EE 219

Large-Scale Data Mining

Project 3

Collaborative Filtering

Winter 2018

By Xudong Li (804944940),

Tao Wu (504946672),

Yangyang Mao (504945234),

Di Jin (305026178)

February 22, 2018

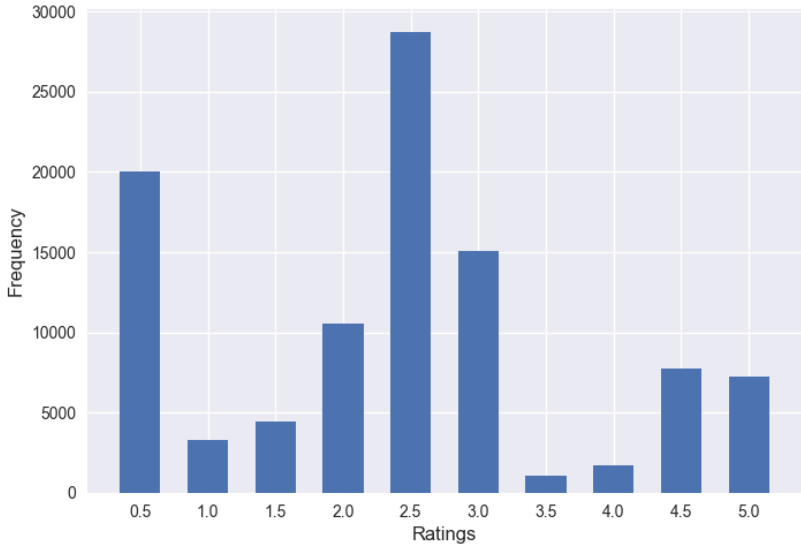
Question 1

The sparsity is defined by the equation,

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

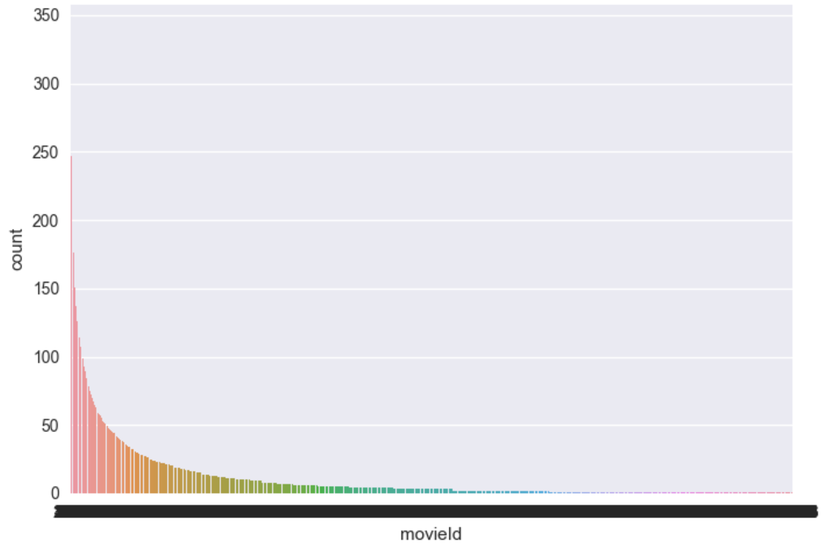
and the sparsity of the movie rating dataset is 0.0164391416087

Question 2



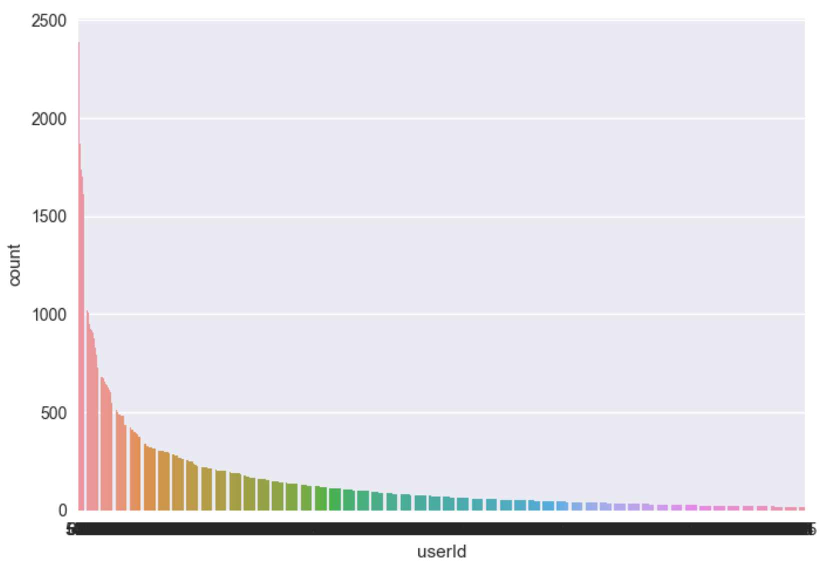
The is a histogram showing the frequency of the rating values. The shape of the histogram is like normal distribution, but a very low score of 0.5 is also frequent. Most of the scores ranges from 2.0 to 3.0, and there are also some ratings from 4.5 to 5.0

Question 3



This is the distribution of the ratings received by movies. We can see that over 90 percent of the movies received less than 50 ratings. Only a few movies have very high amount of ratings, and this indicates that our dataset is very sparse.

Question 4

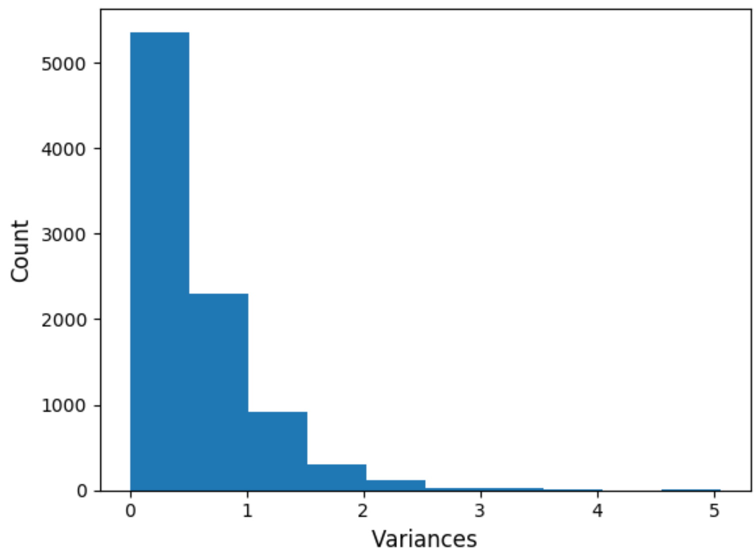


This is a distribution of ratings that users provided. We can see that over 90 percent of the users rated less than 500 movies. Only a few users have very high amount of ratings, and this indicates that our dataset is very sparse.

Question 5

We find that the number of ratings decreases exponentially, which the most popular movies have more than 150 ratings, while over 90% of the movies have less than 50 ratings. This means that the rating matrix is very sparse, which limits the coverage of neighborhood-based collective filtering. It creates a challenge for robust similarity computation when the number of mutually rated items between two users is small. If none of the target users' neighbors have rated a movie, then it is not possible to provide a rating prediction of that movie.

Question 6



This is variance of the rating values received by each movie. It shows that most of the movie ratings have a variance of 0 to 0.5, which means the rating are quite even across different users. The shape of this plot is skewed to the left, and there is hardly any movies that have high variance.

Question 7

$$\mu_u = \frac{\sum_1^k r_{uk}}{size(I_u)}$$

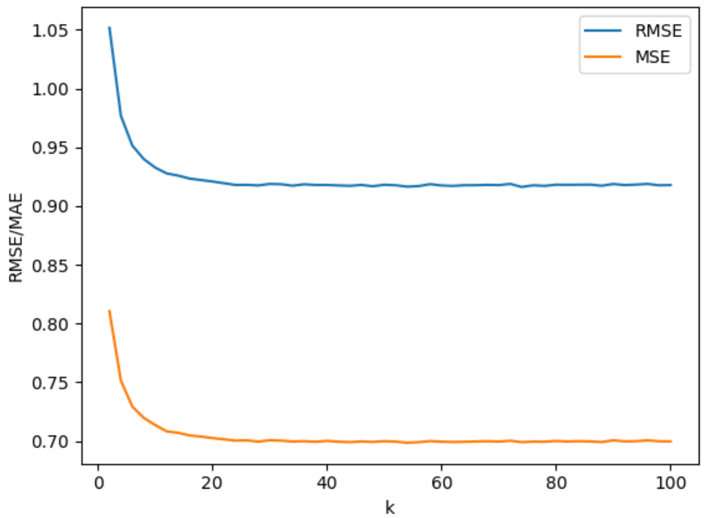
Question 8

The intersection of I_u and I_v is the common movies that user u and user v have rated. Such intersection can be an empty set when the rating matrix R is sparse, which means the two users have not rated a same movie yet.

Question 9

The mean-centering in the prediction function is used to eliminate the innate bias from users. For example, if a user is an easy grader who usually gives a 5-star review as long as the movie is not too bad, then a rating of 3 stars, which means s/he thinks the movie is really bad, could be differentiated from these 5-star reviews. On the other hand, a harsh criticizer usually gives a 2-star review, if s/he rates a 5, then s/he must like the movie a lot. Now, consider the case that these two people are neighbors of each other in the collective filtering system, the prediction will not be affected by such innate bias, because we subtract the mean when doing the predictions.

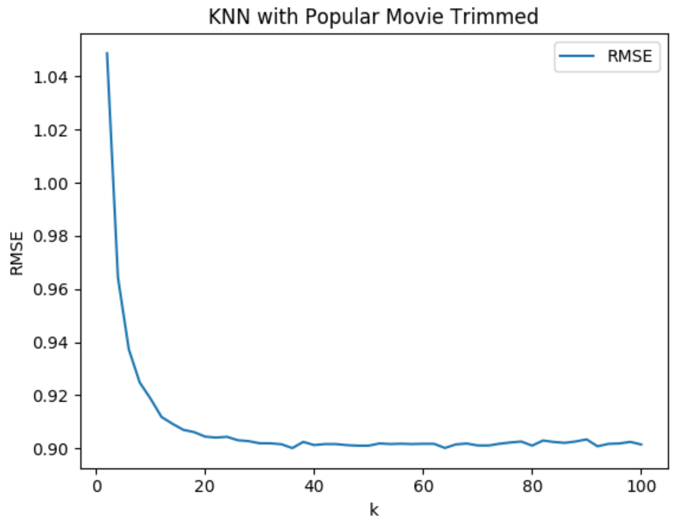
Question 10



Question 11

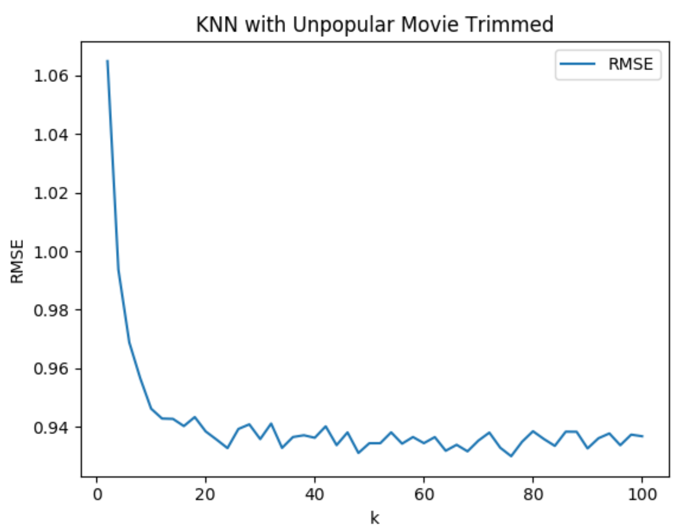
The k value for steady state is 20, and RMSE converges to 0.917, while MAE converges to 0.7

Question 12



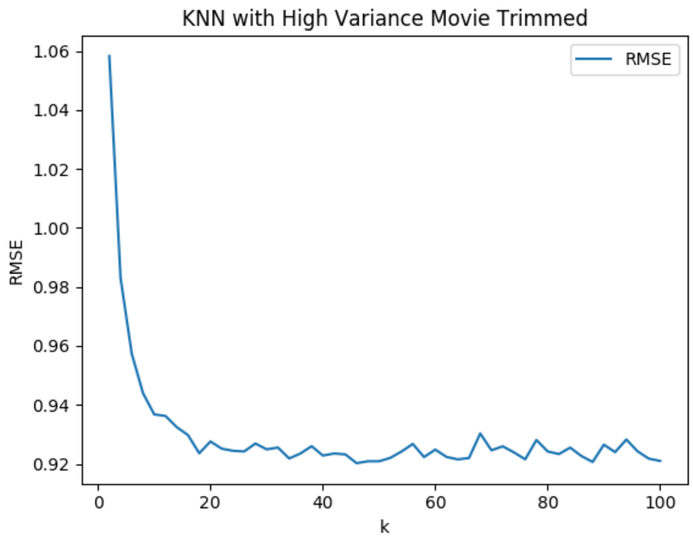
Steady State k = 20, minimum average RMSE = 0.901

Question 13



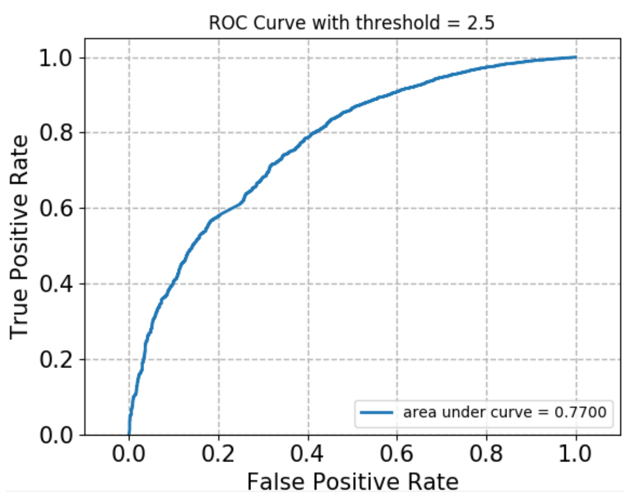
Steady State k = 18, minimum average RMSE = 0.931

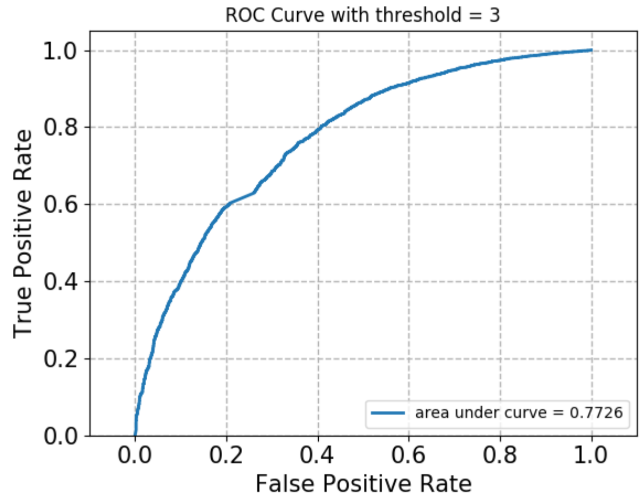
Question 14

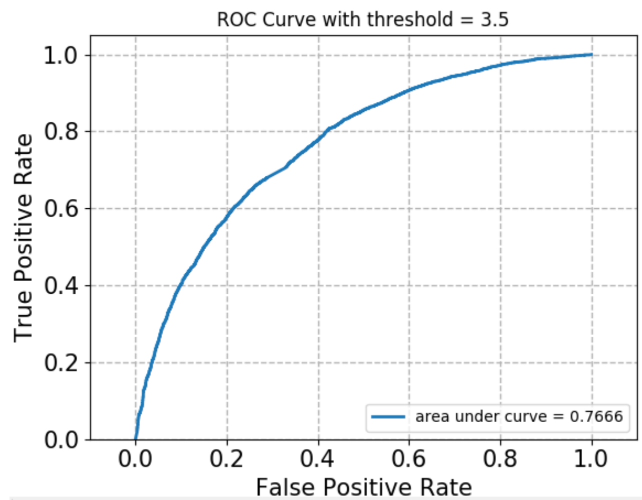


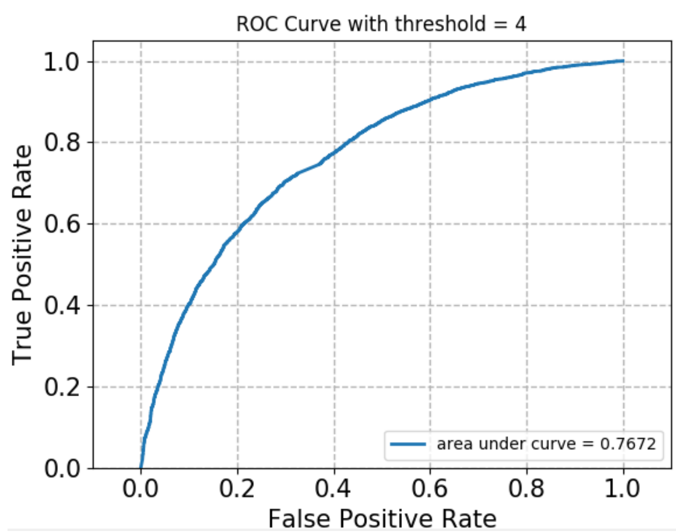
Steady State k = 20, minimum average RMSE = 0.920

Question 15




ROC Curve with threshold = 3
True Positive Rate
False Positive Rate
area under curve = 0.7726


ROC Curve with threshold = 3.5
True Positive Rate
False Positive Rate
area under curve = 0.7666


ROC Curve with threshold = 4
True Positive Rate
False Positive Rate
area under curve = 0.7672

Question 16

$$f = \sum_i \sum_j w_{ij} \left(r_{ij} - (UV^T)_{ij}\right)^2.$$

$$(UV^T)_{ij} = \sum_k u_{ik} v_{jk}.$$

$$f' = \frac{\partial f}{\partial v_{pq}} = \frac{\partial \sum_i \sum_j w_{ij} \left(r_{ij} - \sum_k u_{ik} v_{jk}\right)^2}{\partial v_{pq}}$$

$$= -\sum_i w_{ip} (r_{ip} - u_{iq} v_{pq}) \cdot 2 u_{iq}.$$

$$f'' = \frac{\partial -2 \sum_i w_{ip} u_{iq} (r_{ip} - u_{iq} v_{pq})}{\partial v_{pq}}$$

$$= 2 \sum_i w_{ip} \cdot u_{iq}^2.$$

For $w_{ip} = 1$ or 0. Hence $f'' = 2 \sum_i w_{ip} u_{iq}^2 \geq 0$.

$\Rightarrow$ f is a convex function.

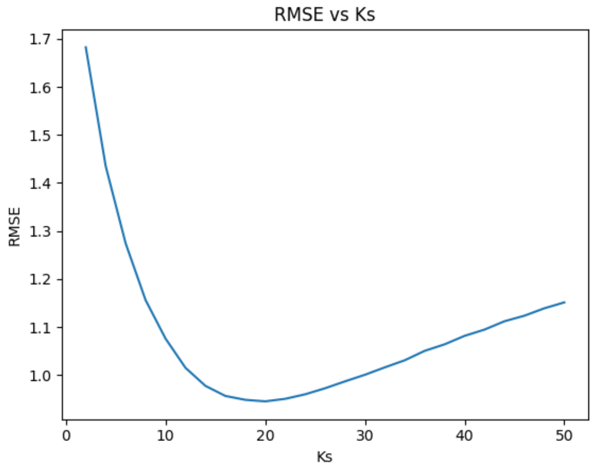
The optimization problem given by equation 5 is convex. In least square format:

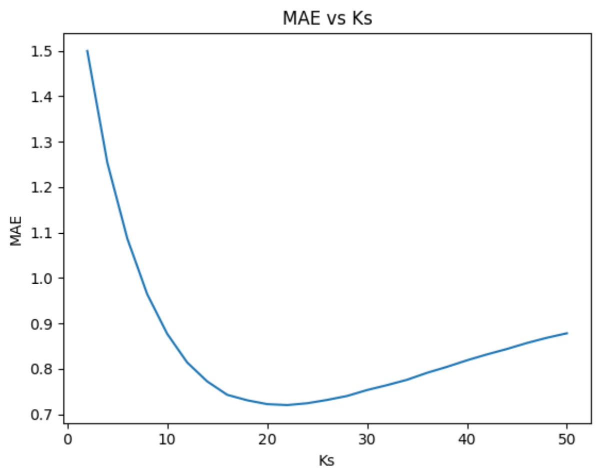
$$x_i = \left(V^T W^i V\right)^{-1} V^T W^i r^i$$

$$y_j = \left(U^T W^j U\right)^{-1} U^T W^j r^j$$

where W^i and W^j are n by n and m by m diagonal matrices with coefficient w_{ij}. r^i and r^j contains element r_{ij} in row j and row i.

Question 17

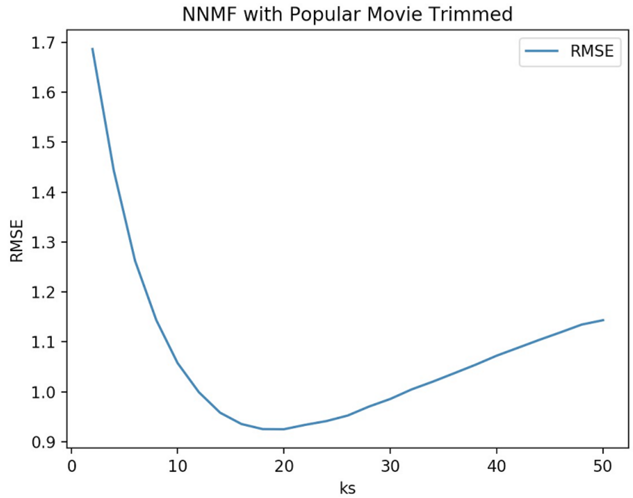




Question 18

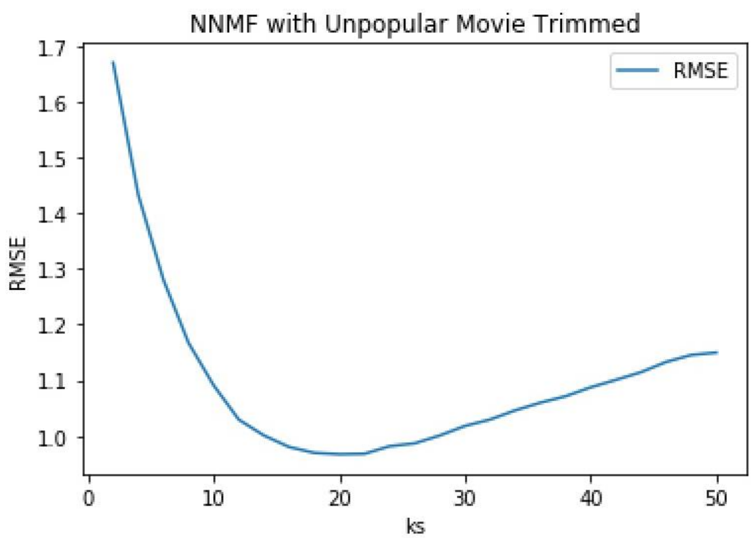
The optimal number of latent factors is 20, with minimum average RMSE of 0.945110449184 and minimum average MAE of 0.720230665665. The optimal number of latent factors is the same as the number of movie genres.

Question 19



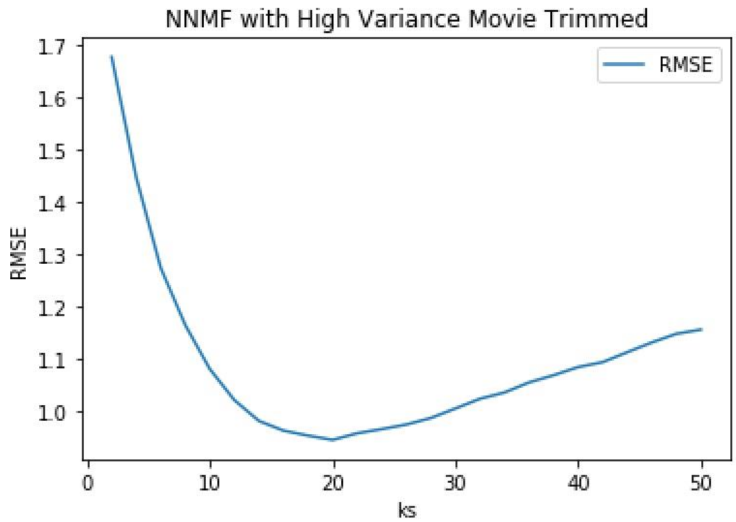
The minimum average RMSE is 0.925022744893

Question 20



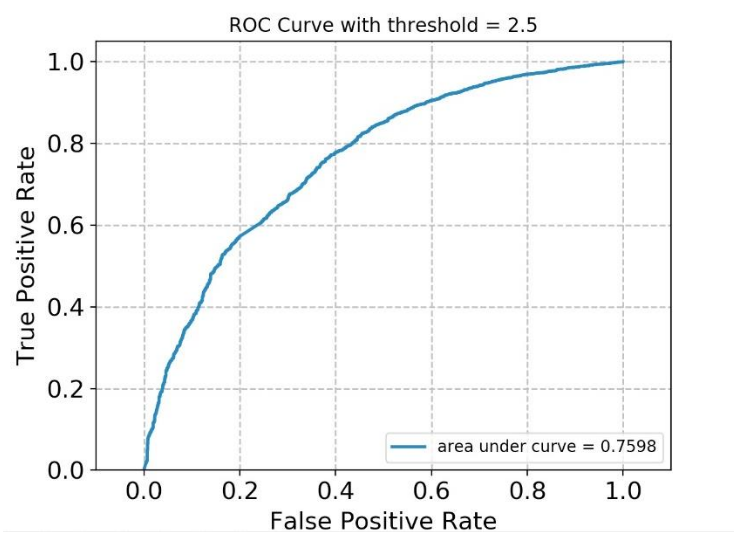
The minimum average RMSE is 0.967282189715

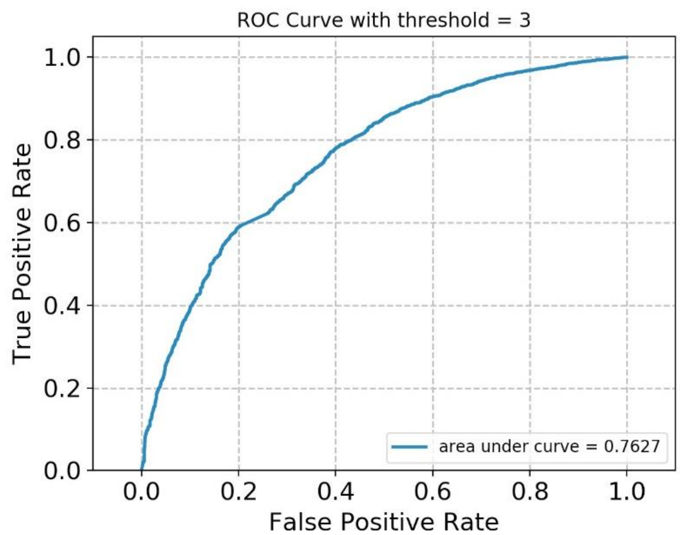
Question 21

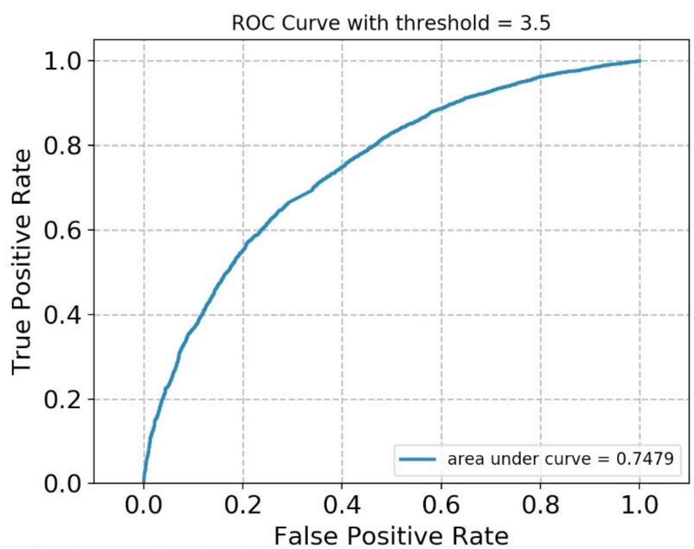


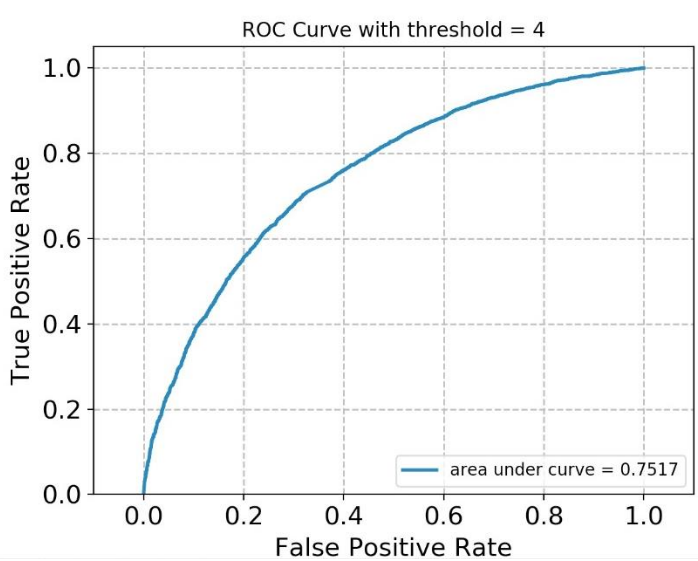
The minimum average RMSE is 0.944532291569

Question 22




ROC Curve with threshold = 3
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
area under curve = 0.7627


ROC Curve with threshold = 3.5
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
area under curve = 0.7479


ROC Curve with threshold = 4
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
area under curve = 0.7517

Question 23

```
(9066, 20)
Movie ID:  ['6219', '5106', '2570', '6450', '2887', '4143', '5530', '4603', '6797', '5117']
Movie Index:  [2062, 2316, 3307, 3617, 3952, 3959, 4187, 4519, 4651, 4817]
Top 10 Movie Genres:  ['Drama|Romance', 'Action|Comedy', 'Horror|Mystery', 'Comedy|Romance', 'Comedy|Musical|Romance',
 'Comedy|Musical', 'Comedy|Drama|Fantasy|Sci-Fi', 'Action|Drama|Thriller', 'Drama|War', 'Crime|Drama']
```

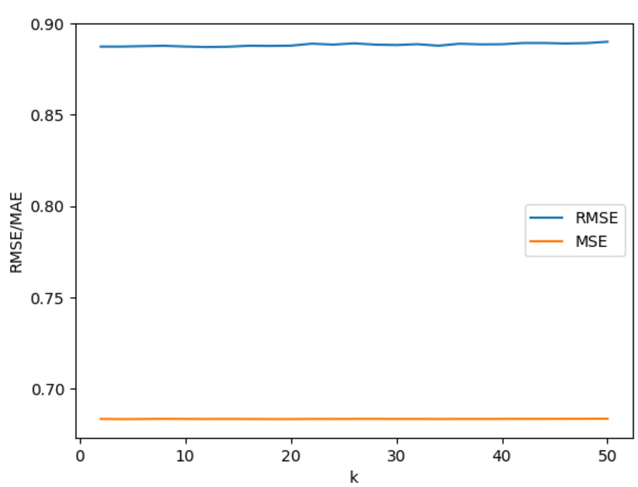
```
(9066, 20)
Movie ID:  ['4630', '3777', '55167', '87522', '1005', '670', '5477', '33681', '4532', '1595']
Movie Index:  [575, 810, 1268, 3022, 3572, 3638, 4158, 6154, 6784, 7830]
Top 10 Movie Genres:  ['Drama', 'Children|Comedy', 'Adventure|Children|Drama', 'Comedy|Horror', 'Comedy|Horror', 'Acti
on', 'Drama|Romance', 'Action|Adventure|Children|Fantasy', 'Action|Adventure|Animation|Crime|Fantasy', 'Comedy|Drama|R
omance']
```

```
(9066, 20)
Movie ID:  ['87522', '2399', '4146', '8574', '3520', '2500', '6772', '3439', '1382', '536']
Movie Index:  [481, 1123, 1910, 2003, 2745, 2813, 3309, 4799, 5457, 7830]
Top 10 Movie Genres:  ['Drama', 'Action|Drama', 'Adventure|Children|Fantasy', 'Comedy', 'Action|Children|Fantasy', 'Co
medy', 'Drama|Mystery|Romance', 'Documentary', 'Animation|Children|Comedy|Musical', 'Comedy|Drama|Romance']
```

```
(9066, 20)
Movie ID:  ['2766', '499', '2042', '33085', '8666', '2793', '97817', '70565', '1105', '91104']
Movie Index:  [447, 895, 1604, 2216, 2238, 5500, 6125, 7297, 7937, 8171]
Top 10 Movie Genres:  ['Comedy|Romance', 'Horror', 'Children|Comedy', 'Comedy|Drama', 'Comedy|Horror|Romance|Thriller'
, 'Action|Crime|Fantasy', 'Horror|Thriller', 'Comedy', 'Adventure|Drama|Fantasy|Romance', 'Documentary']
```

From the result we got. We can easily find that each of these top 10 movies belongs to one specific collection. They all have a genre. Also, we can conclude that the movie latent factors correspond to the movie ID in the movie.csv. Due to this fact, I can use the same ID in the movie.csv to find the index of one specific movie and finally find their genres.

Question 24



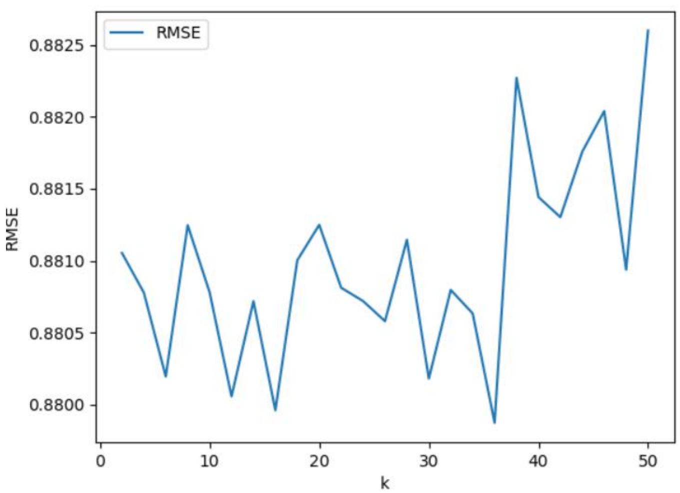
Question 25

The optimal number of latent factors is 16

The minimum average RMSE is 0.8866977950278114

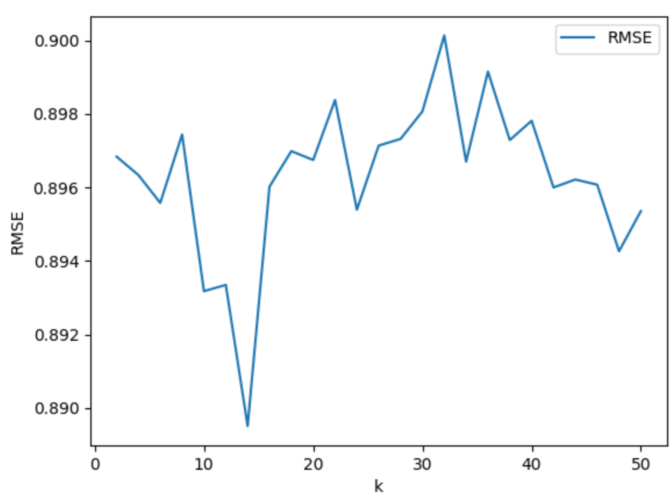
The minimum average MAE is 0.6817322111058285

Question 26



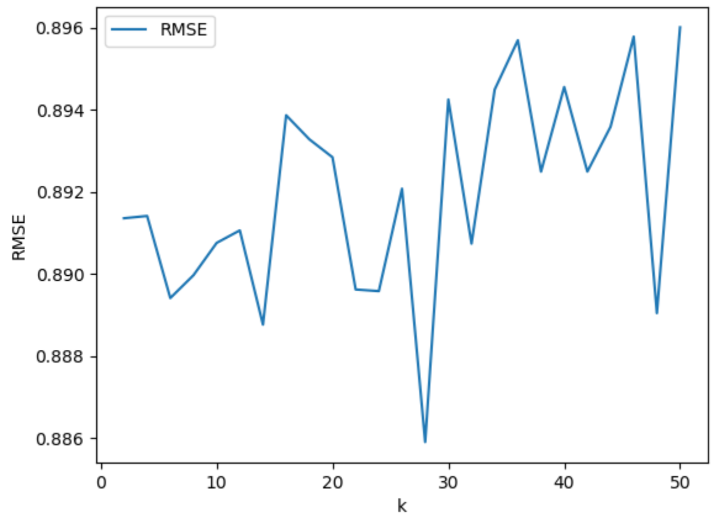
The minimum average RMSE is 0.8798719999790535

Question 27



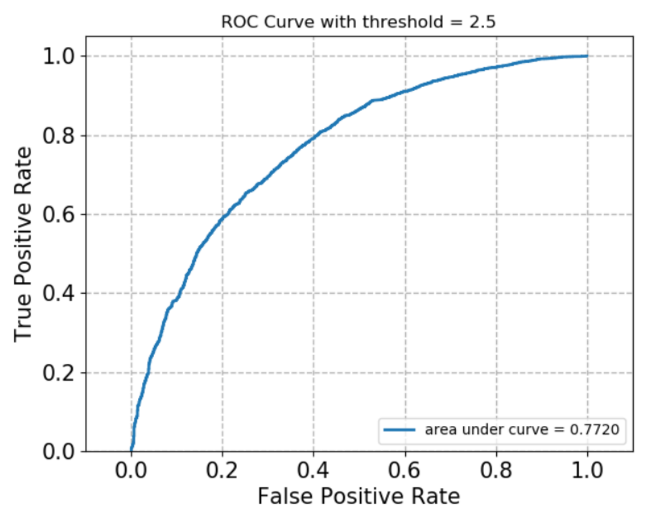
The minimum average RMSE is 0.8895055358754014

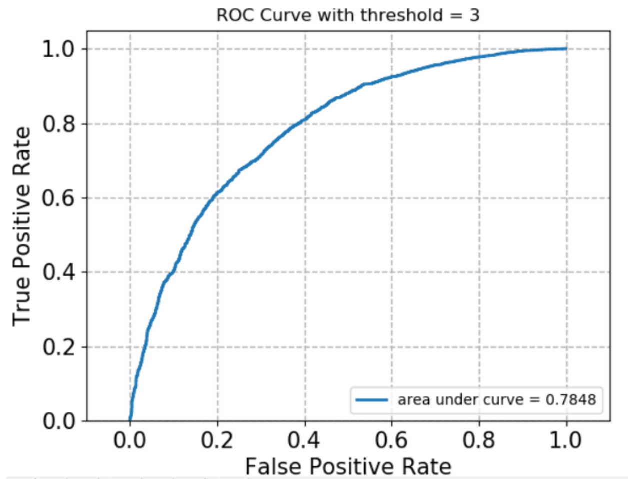
Question 28

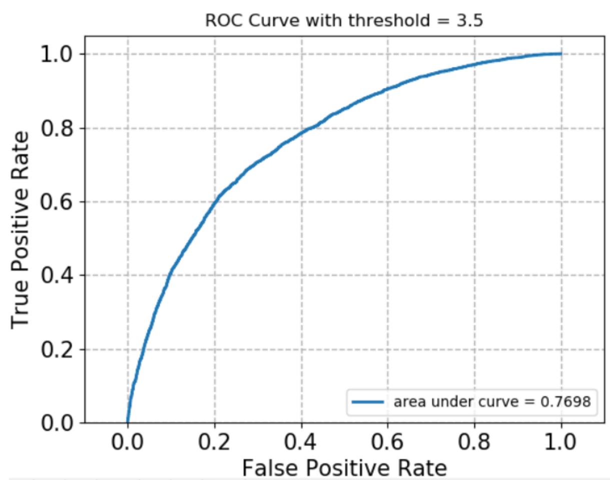


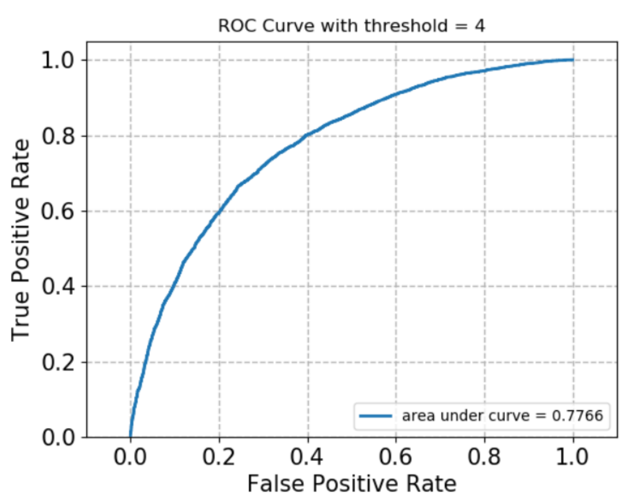
The minimum average RMSE is 0.8859112447396891

Question 29

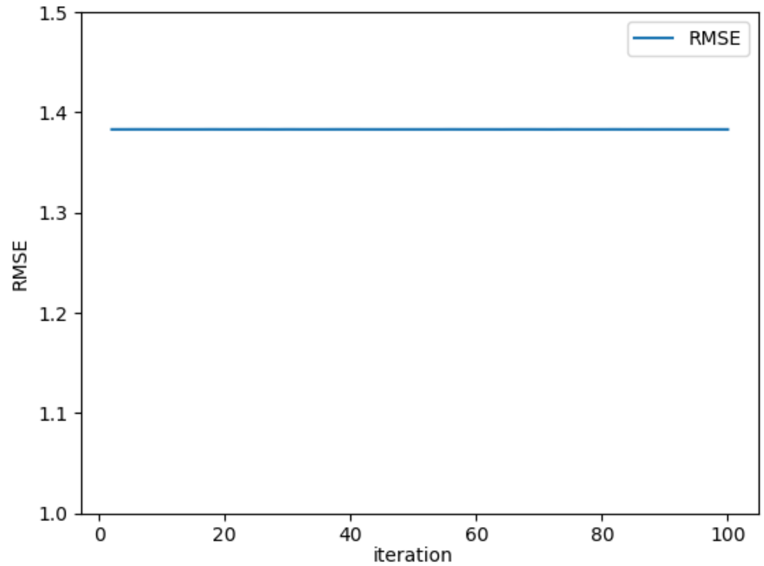



ROC Curve with threshold = 3
True Positive Rate
False Positive Rate
area under curve = 0.7848
0.0
0.2
0.4
0.6
0.8
1.0


ROC Curve with threshold = 3.5
True Positive Rate
False Positive Rate
area under curve = 0.7698
0.0
0.2
0.4
0.6
0.8
1.0

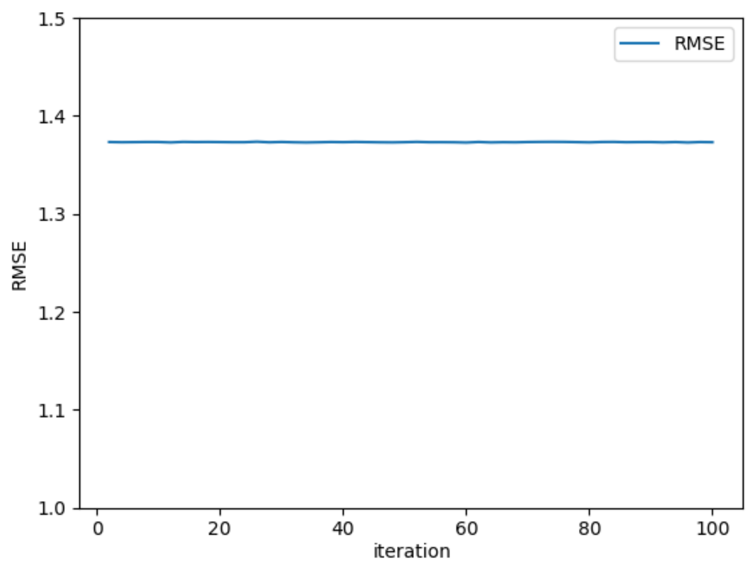

ROC Curve with threshold = 4
True Positive Rate
False Positive Rate
area under curve = 0.7766
0.0
0.2
0.4
0.6
0.8
1.0

Question 30



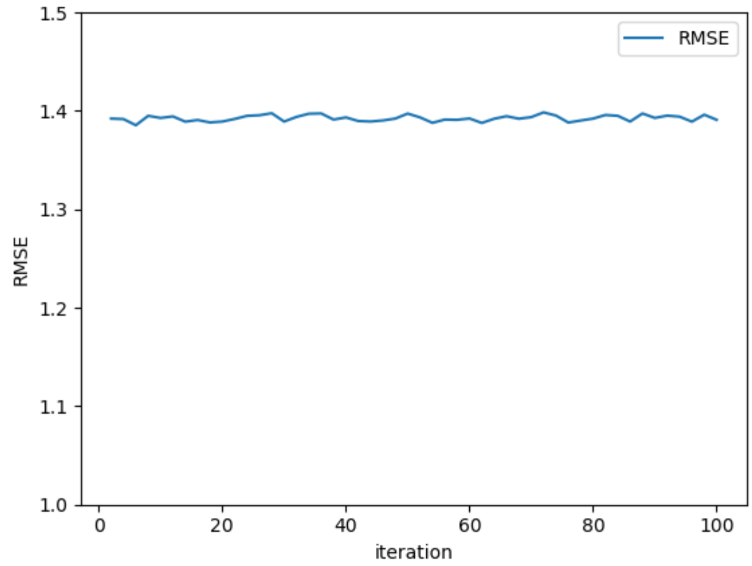
Average RMSE = 1.383

Question 31



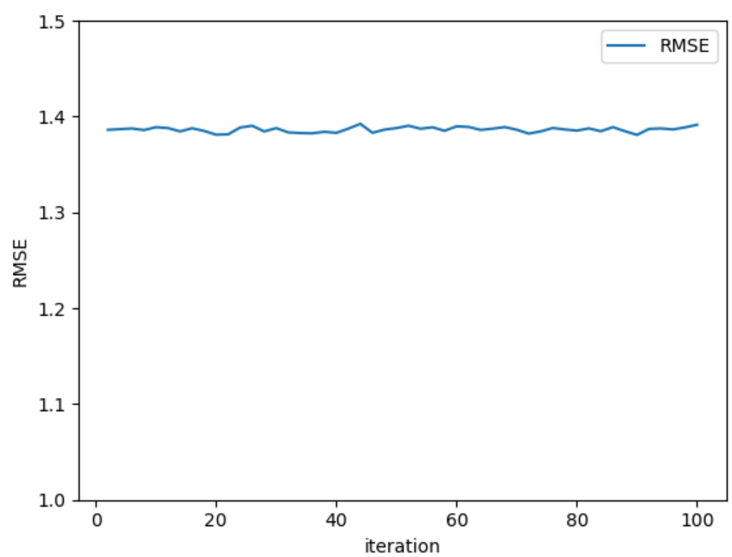
Average RMSE = 1.373

Question 32



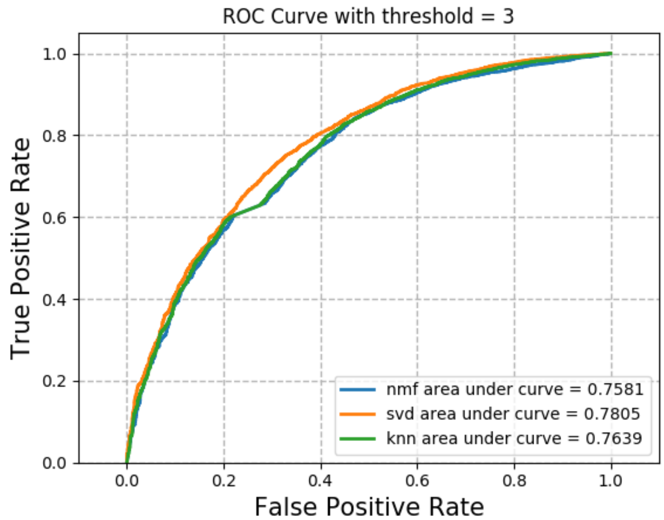
Average RMSE = 1.3925

Question 33



Average RMSE = 1.38644

Question 34



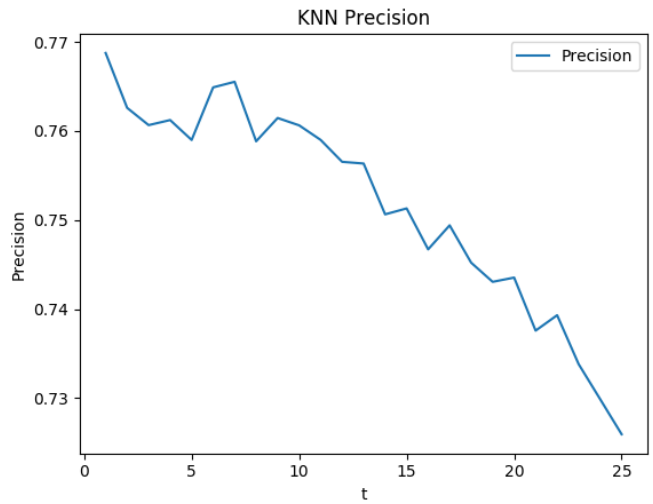
From the above results we can see that the matrix factorization with SVD has greatest area under curve, then k-nearest neighbors algorithm has the second largest area, and finally the matrix factorization with NMF has the smallest area. We conclude that SVD has the best overall performance among the three methods. The optimal point on the ROC curve for SVD is when the false positive rate is around 0.3 and true positive rate around 0.75, which is the most top-left point on the graph.

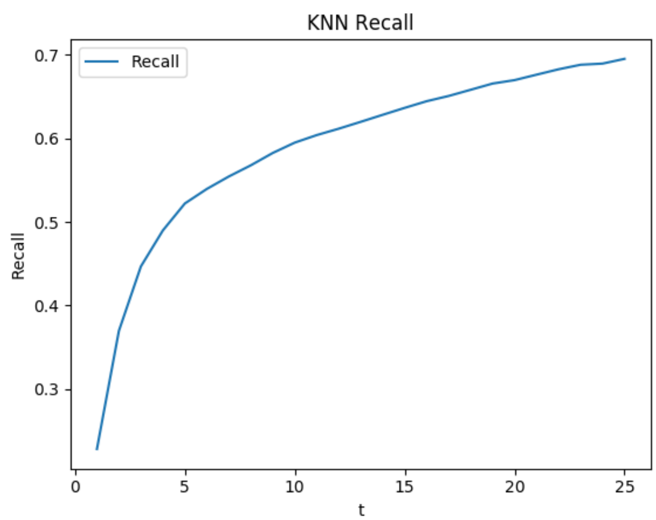
Question 35

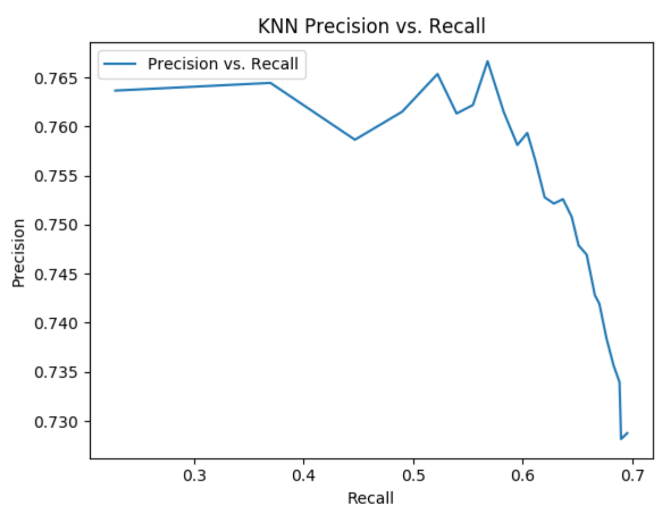
Precision describes how the accurate of the recommendation system is. It is the ratio of items that a user likes to the total items recommended.

Recall describes the percentage of recommended items that a user likes over all items liked by the user.

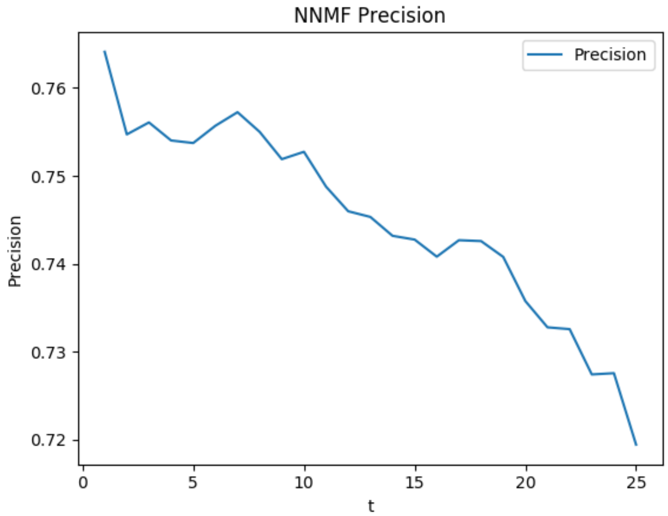
Question 36

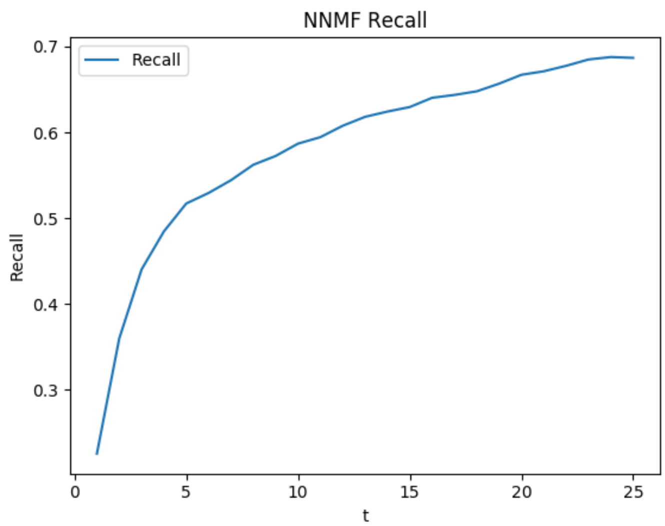

KNN Precision
Precision
0.77
0.76
0.75
0.74
0.73
Precision
0
5
10
15
20
25
t

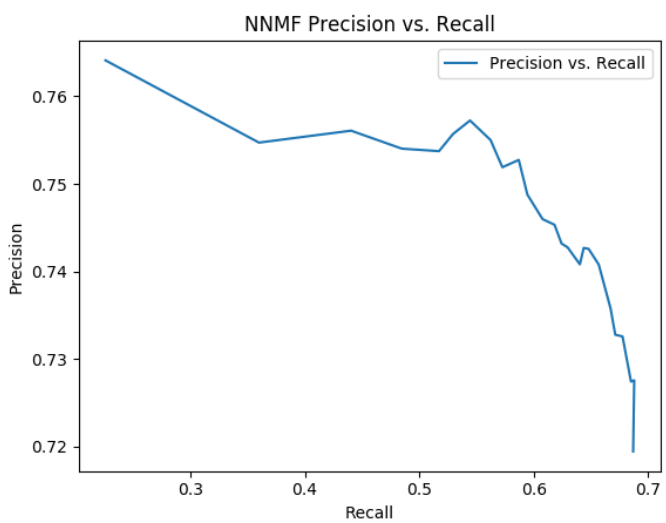

KNN Recall
Recall
0.7
0.6
0.5
0.4
0.3
Recall
0
5
10
15
20
25
t


KNN Precision vs. Recall
Precision vs. Recall
0.765
0.760
0.755
0.750
0.745
0.740
0.735
0.730
Precision
0.3
0.4
0.5
0.6
0.7
Recall

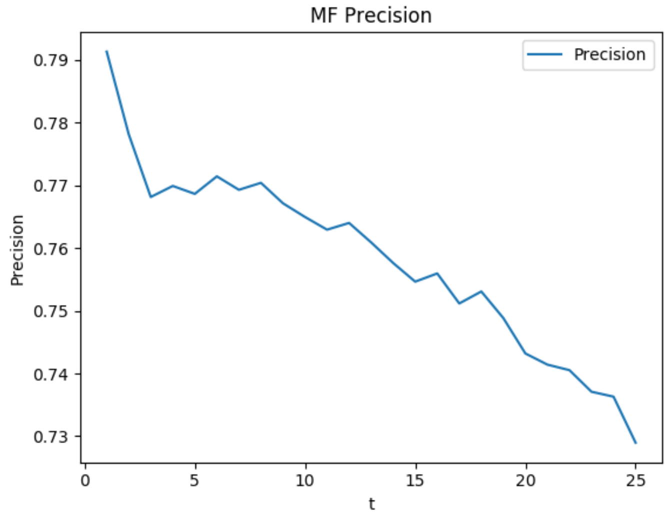
Question 37

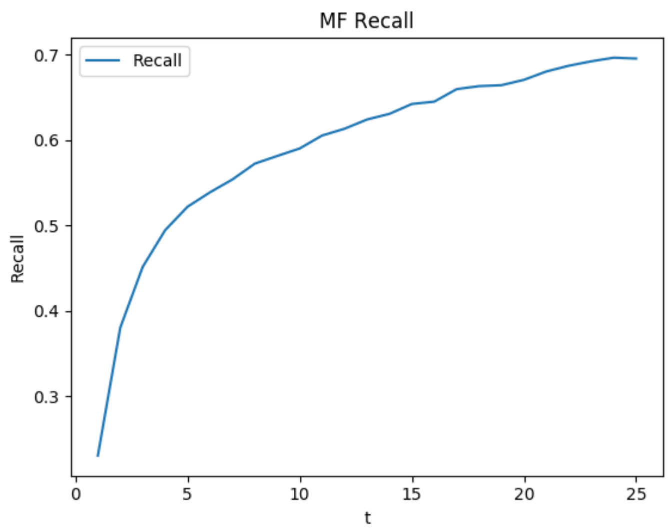

NNMF Precision
Precision
0.76
0.75
0.74
0.73
0.72
Precision
0
5
10
15
20
25
t

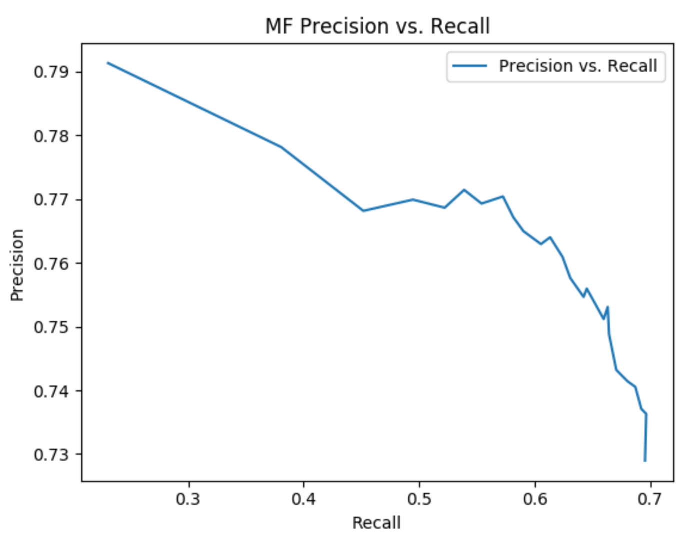

NNMF Recall
Recall
0.7
0.6
0.5
0.4
0.3
Recall
0
5
10
15
20
25
t


NNMF Precision vs. Recall
Precision vs. Recall
0.76
0.75
0.74
0.73
0.72
Precision
0.3
0.4
0.5
0.6
0.7
Recall

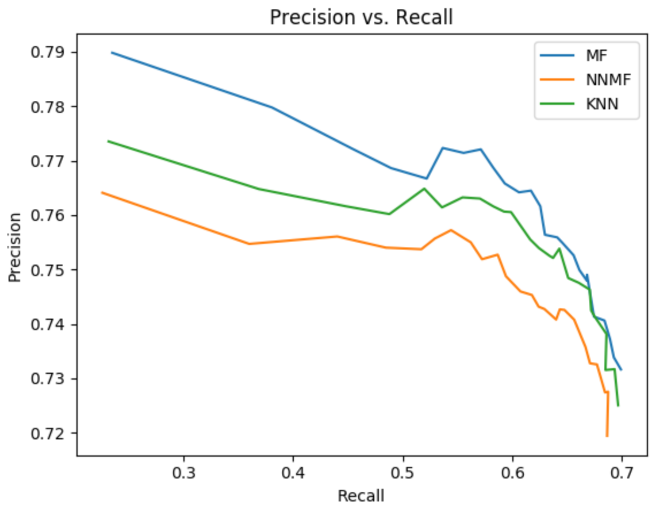
Question 38


MF Precision
Precision
Precision
t
0.79
0.78
0.77
0.76
0.75
0.74
0.73
0
5
10
15
20
25


MF Recall
Recall
Recall
t
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25


MF Precision vs. Recall
Precision vs. Recall
Precision
Recall
0.79
0.78
0.77
0.76
0.75
0.74
0.73
0.3
0.4
0.5
0.6
0.7

Question 39



From the precision vs. recall figure above, we can see that matrix factorization using SVD generates the best result, while the second best is the k-nearest neighbor algorithm. The last is the non-negative matrix factorization using NMF algorithm. The precision vs. recall curve shows the tradeoff between precision and recall for different threshold. The greater the area under the curve, the better the prediction result is. A large area represents both high precision and high recall. Therefore, from our experiment result, it is better to use matrix factorization through SVD to recommend movies to user. However, one thing to notice is that there is some overlap of the curves between the matrix factorization and k-nearest neighbor algorithm. If the dataset is different, or when we change specifications, KNN could be better than MF when the recall is around 0.7. A possible future work is to investigate such variations.