the movies in the MovieLens dataset which we have downloaded from the provided website. We used the movie rating information only for this project. We assumes that the ratings matrix is denoted by R, and it is an m * 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 rij . Before moving on to the collaborative filter implementation, we first analyzed and visualized some properties of this dataset.

Firstly, we computed the sparsity of the movie rating dataset, and the sparsity is defined by total number of available ratings(rr) / total number of possible ratings(m * n). We imported the dataset, got the number of unique values in userId m = 610, movieId n = 9724, and rating rr = 100836, and did the computation according to the formula. As a result, the sparsity equaled to 0.016999683055613623. Sparsity is used to measure how sparse the matrix is and the sparsity value we got is low and it denotes that much movie rating information is not provided.

Question 2:

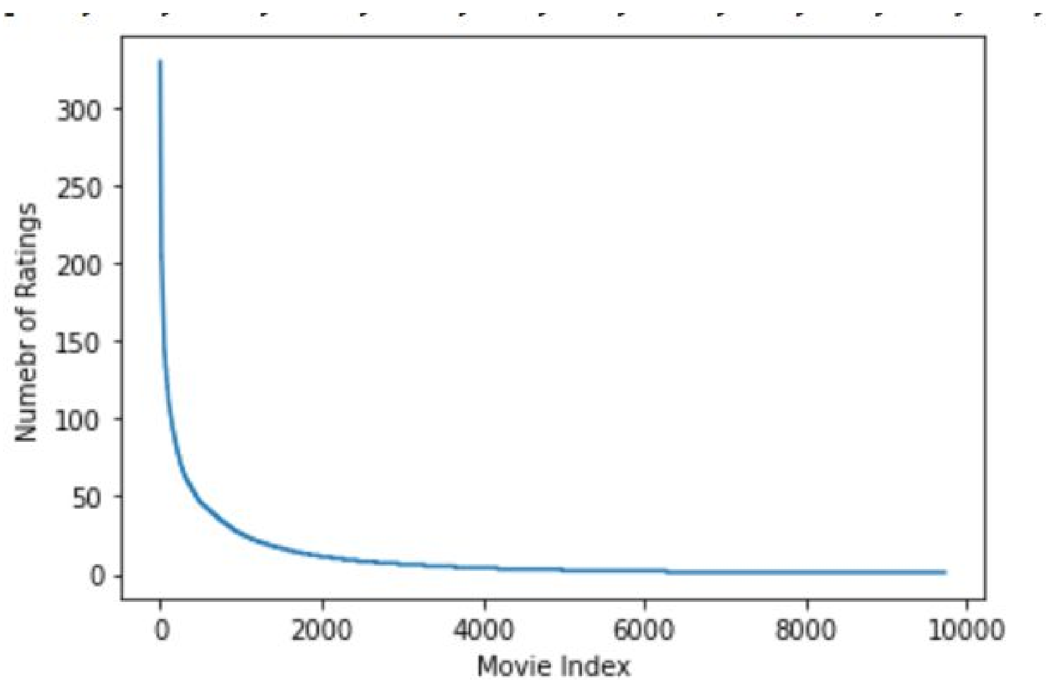
In this problem, we plotted a histogram showing the frequency of the rating values. Specifically, we binned the rating values into intervals of width 0.5 and used those binned rating values as the x-axis. And then we counted the number of entries in the rating matrix R with rating values in the binned intervals and used this count as the y-axis. With the values of both x (binned rating values) and y (Frequencies) axis, we were able to plot the required histogram, which is attached below:



From the histogram we could see that it was not a symmetric distribution because it was skewed to the left, and the rating values with great frequencies were between 3.0 to 4.5. Rating of 4 appeared more than 25,000 times. And there were so few low ratings (0.5 to 2.0) maybe because people liked to write positive feedback rather than negative feedback.

Question 3:

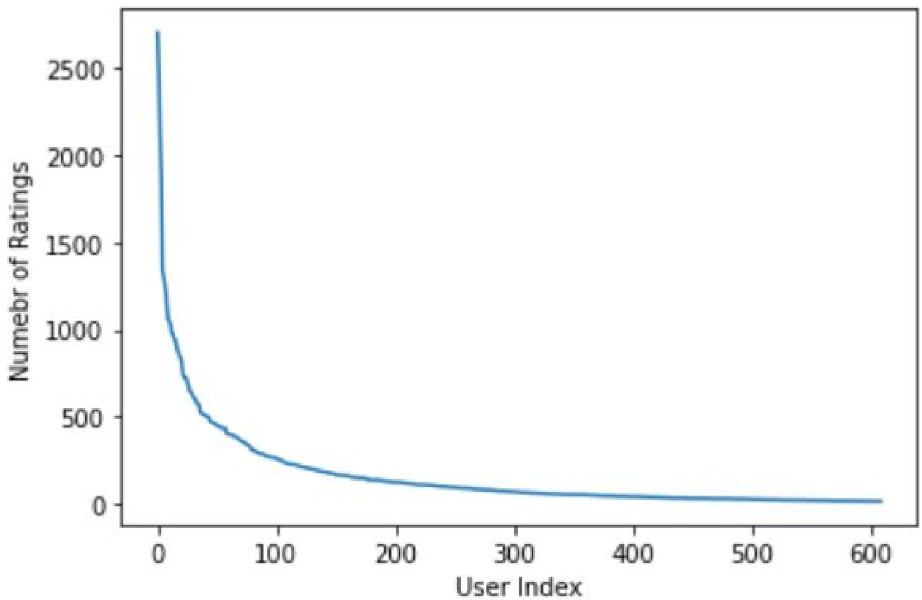
In the third problem, we were asked to plot the distribution of the number of ratings received among movies. The x-axis was the movie index ordered by decreasing frequency and the y-axis was the number of ratings the movie has received. Specifically, we used a counter for counting the elements in movieId, and sorted the values of the counting dictionary in descending order. A monotonically decreasing curve instead of a histogram was expected and the curve is shown below:



This curve showed to us that most of the movies did not receive more than 50 ratings, whereas the most popular ones received more than 100 ratings. And we did not consider those movies that did not receive a single rating at all.

Question 4:

We plotted the distribution of ratings among users by letting the x-axis be the user index ordered by decreasing frequency and letting the y-axis be the number of movies the users have rated. Specifically, we used a counter for counting the elements in userId, and sorted the values of the counting dictionary in descending order. A monotonically decreasing curve instead of a histogram was expected and the curve is shown below:



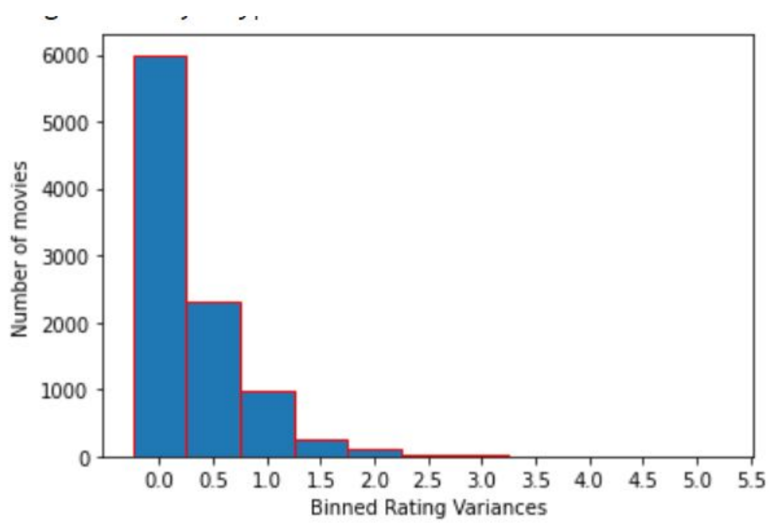
From the curve we could see that most of the users gave ratings less than 500, whereas only few dedicated users gave more than 1000 ratings.

Question 5:

From Question 3 we observed that most of the movies did not receive more than 50 ratings, whereas the most popular ones received more than 100 ratings. It is reasonable since popular movies are often watched and rated by a large number of users and unpopular movies are seldomly watched or rated. Moreover, there were only a few popular movies and most of the movies were unpopular and received few ratings. As a result, the recommendation process is more likely to recommend popular movies rather than unpopular ones to other users. And we need to use inter-user correlation instead of inter-item correlation to predict the rating of a user toward a specific movie.

Question 6:

We computed the variance of the rating values received by each movie, and then binned the variance values into intervals of width 0.5 and used these binned variance values as the x-axis. We then counted the number of movies with variance values in the binned intervals and used this count as the y-axis. With the values of both x (binned rating variances) and y (Number of movies) axis, we were able to plot the required histogram, which is attached below:



As binned rating variances increased, the number of movies decreased. Most of the movies received consistent ratings. Nearly 6000 movies received ratings with nearly no difference. And nearly no movies received ratings that differed more than 2.5. Therefore, we could tell that the users had consistent tastes.

Question 7:

Pearson-correlation coefficient between users u and v, denoted by Pearson(u,v), captures the similarity between the rating vectors of users u and v. Before stating the formula for computing Pearson(u,v), let's first introduce some notation:

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

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

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

ruk: Rating of user u for item k

The formula is shown below:

$$\mu_u = \frac{1}{|I_u|} \sum_{k \in I_u} r_{uk}$$

Question 8:

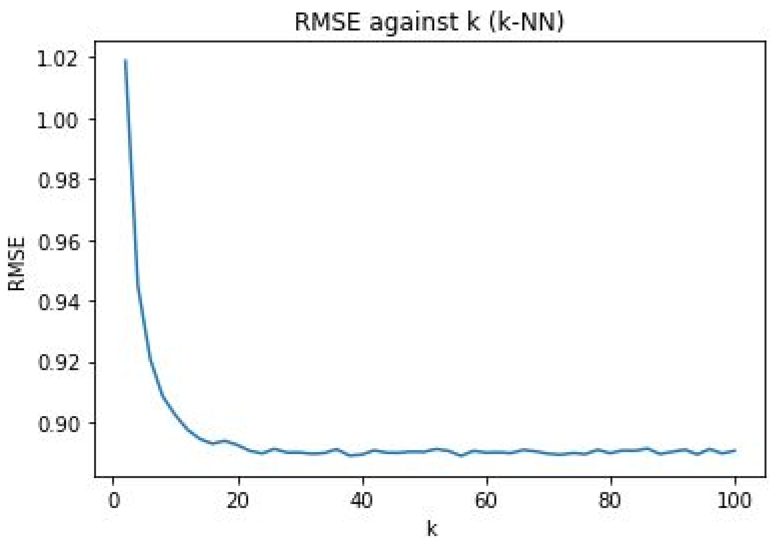
It means the set of item indices of those ratings that are given by both user u and user v because as mentioned above, Iu denotes the set of item indices for which ratings have been specified by user u, and Iv denotes the set of item indices for which ratings have been specified by user v. Its value can be an empty set since our rating matrix R is sparse and the empty set happens when a movie is not rated by both user u and user v.

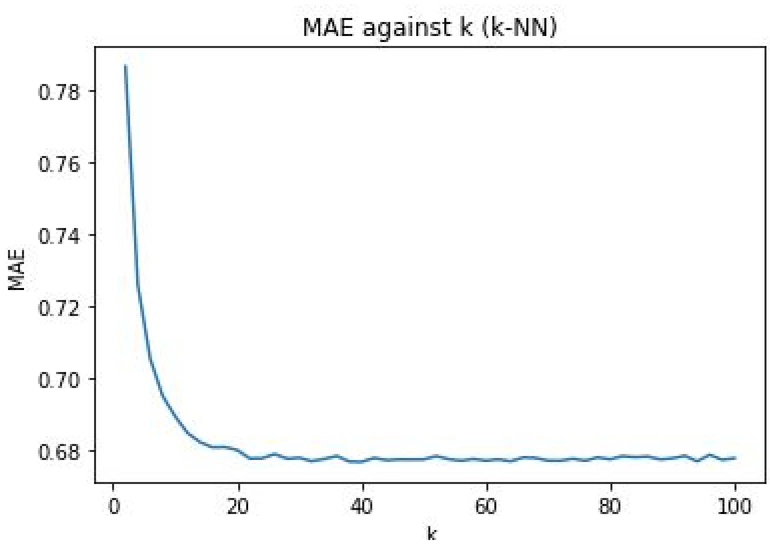
Question 9:

Some users are lenient and they rate all items highly and some users are strict and they rate all items poorly, and this will lead to inaccurate predictions of our recommendation system. The reason behind mean-centering the raw ratings in the prediction function is that it helps to remove the biases in different users' rating preferences (both overly strict and overly lenient), and therefore balance the ratings.

Question 10:

The previous problems gave us the basics needed to implement a k-NN collaborative filter for predicting the ratings of the movies. And in this part, we designed a k-NN collaborative filter and tested the performance via 10-fold cross validation. Specifically, we designed a k-NN collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated the performance using 10-fold cross validation. We swept k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k computed the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. The plotted average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis) are attached below:



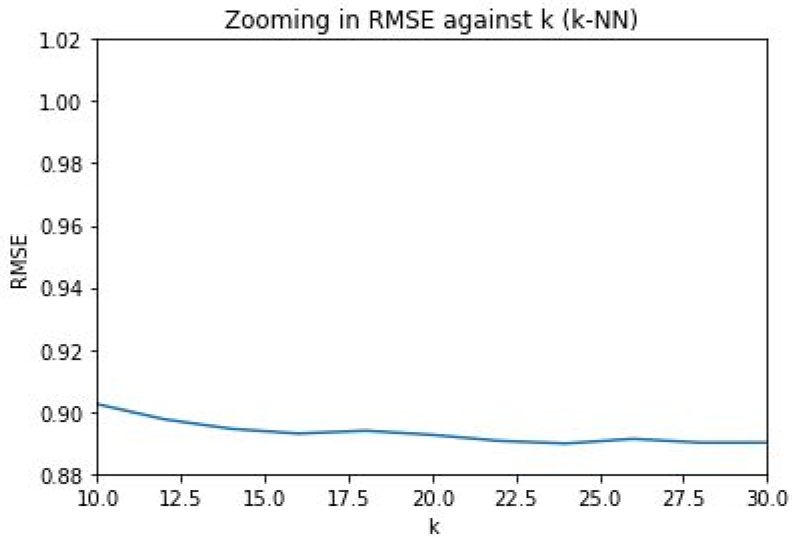


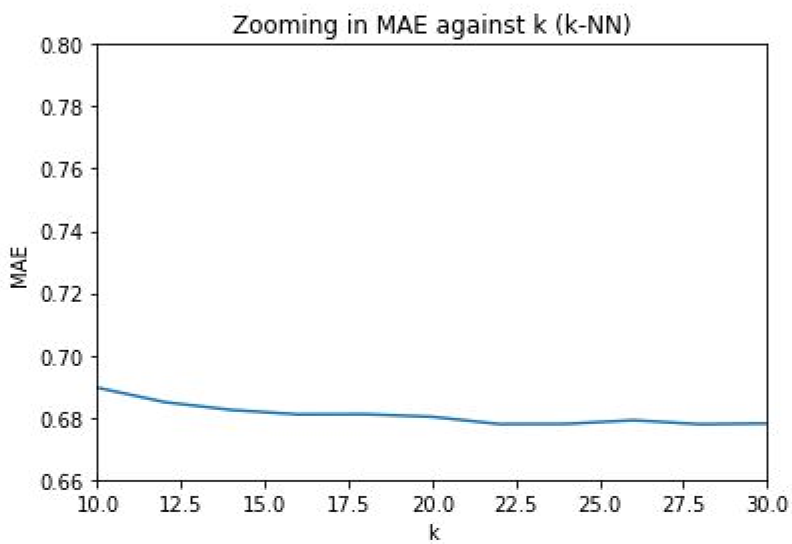
Question 11:

We used the plot generated from Question 10 to find a minimum k that increasing the k value above the minimum value would not result in a significant decrease in average RMSE or average MAE, converging to a steady-state value. Firstly, we printed out all the k values and its corresponding RMSE and MAE values that are shown below:

k = 2 RMSE = 1.018753961341217 MAE = 0.7865028222248267
k = 4 RMSE = 0.9448086249946543 MAE = 0.7259519486325948
k = 6 RMSE = 0.9207233997151036 MAE = 0.7054250884059318
k = 8 RMSE = 0.9084954853222026 MAE = 0.6951668018792503
k = 10 RMSE = 0.9023884854311323 MAE = 0.6895454030600743
k = 12 RMSE = 0.8975122229355617 MAE = 0.6849251534338253
k = 14 RMSE = 0.8944474998510856 MAE = 0.682379624977852
k = 16 RMSE = 0.8929118299153922 MAE = 0.6809633417421355
k = 18 RMSE = 0.893843384723963 MAE = 0.6810158716998428
k = 20 RMSE = 0.8925001229924673 MAE = 0.6801795164406298
k = 22 RMSE = 0.890570203654071 MAE = 0.6778478008600152
k = 24 RMSE = 0.8897145215133359 MAE = 0.6778889120287827
k = 26 RMSE = 0.8912116941931633 MAE = 0.6790774034295584
k = 28 RMSE = 0.8900083667921012 MAE = 0.6777989523549764
k = 30 RMSE = 0.8900234929774886 MAE = 0.678030902430663
k = 32 RMSE = 0.8895831875157405 MAE = 0.6771166355639057
k = 34 RMSE = 0.889829660777621 MAE = 0.6777286722905804
k = 36 RMSE = 0.8910983228687572 MAE = 0.6785410790913022
k = 38 RMSE = 0.8890047561776628 MAE = 0.6770071350126081
k = 40 RMSE = 0.8892959156041457 MAE = 0.676928001752545
k = 42 RMSE = 0.8907241188043347 MAE = 0.6780041314749649
k = 44 RMSE = 0.8899470193843735 MAE = 0.6773829650051091
k = 46 RMSE = 0.889917612973951 MAE = 0.677595342856318
k = 48 RMSE = 0.8902802785347068 MAE = 0.6775519769433256
k = 50 RMSE = 0.8902041109476742 MAE = 0.6776087973474335
k = 52 RMSE = 0.891164928127651 MAE = 0.678524296938475
k = 54 RMSE = 0.8905291581238333 MAE = 0.6776767439499214
k = 56 RMSE = 0.8888914448457557 MAE = 0.6772347785718402
k = 58 RMSE = 0.8905708080089013 MAE = 0.6777273606993333
k = 60 RMSE = 0.8899542285991935 MAE = 0.6772704168603493
k = 62 RMSE = 0.8900929698596002 MAE = 0.6776046172886308
k = 64 RMSE = 0.8897848031977134 MAE = 0.6770946013143834
k = 66 RMSE = 0.8908640437911833 MAE = 0.6781536115235517
k = 68 RMSE = 0.8903840509033643 MAE = 0.6780090438377211
k = 70 RMSE = 0.8896229451456389 MAE = 0.6772641260695488
k = 72 RMSE = 0.8892993325590396 MAE = 0.6772403193838372
k = 74 RMSE = 0.8898425262201423 MAE = 0.6778082529358941
k = 76 RMSE = 0.8894842484056337 MAE = 0.6772568195531957
k = 78 RMSE = 0.890917314545751 MAE = 0.678160117614345
k = 80 RMSE = 0.8897615910720157 MAE = 0.6776286288829969
k = 82 RMSE = 0.8907141166668326 MAE = 0.6785154315877768
k = 84 RMSE = 0.8905891936963715 MAE = 0.6782159707223102
k = 86 RMSE = 0.8913370887418436 MAE = 0.6784208581187816
k = 88 RMSE = 0.889512649722539 MAE = 0.6776078059224167
k = 90 RMSE = 0.8902621339538213 MAE = 0.6778811035944311
k = 92 RMSE = 0.8909430192951502 MAE = 0.6786184548314951
k = 94 RMSE = 0.8893647125826181 MAE = 0.6770748162097371
k = 96 RMSE = 0.8911512803236711 MAE = 0.6788802670824323
k = 98 RMSE = 0.8896868126060614 MAE = 0.677507294944002
k = 100 RMSE = 0.8905795344494619 MAE = 0.677927742706341

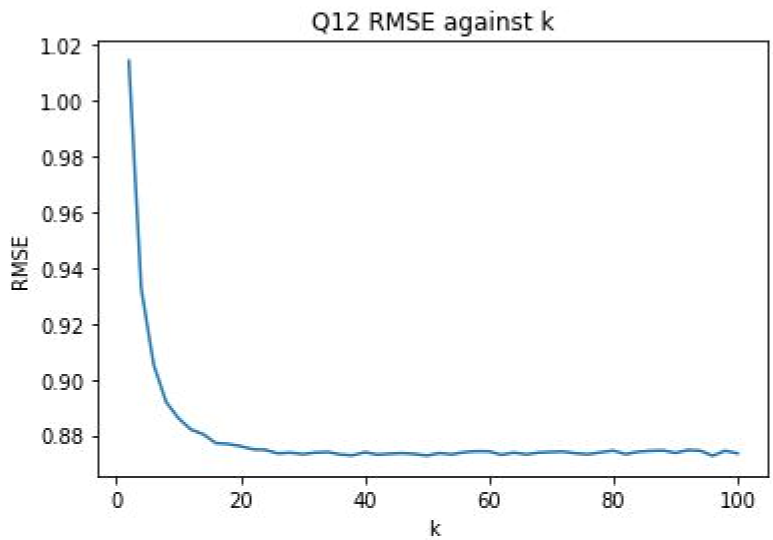
After that, we zoomed in the RMSE and MAE plots generated in Question 10. And from the plots, we could see that k = 22 gave us the steady-state.

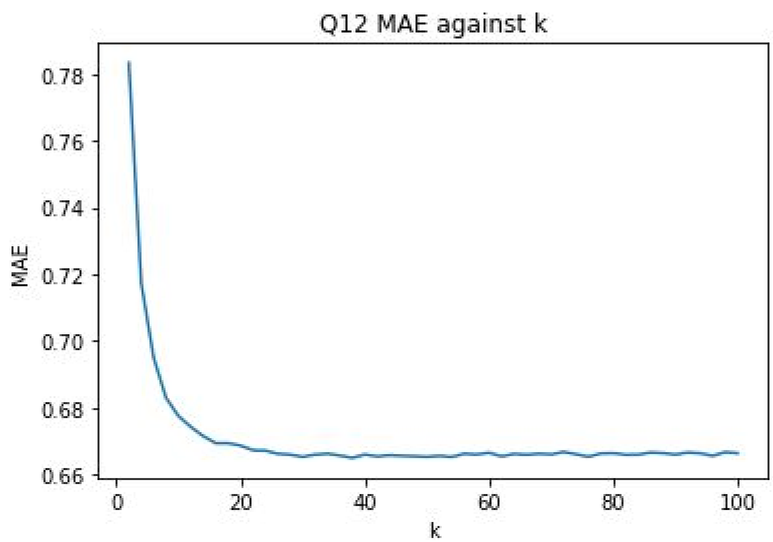




Question 12:

We analyzed the performance of the k-NN collaborative filter in predicting the ratings of the movies in the trimmed test set. There were 3 trimmings that we implemented which were popular movie trimming, unpopular movie trimming, adn high variance move trimming. Having coded the functions of these 3 trimmings, we designed a k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

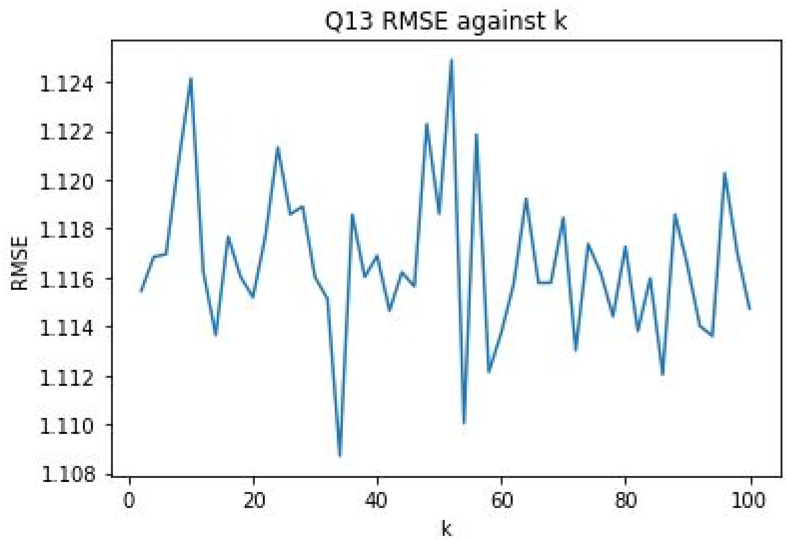


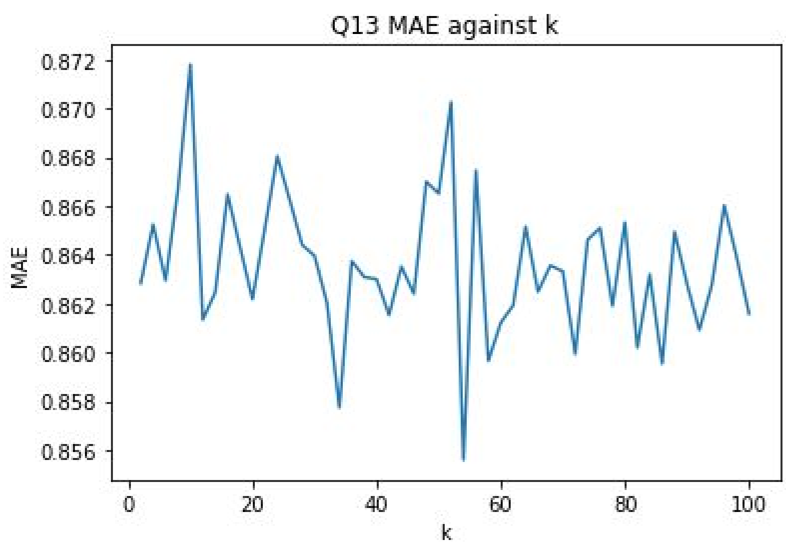



```
Q12 The minimum average RMSE = 0.8726399316549381
```

Question 13:

Following the same steps, we designed a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

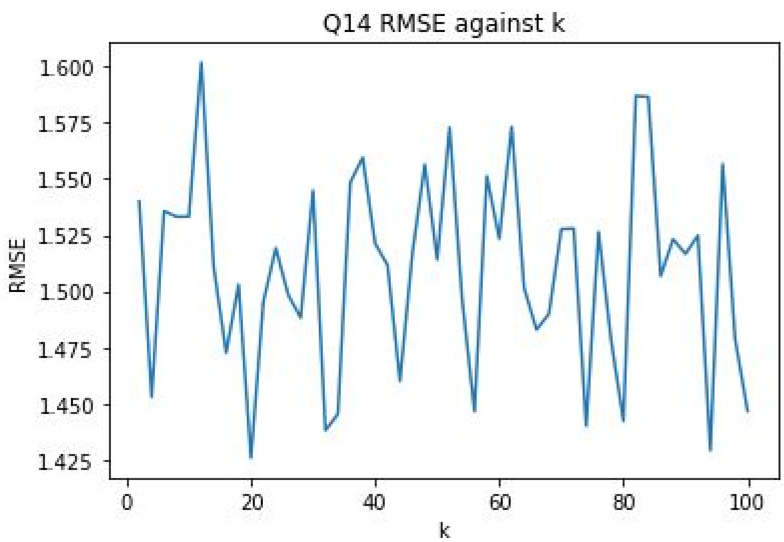


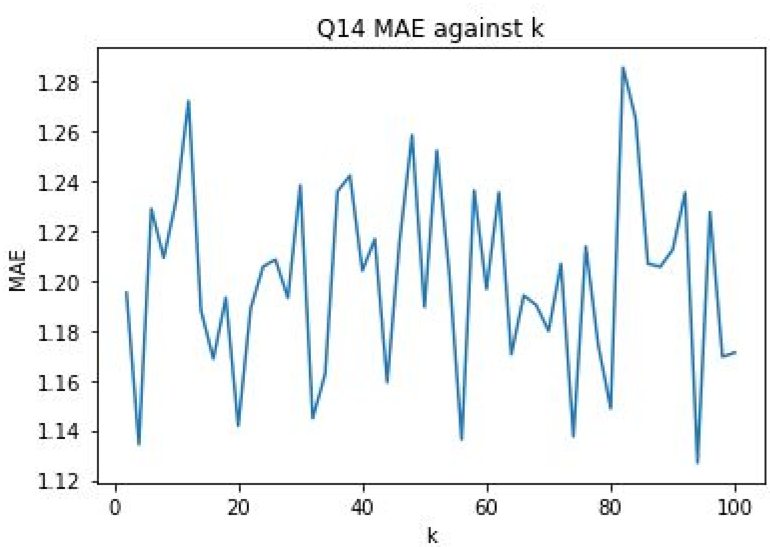



```
Q13 The minimum average RMSE = 1.1087032145110896
```

Question 14:

Following the same steps, we designed a k-NN collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

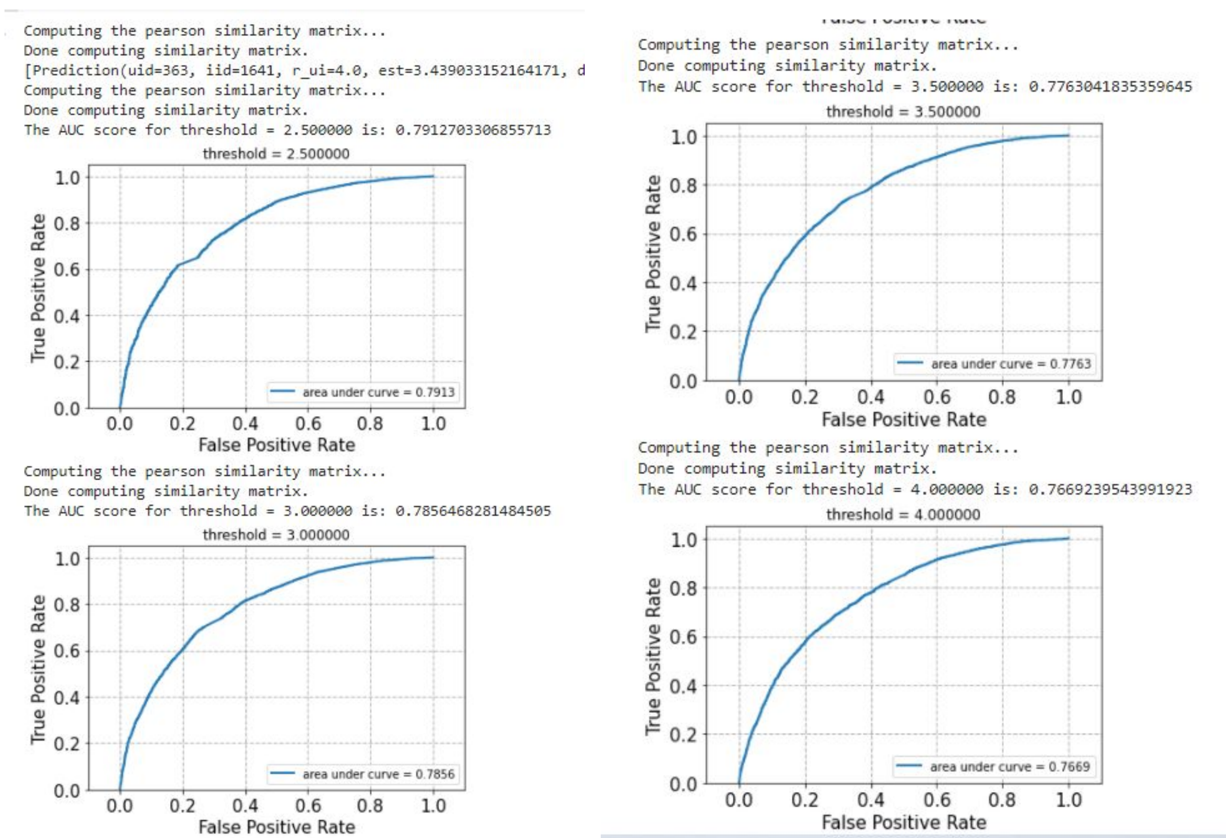





```
Q14 The minimum average RMSE = 1.4258087785916267
```

Question 15:

Receiver operating characteristic (ROC) curve is a commonly used graphical tool for visualizing the performance of a binary classier. It plots the true positive rate (TPR) against the false positive rate (FPR). And since the observed ratings were on a continuous scale (0-5), we converted the observed ratings to a binary scale. If the observed rating was greater than the threshold value, we set it to 1 (implies that the user liked the item). If the observed rating was less than the threshold value, we set it to 0 (implies that the user disliked the item). After performing this conversion, we plotted the ROC curve for the recommendation system for threshold values [2.5, 3, 3.5, 4] and reported the area under the curve (AUC) values.



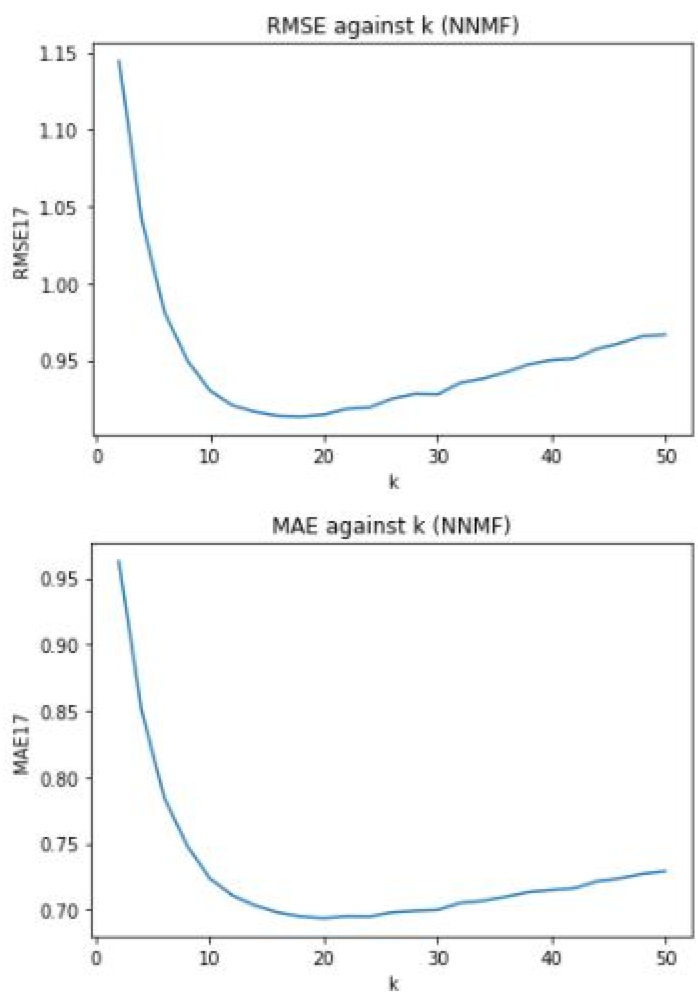
Question 16:

Yes, it is convex because it is the sum of least square problems and we know that least squares are convex because their second order derivative is positive semi-definite.. For fixed U we can get the following least square problem.

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

Question 17:

The previous problems gave us the basics needed to implement a NNMF-based collaborative filter for predicting the ratings of the movies. And in this part, we designed a NNMF-based collaborative filter and tested the performance via 10-fold cross validation. Specifically, we designed a k-NN collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated the performance using 10-fold cross validation. We swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. The plotted average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis) are attached below:

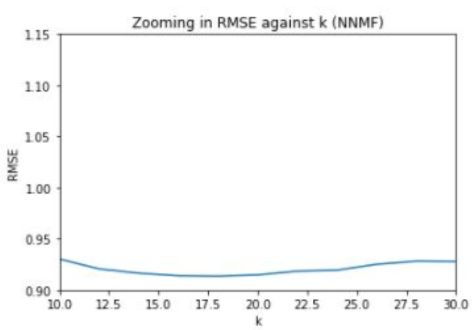


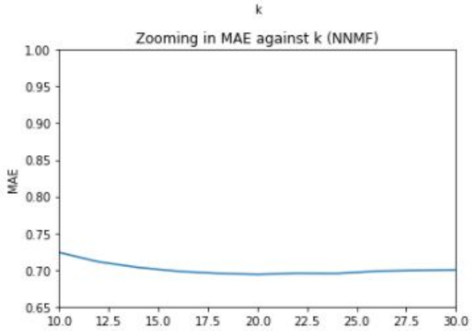
Question 18:

We used the plot generated from Question 17 to find a minimum k that gave us the minimum average RMSE or the minimum average MAE. Firstly, we printed out all the k values and its corresponding RMSE and MAE values that are shown below:

k = 2 RMSE = 1.1444937971494193 MAE = 0.962635250363969
k = 4 RMSE = 1.0411135982360455 MAE = 0.8503071368667923
k = 6 RMSE = 0.9810692302236065 MAE = 0.7843614437100532
k = 8 RMSE = 0.9498989615465245 MAE = 0.7484941268923748
k = 10 RMSE = 0.930149568912085 MAE = 0.7237264585294048
k = 12 RMSE = 0.9205582166316969 MAE = 0.7108925832080641
k = 14 RMSE = 0.9163501758769446 MAE = 0.7032680022059059
k = 16 RMSE = 0.9138709505889615 MAE = 0.6980658612005299
k = 18 RMSE = 0.9134135720940684 MAE = 0.6951725432075364
k = 20 RMSE = 0.9146794004138192 MAE = 0.6938313664092042
k = 22 RMSE = 0.918341333262557 MAE = 0.6953109473486624
k = 24 RMSE = 0.9192921374779603 MAE = 0.695050875028436
k = 26 RMSE = 0.9250625498709745 MAE = 0.6982195198039988
k = 28 RMSE = 0.928068384199636 MAE = 0.6993120366096315
k = 30 RMSE = 0.9277303083201296 MAE = 0.6999870287422746
k = 32 RMSE = 0.9354237396704018 MAE = 0.705278082094918
k = 34 RMSE = 0.9380945453985114 MAE = 0.7069956071013339
k = 36 RMSE = 0.9421542311930736 MAE = 0.7099971258271346
k = 38 RMSE = 0.9473644579764977 MAE = 0.7134154882474092
k = 40 RMSE = 0.9501103021995668 MAE = 0.7149768087130288
k = 42 RMSE = 0.9510039238473411 MAE = 0.7162146709533888
k = 44 RMSE = 0.9573824192196533 MAE = 0.7215998685932075
k = 46 RMSE = 0.9610480089847204 MAE = 0.7238507871377058
k = 48 RMSE = 0.9656055172735959 MAE = 0.7271336423461404
k = 50 RMSE = 0.9666517731976685 MAE = 0.7291389472814241

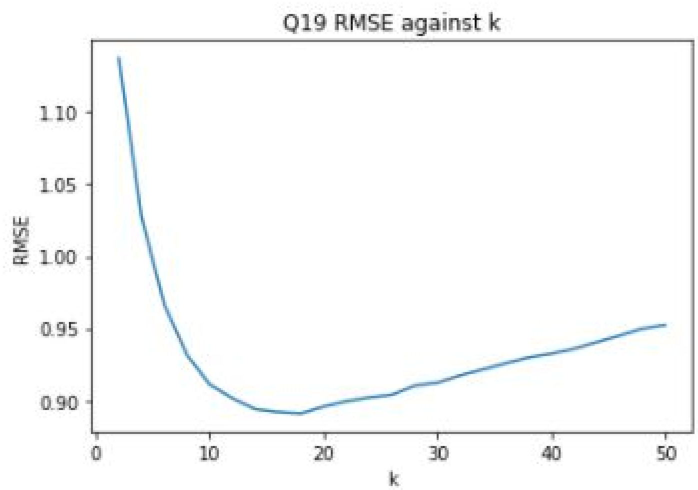
After that, we zoomed in the RMSE and MAE plots generated in Question 10. And from the plots, we could see that k = 18 gave us the minimum value.

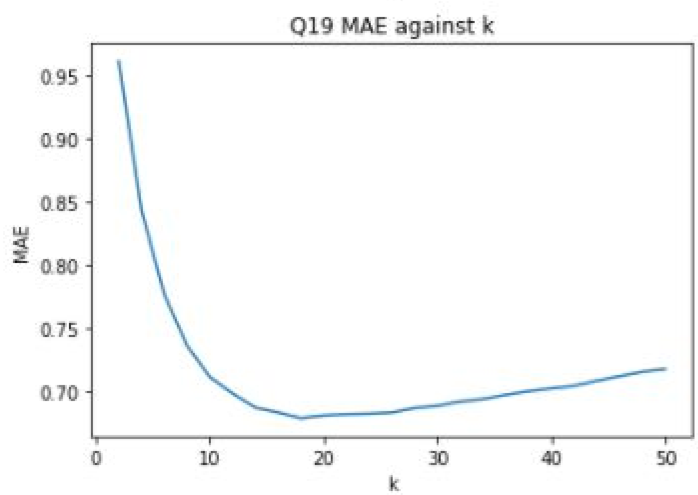




Question 19:

We analyzed the performance of the NNMF-based collaborative filter in predicting the ratings of the movies in the trimmed test set. There were 3 trimmings that we implemented which were popular movie trimming, unpopular movie trimming, adn high variance move trimming. Having coded the functions of these 3 trimmings, we designed a NNMF-based collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

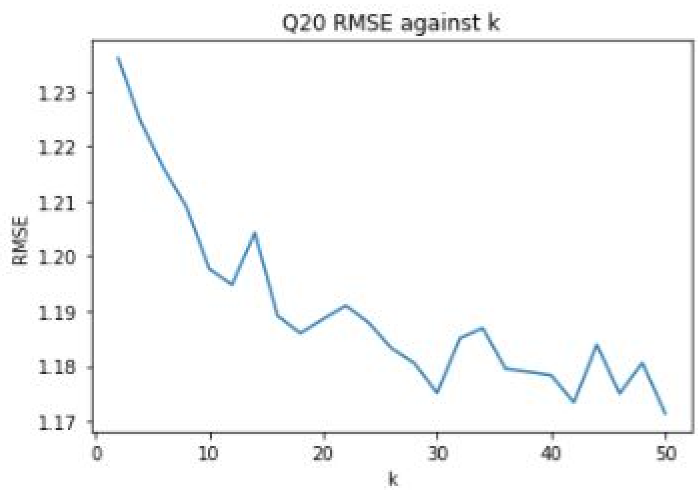


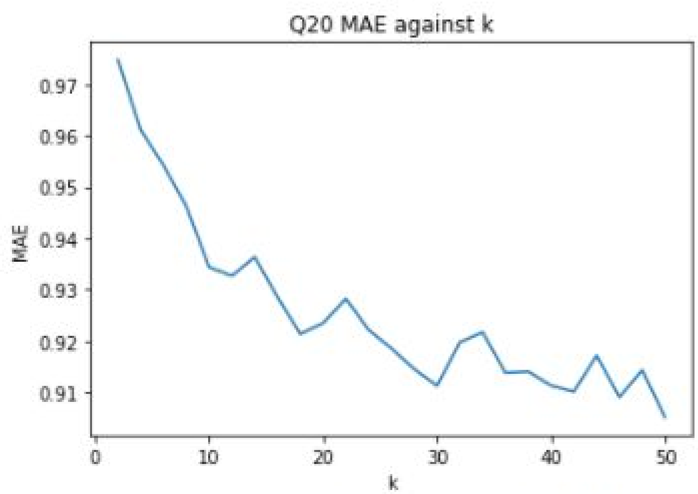



```
Q19 The minimum average RMSE = 0.8910246216755748
```

Question 20:

Following the same steps, we designed a NNMF-based collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

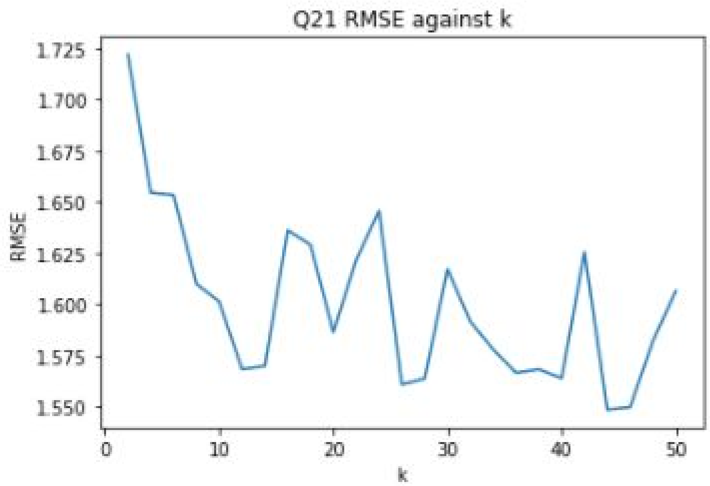


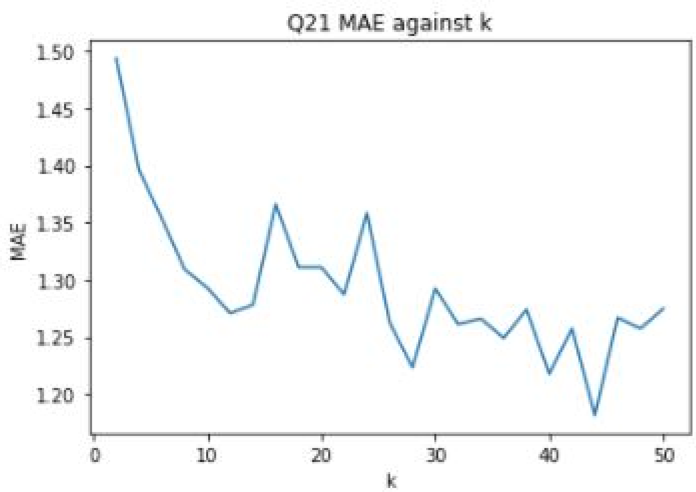



```
Q20 The minimum average RMSE = 1.1713568240697168
```

Question 21:

Following the same steps, we designed a NNMF-based collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.



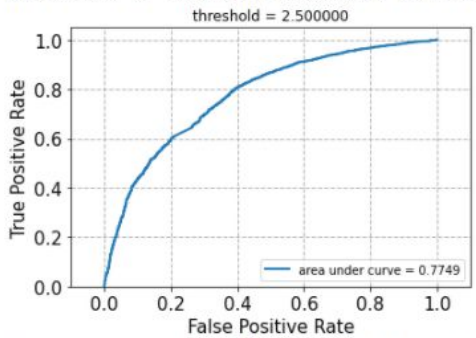



```
Q21 The minimum average RMSE = 1.548601768663448
```

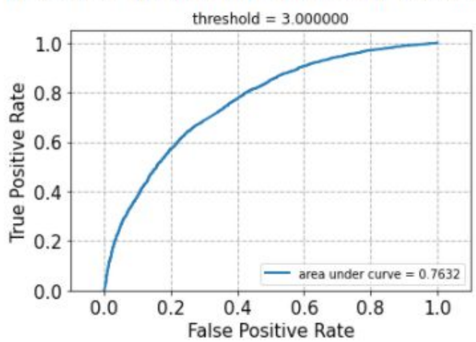
Question 22:

Since the observed ratings were on a continuous scale (0-5), we converted the observed ratings to a binary scale. If the observed rating was greater than the threshold value, we set it to 1 (implies that the user liked the item). If the observed rating was less than the threshold value, we set it to 0 (implies that the user disliked the item). After performing this conversion, we plotted the ROC curve for the recommendation system for threshold values [2.5, 3, 3.5, 4] and reported the area under the curve (AUC) values for NNMF-based collaborative filtering.

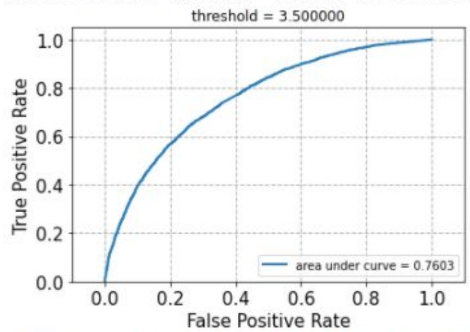
```
[Prediction(uid=18, iid=5418, r_ui=4.0, est=3.9670728207712265
The AUC score for threshold = 2.500000 is: 0.7749123540607836
```



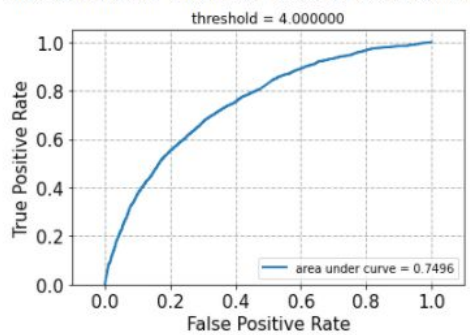

```
The AUC score for threshold = 3.000000 is: 0.763182701096297
```




```
The AUC score for threshold = 3.500000 is: 0.7603021893651041
```




```
The AUC score for threshold = 4.000000 is: 0.7496145538010528
```



Question 23:

For Latent Factor 0, the top 10 movies' genres are

```
Comedy|Crime
Comedy
Adventure|Romance|Thriller
Comedy|Romance
Comedy|Musical|Romance
Comedy
Comedy
Comedy|Documentary
Comedy|Drama|Fantasy|Romance
Action|Comedy|Thriller
```

For Latent factor 2, the top 10 movies' genres are

```
Drama
Drama|Romance
Horror
Crime|Drama
Action|Crime|Drama
Comedy|Romance
Comedy|Drama|Romance
Horror|Thriller
Drama
Action|Thriller
```

For Latent Factor 4, the top 10 movies' genres are

```
Action|Thriller
Action|Adventure|Fantasy
Action|Comedy|Crime
Adventure|Animation|Children|Fantasy
Drama
Action|Comedy|Drama|Horror|Thriller
Crime|Drama|Film-Noir|Thriller
Action|Horror|Thriller
Horror
Comedy|Horror
```

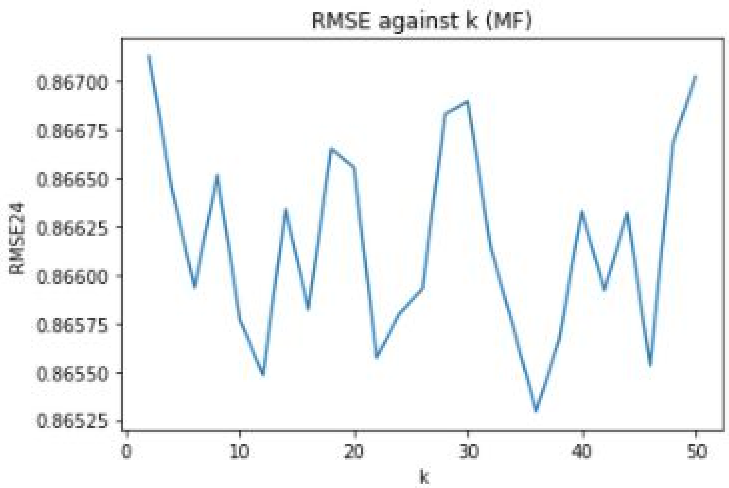
Latent Factor 19:

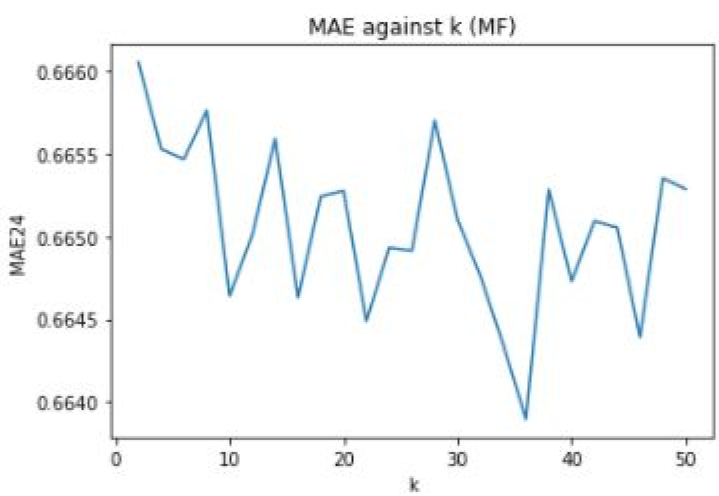
```
Sci-Fi
Comedy
Comedy
Crime|Drama|Thriller
Drama
Comedy|Drama|Romance
Comedy|Romance
Comedy|Romance
Drama
Documentary
```

We can see that in Latent Factor 0, most of the movies are comedies, and two of them are thrillers. For Latent Factor 2, most genres' are drama and romance. And for Latent Factor 4, most of movies' genres are action, horror and thriller. We can also see that, as the number of latent factors increases, the number of distinct movie genres decreases.

Question 24:

The previous problems gave us the basics needed to implement a MF with bias collaborative filter for predicting the ratings of the movies. And in this part, we designed a NNMF-based collaborative filter and tested the performance via 10-fold cross validation. Specifically, we designed a MF with bias collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluated the performance using 10-fold cross validation. We swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. The plotted average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis) are attached below:





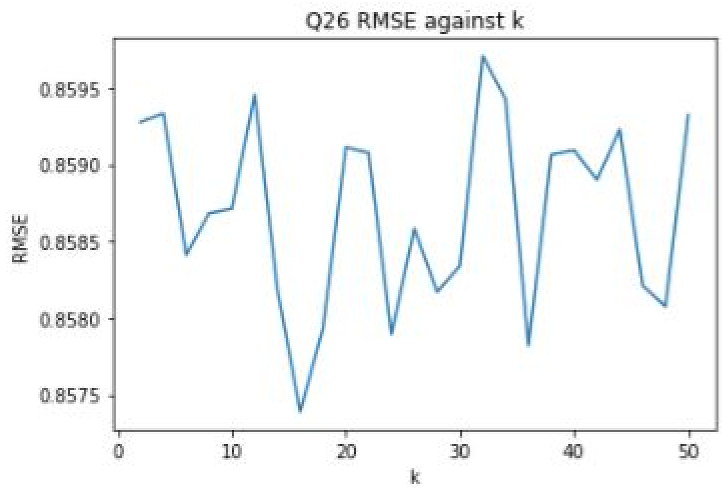
Question 25:

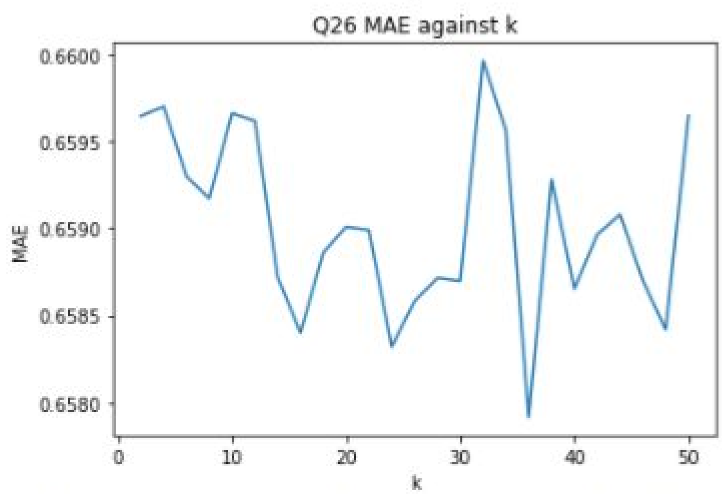
We used the plot generated from Question 24 to find a minimum k that gave us the minimum average RMSE or the minimum average MAE. Firstly, we printed out all the k values and its corresponding RMSE and MAE values that are shown below and found the minimum k to be 36.

k = 2 RMSE = 0.8671318938383331 MAE = 0.6660545943242185
k = 4 RMSE = 0.8664474412393155 MAE = 0.6655290971301306
k = 6 RMSE = 0.8659341135988881 MAE = 0.6654680956365636
k = 8 RMSE = 0.8665173469867149 MAE = 0.6657636631689424
k = 10 RMSE = 0.8657673513316153 MAE = 0.6646416038976244
k = 12 RMSE = 0.8654826712742685 MAE = 0.6650125720997545
k = 14 RMSE = 0.8663397909424619 MAE = 0.6655929119194799
k = 16 RMSE = 0.8658208713483433 MAE = 0.6646319285209685
k = 18 RMSE = 0.8666515459316511 MAE = 0.6652434227436481
k = 20 RMSE = 0.8665556414578427 MAE = 0.6652772814720593
k = 22 RMSE = 0.8655709253315205 MAE = 0.6644891983620438
k = 24 RMSE = 0.8658001881029966 MAE = 0.6649327639715172
k = 26 RMSE = 0.8659268123115812 MAE = 0.6649149363167196
k = 28 RMSE = 0.8668318616798395 MAE = 0.6657026493509723
k = 30 RMSE = 0.866896778526581 MAE = 0.6651017171651408
k = 32 RMSE = 0.8661428014004254 MAE = 0.6647619694751016
k = 34 RMSE = 0.8657273742743301 MAE = 0.6643468565079712
k = 36 RMSE = 0.8652934037649486 MAE = 0.6638956679344885
k = 38 RMSE = 0.8656604320420183 MAE = 0.6652832587116214
k = 40 RMSE = 0.8663284925916479 MAE = 0.6647291733219809
k = 42 RMSE = 0.8659190598726558 MAE = 0.6650934929373904
k = 44 RMSE = 0.8663204177102598 MAE = 0.6650545690164152
k = 46 RMSE = 0.8655320816772367 MAE = 0.664391464282404
k = 48 RMSE = 0.866677352587442 MAE = 0.6653533329346306
k = 50 RMSE = 0.8670251666240176 MAE = 0.6652875065770403
RMSE24 min k = 36
The min RMSE24 is: 0.8652934037649486
MAE24 min k = 36
The min MAE24 is: 0.6638956679344885

Question 26:

We analyzed the performance of the MF with bias collaborative filter in predicting the ratings of the movies in the trimmed test set. There were 3 trimmings that we implemented which were popular movie trimming, unpopular movie trimming, adn high variance move trimming. Having coded the functions of these 3 trimmings, we designed a MF with bias collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

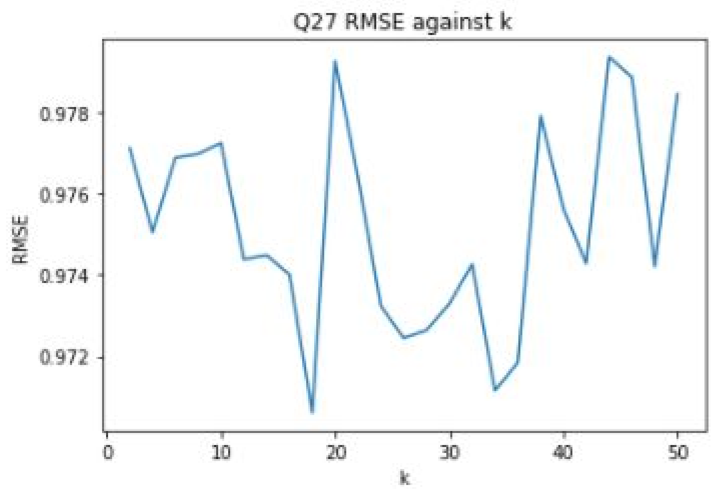


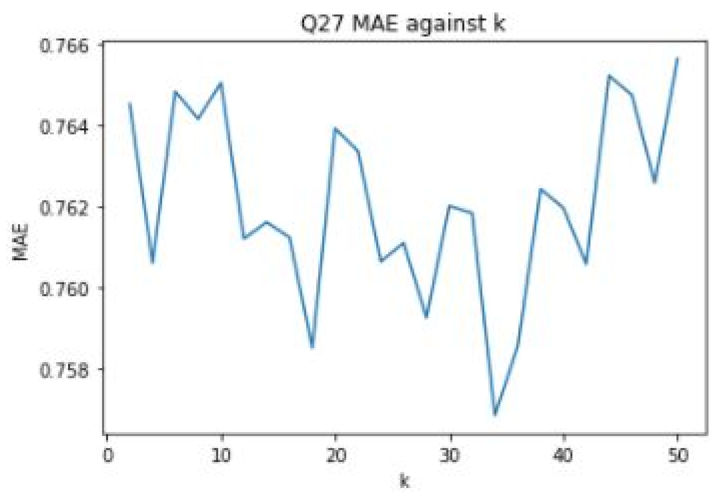



```
Q26 The minimum average RMSE = 0.8573905938938899
```

Question 27:

Following the same steps, we designed a MF with bias collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

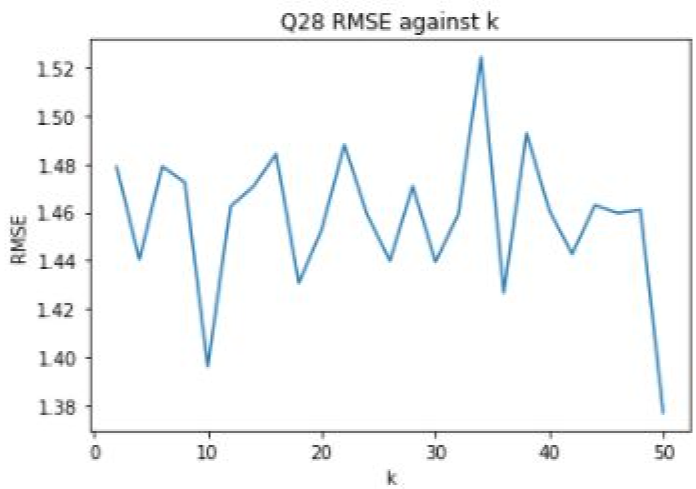


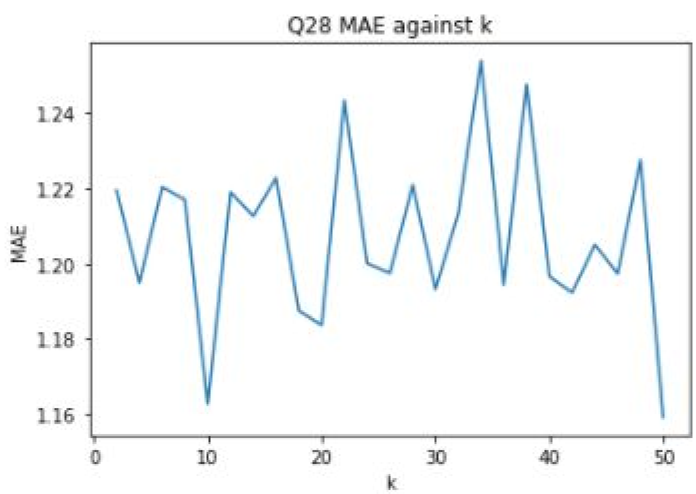



```
Q27 The minimum average RMSE = 0.9706150440593853
```

Question 28:

Following the same steps, we designed a MF with bias collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate the performance using 10-fold cross validation. And we swept k (number of neighbors) from 2 to 50 in step sizes of 2, and for each k computed the average RMSE obtained by averaging the RMSE across all 10 folds. We plotted average RMSE (Y-axis) against k (X-axis) and reported the minimum average RMSE.

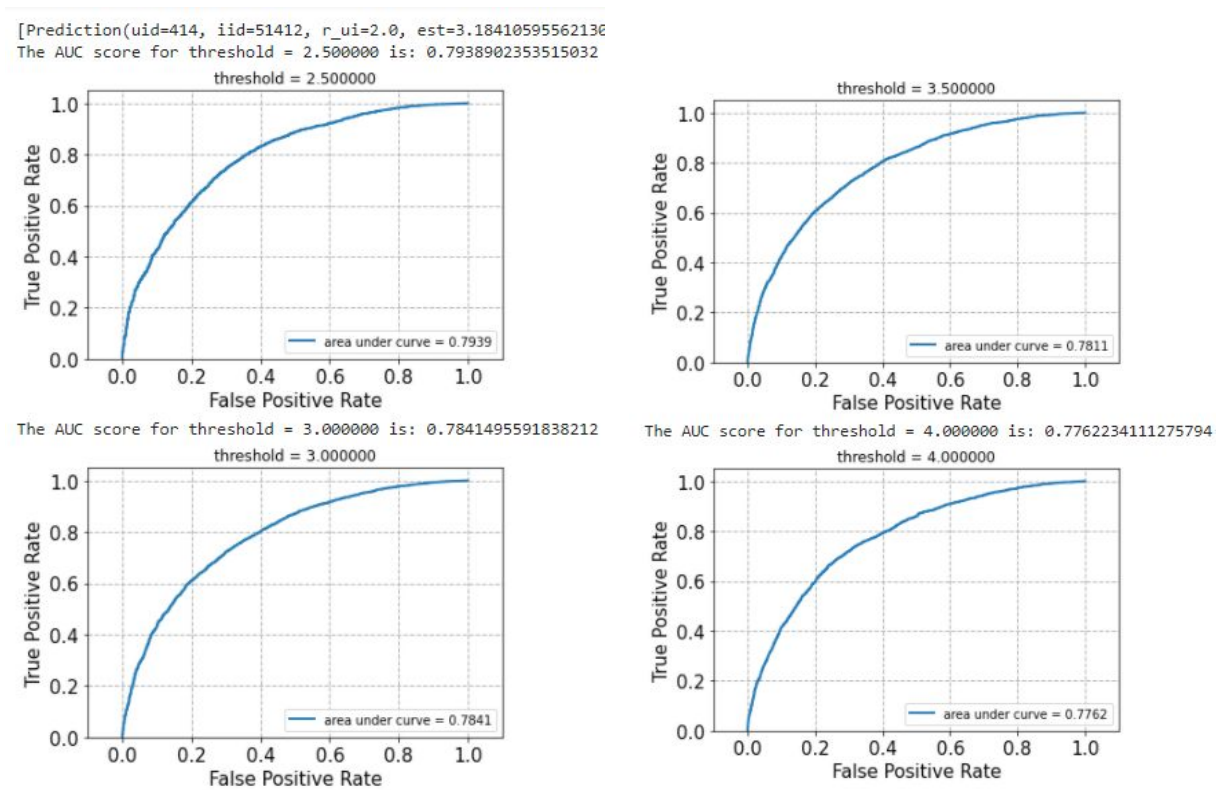





```
Q28 The minimum average RMSE = 1.3768954366521067
```

Question 29:

Since the observed ratings were on a continuous scale (0-5), we converted the observed ratings to a binary scale. If the observed rating was greater than the threshold value, we set it to 1 (implies that the user liked the item). If the observed rating was less than the threshold value, we set it to 0 (implies that the user disliked the item). After performing this conversion, we plotted the ROC curve for the recommendation system for threshold values [2.5, 3, 3.5, 4] and reported the area under the curve (AUC) values for MF with bias collaborative filtering.



Question 30

Q30 RMSE = 1.0425239038060634

Question 31

For popular movie trimming:

Q31 RMSE = 1.0356918285924035

Question 32

For unpopular movie trimming

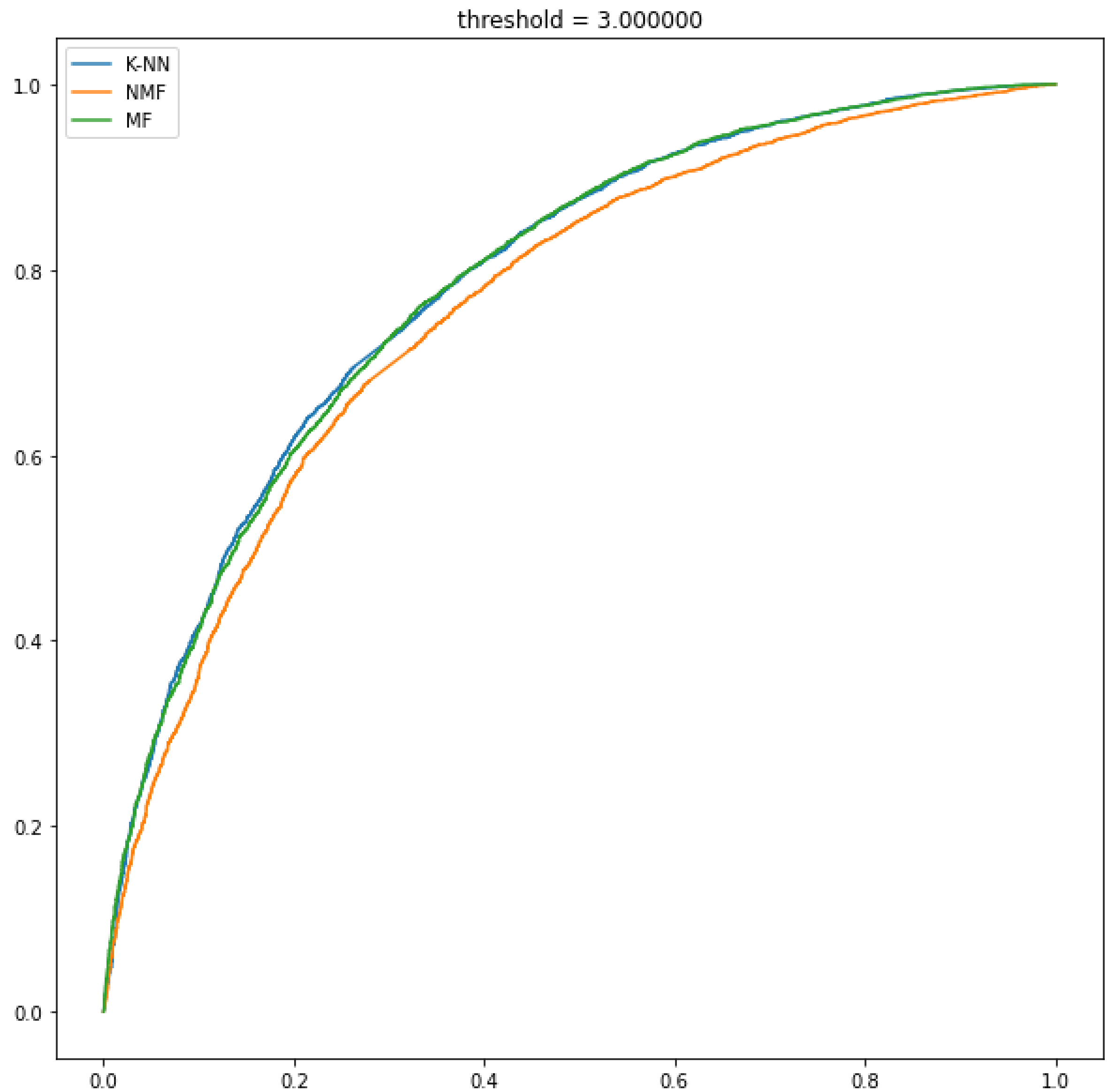
Q32 RMSE = 1.144512973267003

Question 33

For High variance movie trimming

```
Q33 RMSE = 1.1188375038121632
```

Question 34



AUC for each Method:

knn 0.7892620774200663

nmf 0.7680042848885447

mf 0.7913493027763276

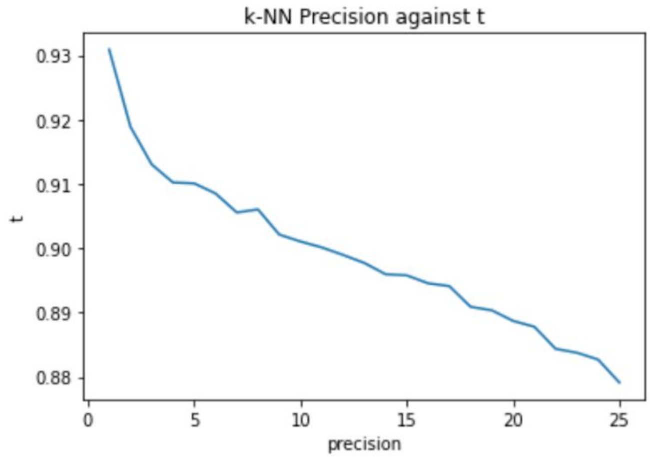
From the graph and the calculated AUC. We can see that MF with bias has a slightly smoother curve and has the largest AUC at threshold = 3. Therefore, MF with bias performs best among three methods.

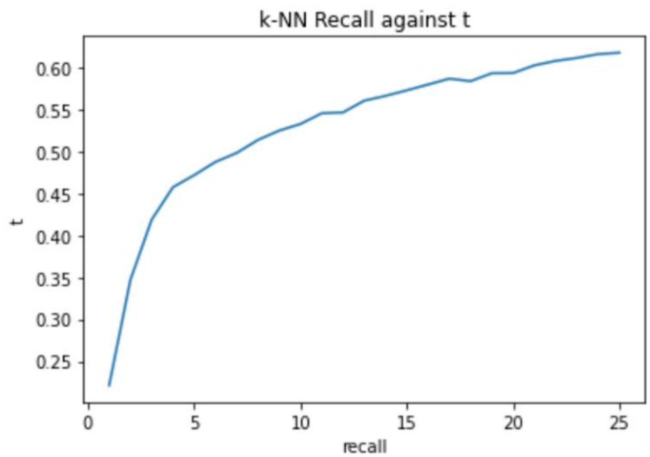
Question 35

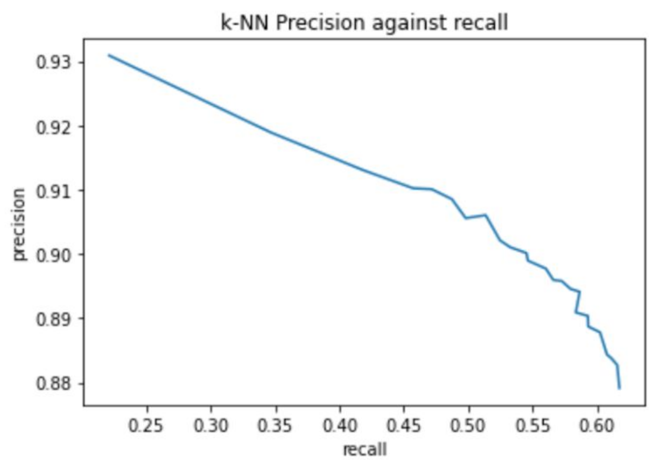
Precision is the parameter of how many correct recommendations compare to all recommendations. This factor considers all of the values. Recall is the parameter which considers how many ratings that people actually like. This is more accurate since this represents ground-truth positives.

Question 36

In this question, k-NN based collaborative filtering is applied with kmin from question 11.



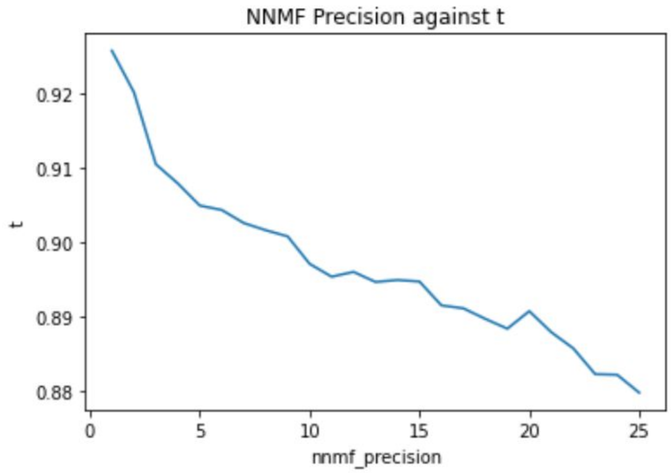


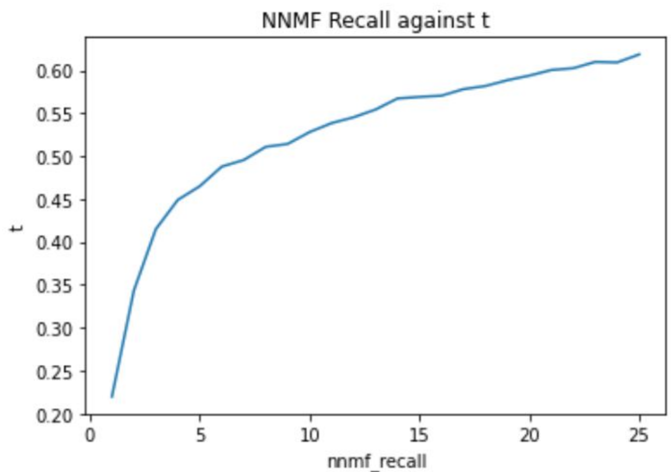


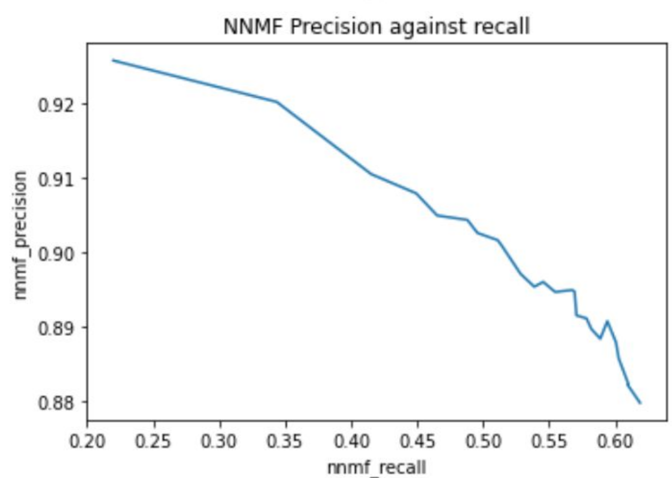
From the plots we can see that precision decreases along with t while recall increases along t, and the relationship between precision and recall is negatively related.

Question 37

In this question, NNMF based collaborative filtering is applied with kmin from question 18.

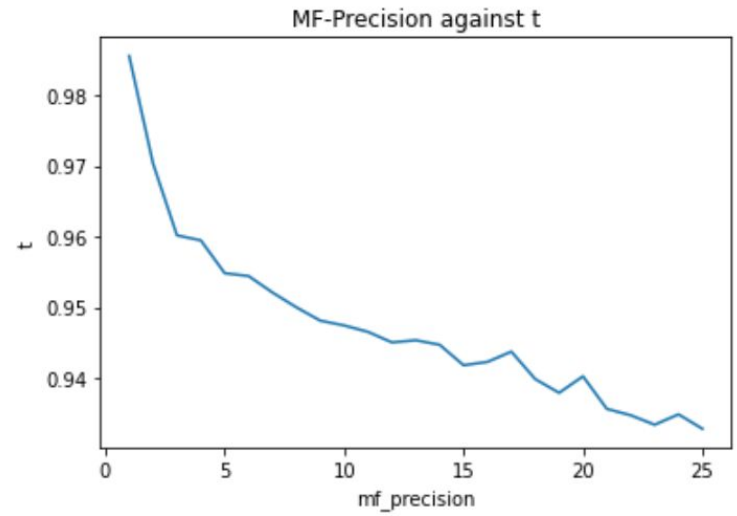


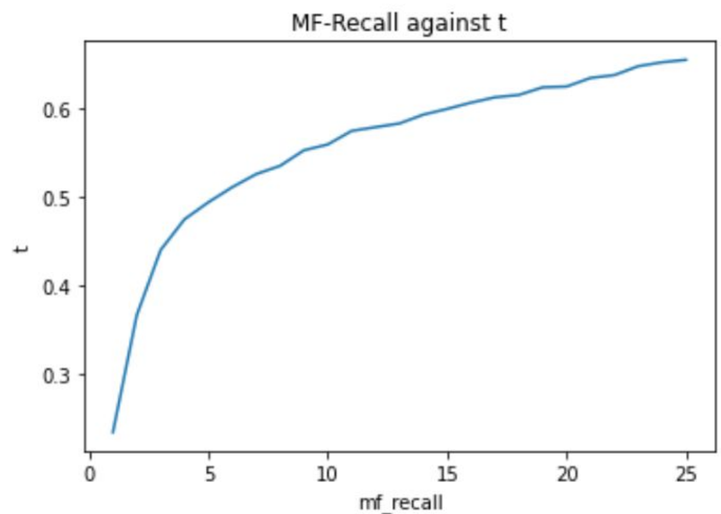


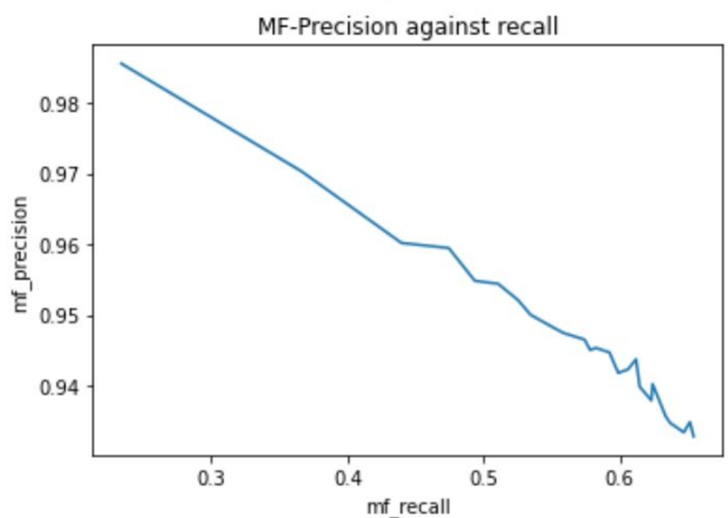


Question 38

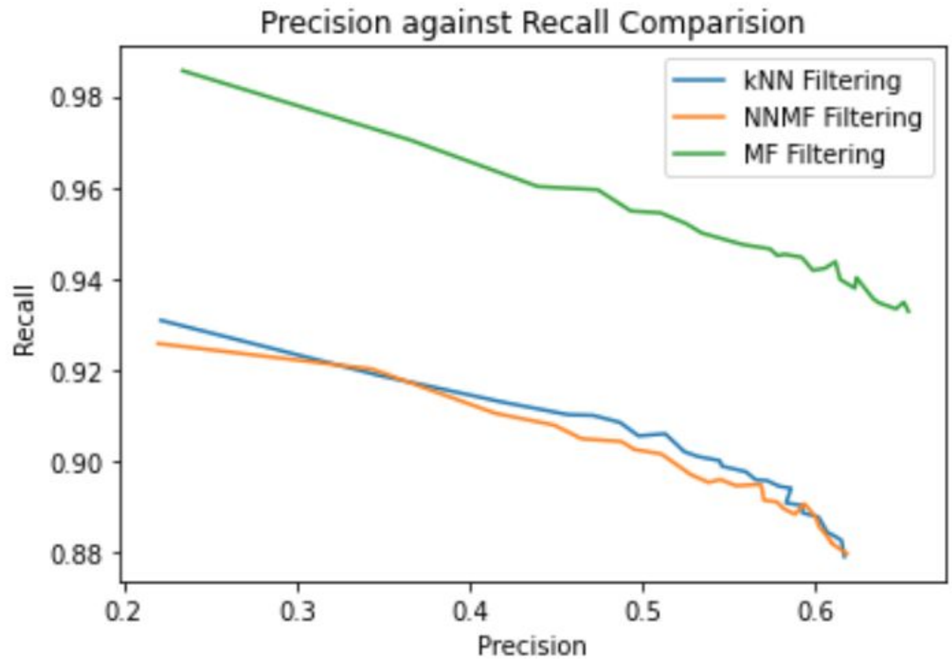
In this question, MF with biased-based collaborative filtering is applied with kmin from question 25.


MF-Precision against t
0.98
0.97
0.96
0.95
0.94
t
0
5
10
15
20
25
mf_precision


MF-Recall against t
0.6
0.5
0.4
0.3
t
0
5
10
15
20
25
mf_recall


MF-Precision against recall
0.98
0.97
0.96
0.95
0.94
mf_precision
0.3
0.4
0.5
0.6
mf_recall

Question 39



From the plot above we can see that MF filtering achieves highest accuracy while NNMF overall has the poorest performance, which matches the ROC curves.