# EE219 Large Scale Data Mining Models and Algorithms

## Winter 2018_Project3


Team member:

Dui Lin (504759948)

Xinyi Jiang (904818856)

Zhenli Jiang (304878235)

# Content

# 1. Introduction

In this project, we built a recommendation system with collaborative filtering models on MovieLens datasets. Collaborative filtering models use the collaborative power of the ratings provided by multiple users to make recommendations. The main challenge in designing collaborative filtering methods is that the underlying ratings matrices are sparse. Consider an example of a movie application in which users specify ratings indicating their like or dislike of specific movies. Most users would have viewed only a small fraction of the large universe of available movies and as a result most of the ratings are unspecified.

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. Most of the collaborative filtering methods focuses on leveraging either inter-item correlations or inter-user correlations for the prediction process. In this project, we will implement and analyze the performance of two types of collaborative filtering methods:

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

We also explore techniques for recommending the top k favorite movies to users, which can be addressed as a ranking version of recommendation system problem.

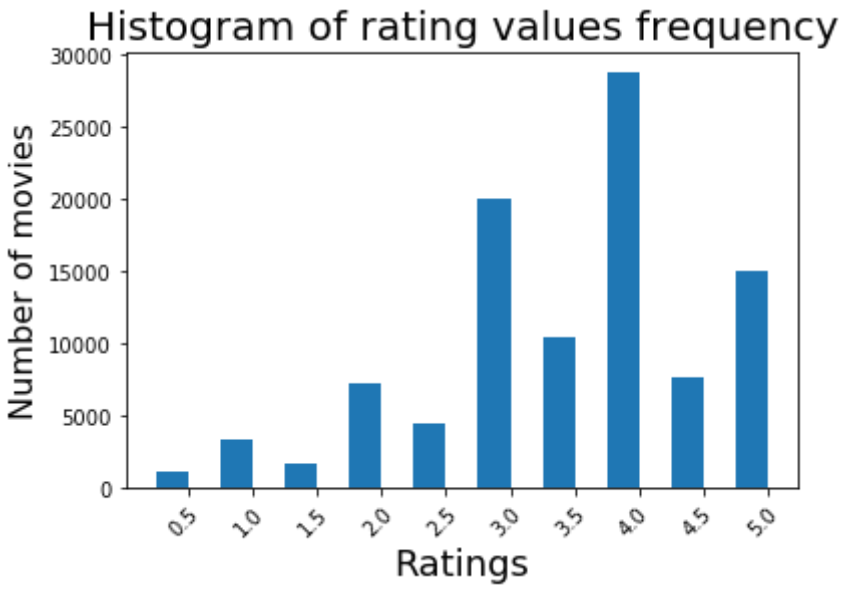
# 2. Problem Statement and Results

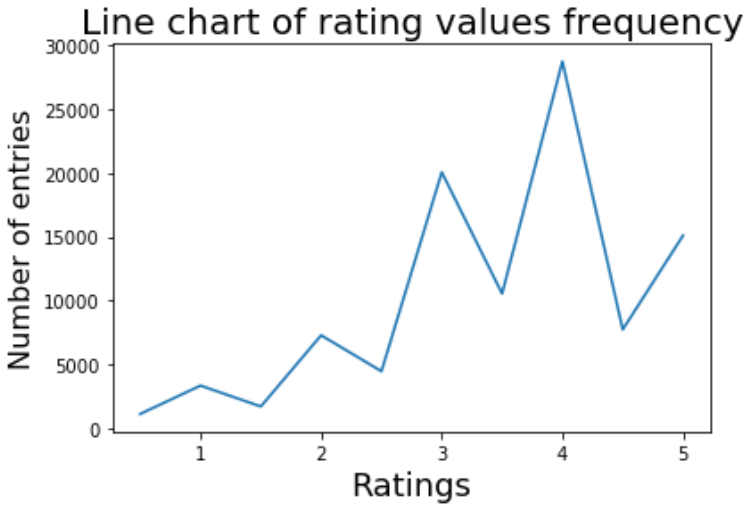
## 2.1 Question 1

With the given equation, the sparsity of the dataset is 0.016439141608663475. The dataset matrix shape is [671, 9066].

## 2.2 Question 2

The histogram and line chart of the frequency of the rating values are as following:

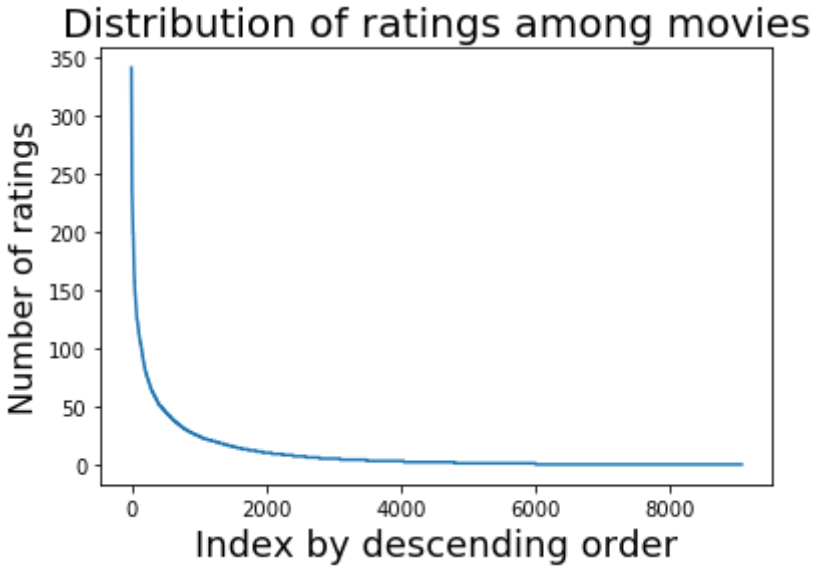


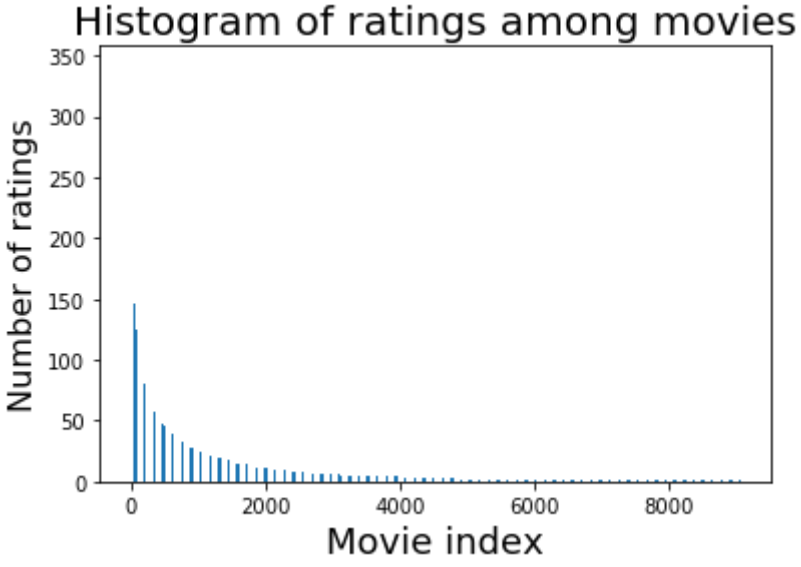



In the plot above, the number of sparse (0 entries) is omitted since it dwarfs the other values by more than an order of magnitude. And we could see that most of the ratings are 4 star, but the lower ratings in the range of 0.5-2.5 bring this average down.

## 2.3 Question 3

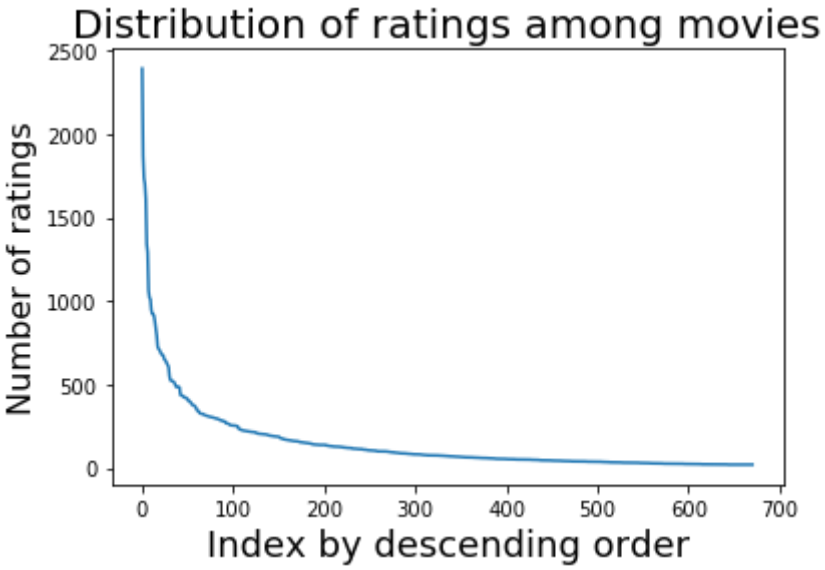
The histogram and line chart of the distribution of ratings are as following, the x-axis is ordered by movie index:

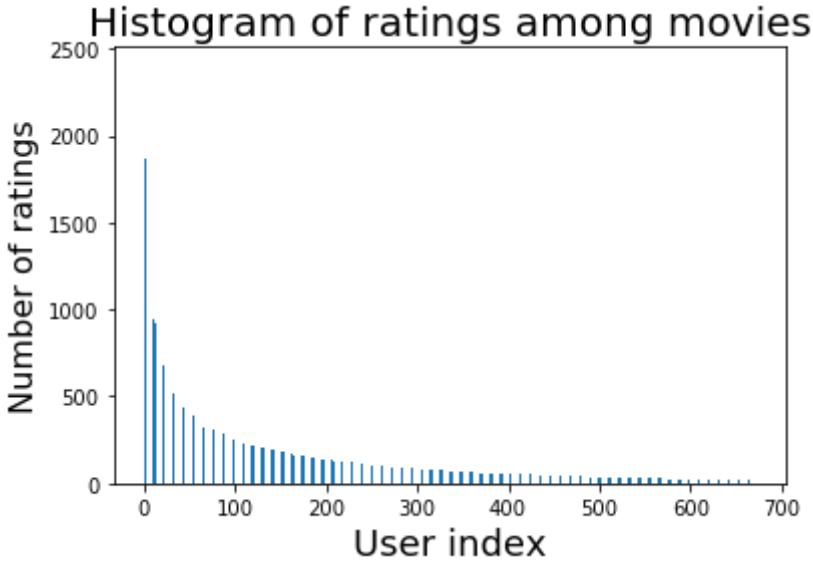


## 2.4 Question 4

The histogram and line chart of the distribution of ratings are as following:




## 2.5 Question 5

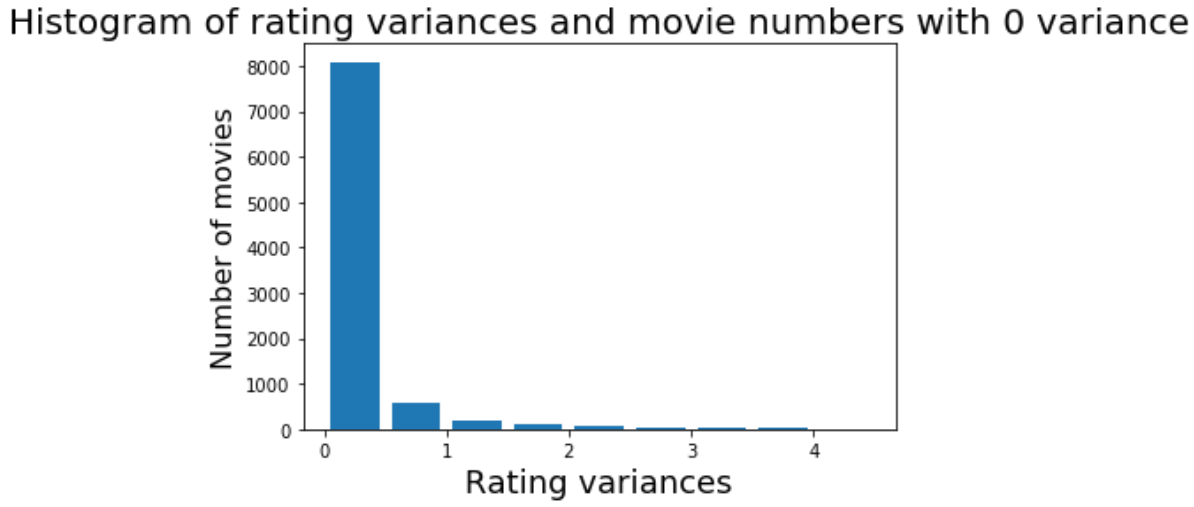
We can observe from the distribution in question_3 that some movies are much more heavily reviewed than others. Since a user cannot review the same movie twice, we can assume that in the case where a movie has 671 reviews and there are 671 users, each user has reviewed the movie. Therefore, if we know our population size, the number of reviews that a movie has

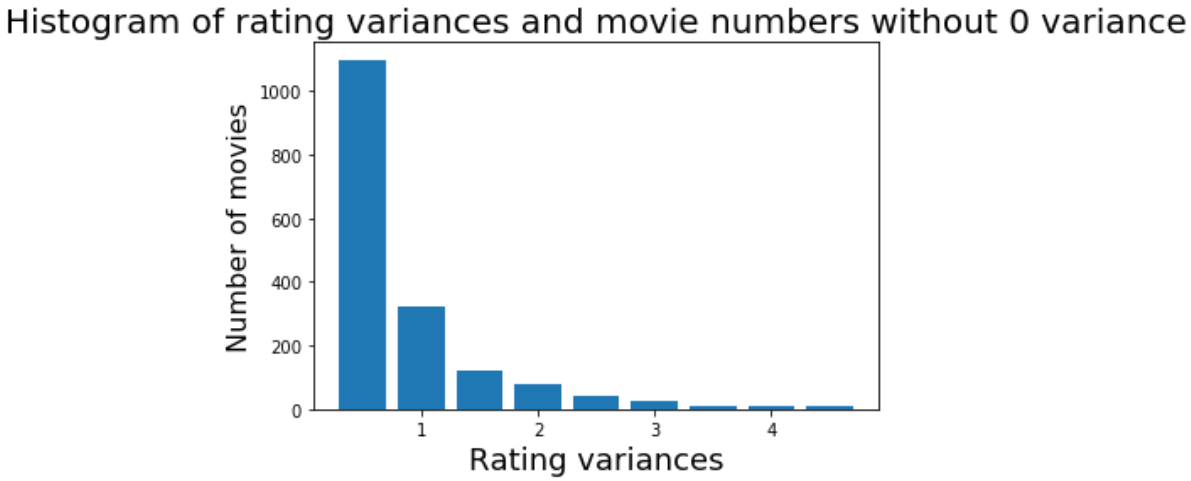
makes it a good indicator of population-wide preference. With collaborative filtering, we want to use commonalities in order to perform interpolation and fill in gaps. Since there is a lot of sparsity in this data, one reduction technique could be to remove movies that fall below a certain threshold of number of reviews. Movies with fewer reviews will cover less of the population spread and be less effective indicators of preference in the collaborative algorithm.

Concretely, question_3's plot shows that one movie had nearly 200 reviews, which spans approximately 29% of the population. Other movies had nearly 150 reviews, or nearly 22% of the population. Thus, these two movies would be excellent choices if we were required to only choose two members of the dataset.

## 2.6 Question 6

In this question, we firstly compute the variance of the rating values received by each movie. Then we plot two histograms, one of which contains 0 variance and the other neglect 0 variance, since 0 variance are prevailing. Here are the two plots:






## 2.7 Question 7

By summing up all of the user's ratings. The denominator is the length of the set containing indices for the user's specified ratings.

$$\frac{\Sigma_{i \in I_u} r_{uk}}{|I_u|} \forall u \in 1....m$$

## 2.8 Question 8

$I_u \cap I_v$ represents the intersection of the two sets of indices for users u and v. Concretely, this intersection is the set containing the indices of items both users rated. It's possible for the intersection to be the empty set $\varnothing$ if two users have not rated any of the same movies.

For example, let's say two users rate one movie each. User 1 rates Batman and User 2 rates Star Wars. The intersection of indices would be the empty set

$$(I_u \cap I_v = \varnothing).$$

## 2.9 Question 9

In this project we use mean centering as a form of normalization. Without normalization, the approach would be to define a match or peer group as the set consisting of the k-nearest neighbors given highest Pearson correlation coefficient. However, since we're iterating over

items that don't necessarily have the same number of ratings per user, there will be differences from iteration to iteration. If we return the weighted average of the ratings as the predicted rating for an item, an especially sparsely rated value (one rated by a single user) can be skewed if the user is biased. For example, a user may rate every movie with 5 stars. Conversely, a user may rate every movie with 0.5 stars. A biased user such as this would skew a sparse row of ratings and offset the prediction. Mean subtraction is performed in order to minimize the skew these types of users can have on predictions.

## 2.10 Question 10

Design a k-NN collaborative filter and sweep k from 2 to 100 in step size of 2. Plot the average RMSE and average MAE against k.

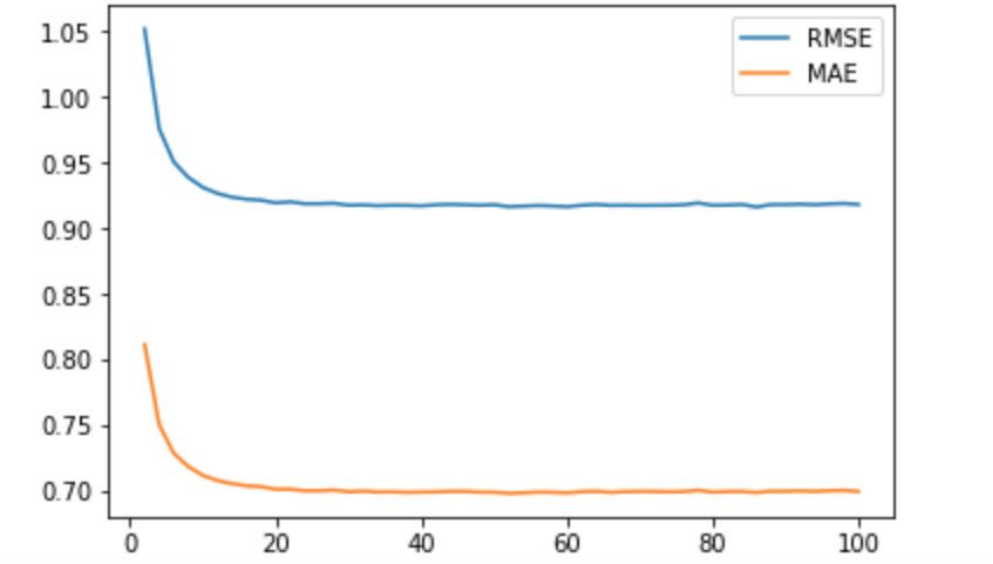



Figure Q11 Average RMSE and Average MAE against k

## 2.11 Question 11

According to the figure of question 10, it can be seen that the two plots converges to a steady-state value when k is around 22.

Thus We can concluded that the ‘minimum k’ equals 22.

## 2.12 Question 12

Design a k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set. Sweep k from 2 to 100 in step sizes of 2.

Here is the plot of the average RMSE and the minimum average RMSE.

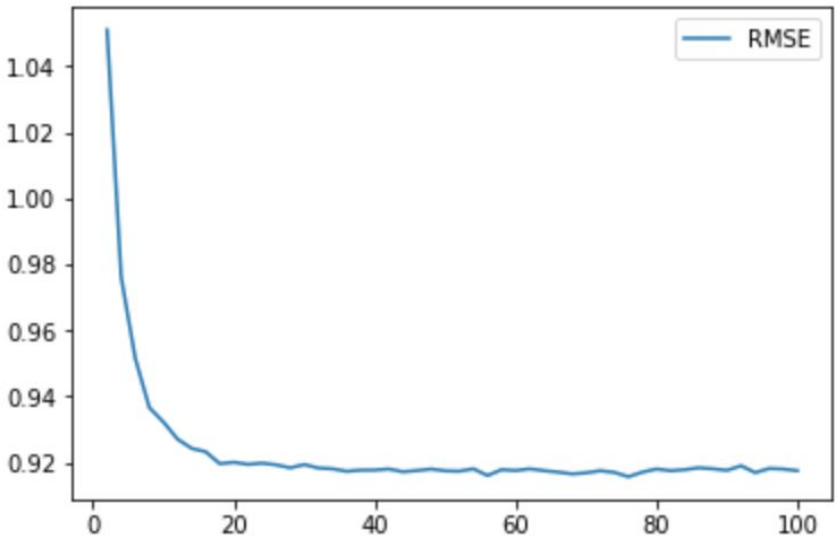



Figure Q12. Average RMSE against k in popular moive set

```
Minimum average RMSE for popular list=  0.915729862039
```

## 2.13 Question 13

Design a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set. Sweep k from 2 to 100 in step sizes of 2.

Here is the plot of the average RMSE and the minimum average RMSE.

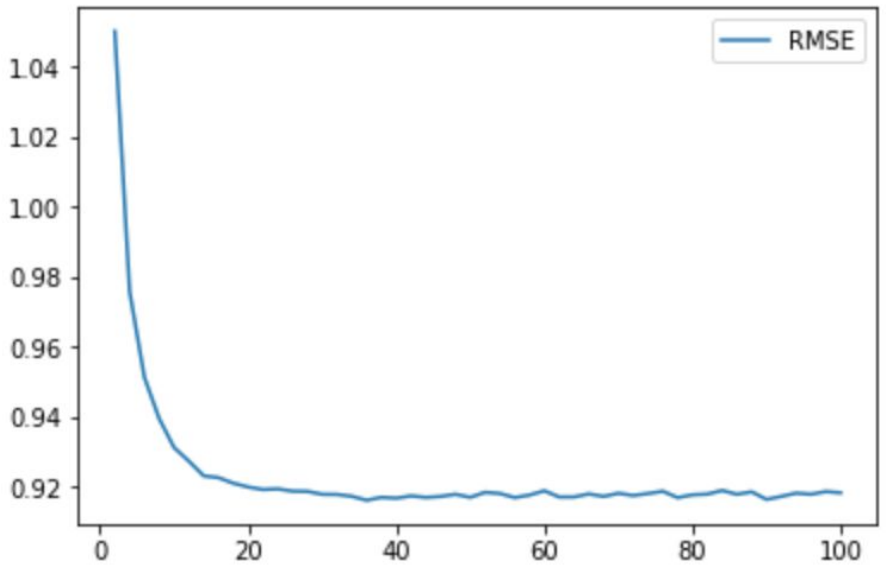



Figure Q12. Average RMSE against k in the unpopular movie set

```
Minimum average RMSE for unpopular list =  0.916178820976
```

## 2.14 Question 14

Design a k-NN collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set. Sweep k from 2 to 100 in step sizes of 2.

Here is the plot of the average RMSE and the minimum average RMSE.

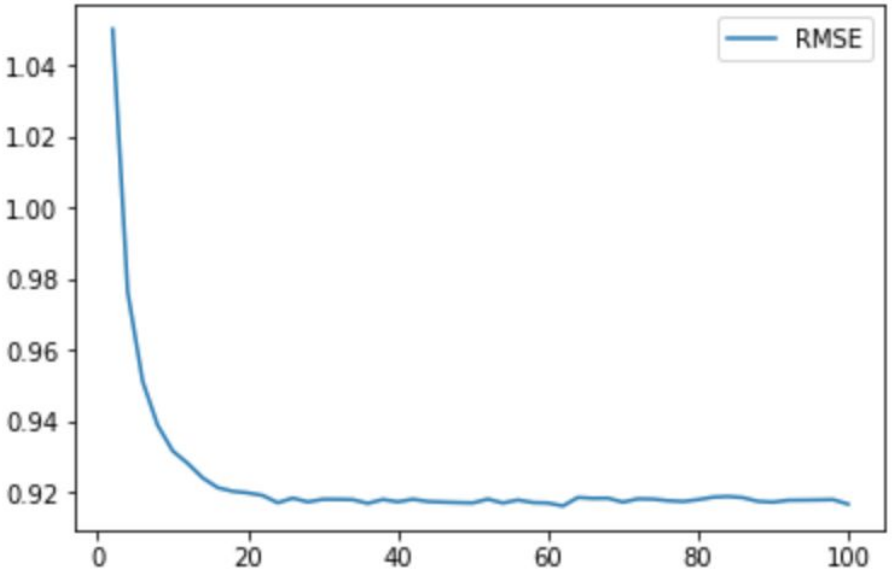


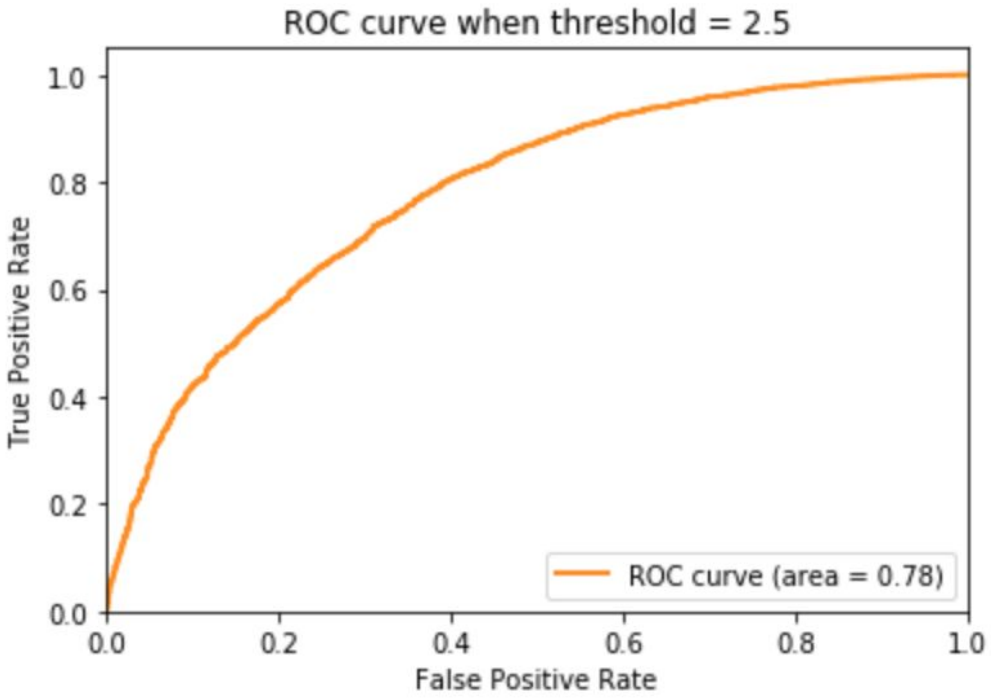

Figure Q12. Average RMSE against k in the high variance set

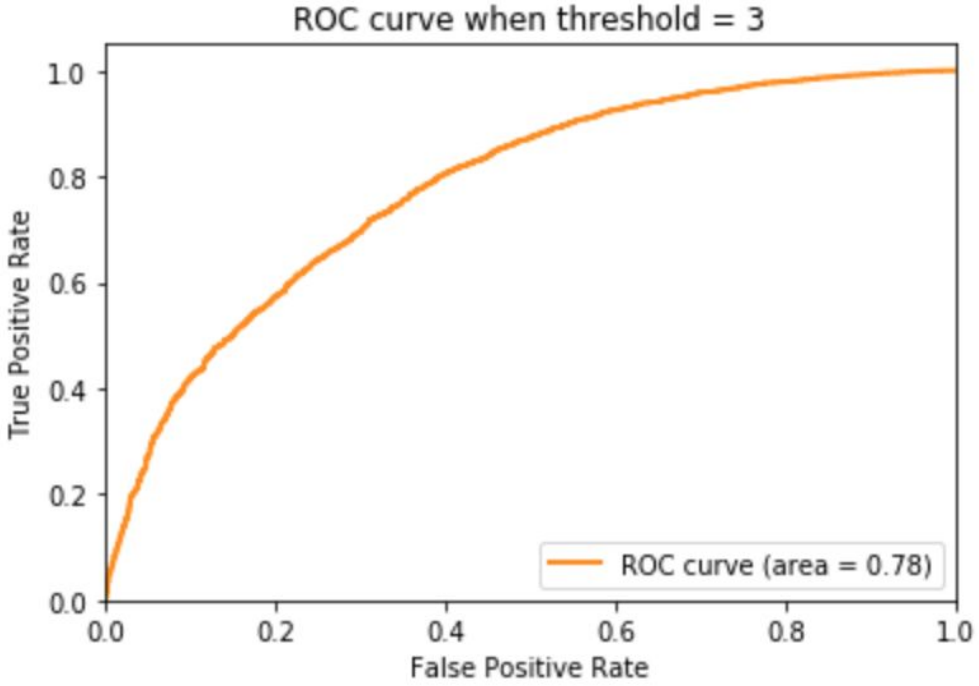
```
Minimum average RMSE for high_variance_list =  0.91623007
5784
```

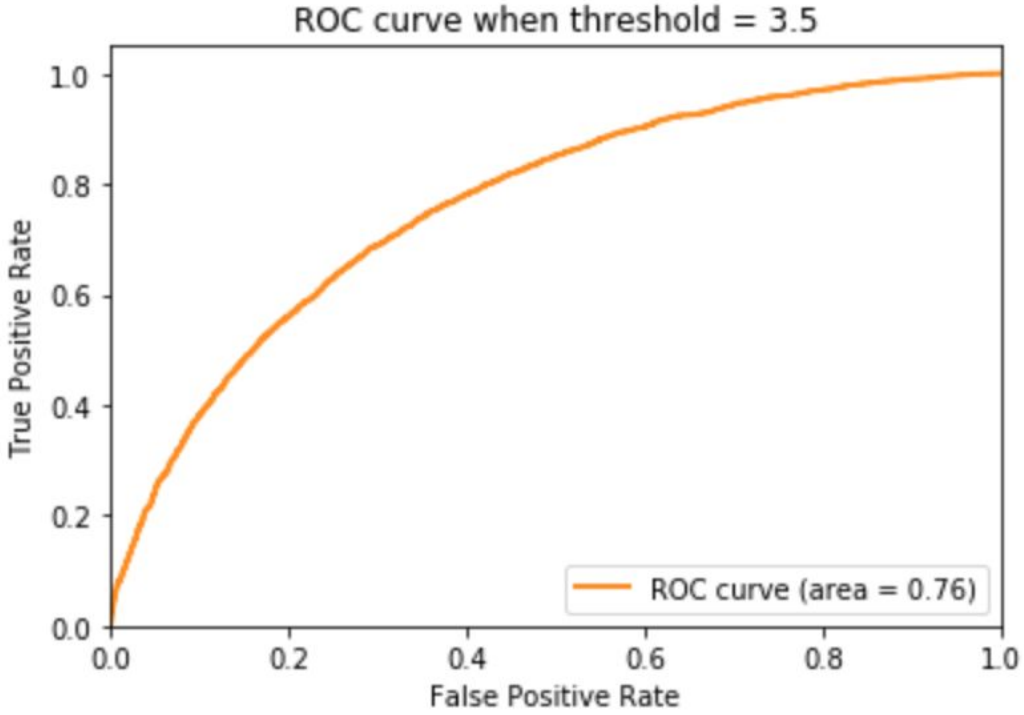
## 2.15 Question 15

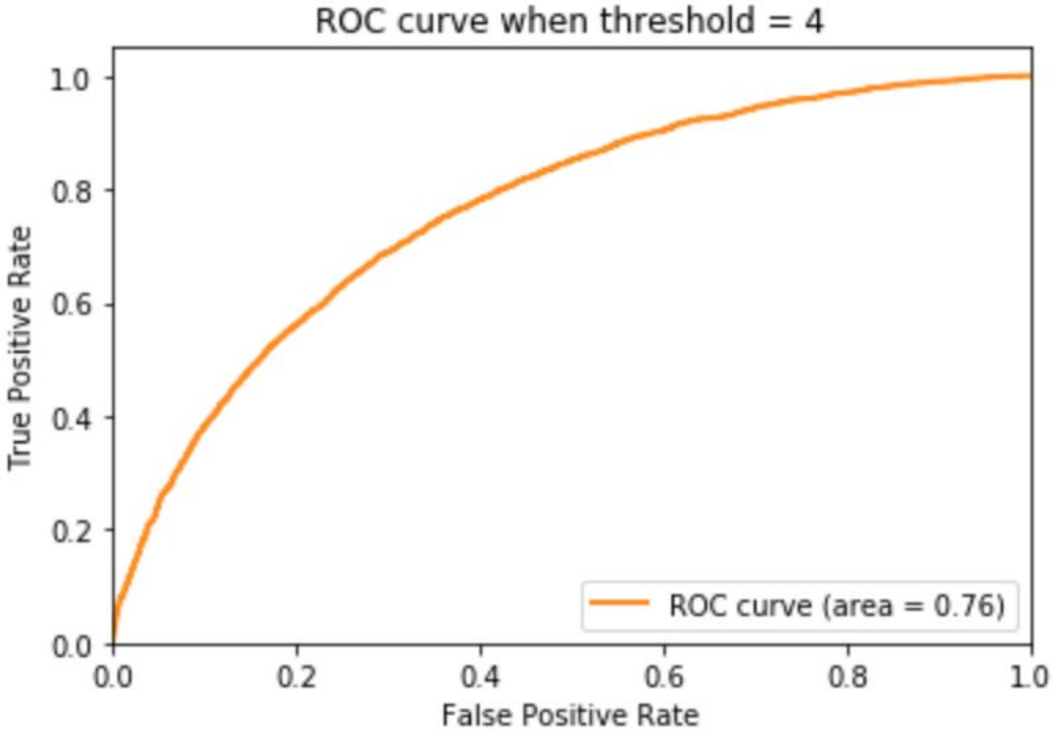
In this question, we are going to plot the ROC curves for the three kinds of k-NN collaborative filters for threshold value (2.5,3,3.5,4).

Minimum k = 22




ROC curve when threshold = 3
True Positive Rate
False Positive Rate
ROC curve (area = 0.78)
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve when threshold = 3.5
True Positive Rate
False Positive Rate
ROC curve (area = 0.76)
0.0
0.2
0.4
0.6
0.8
1.0

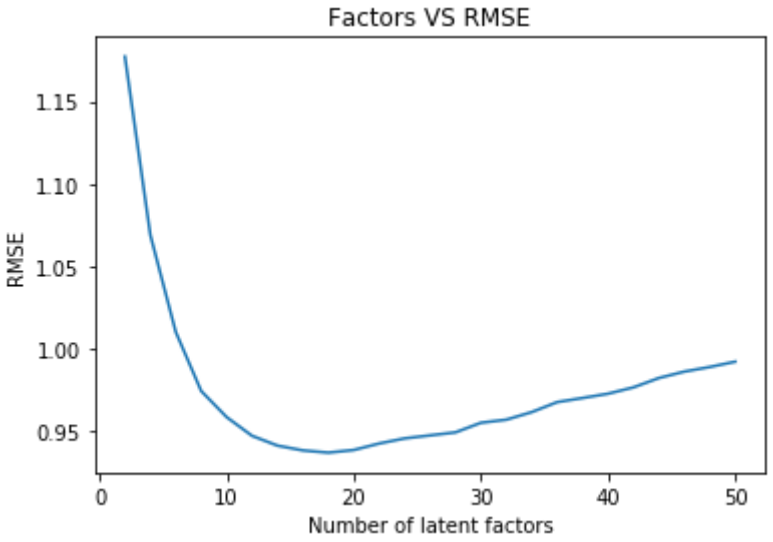


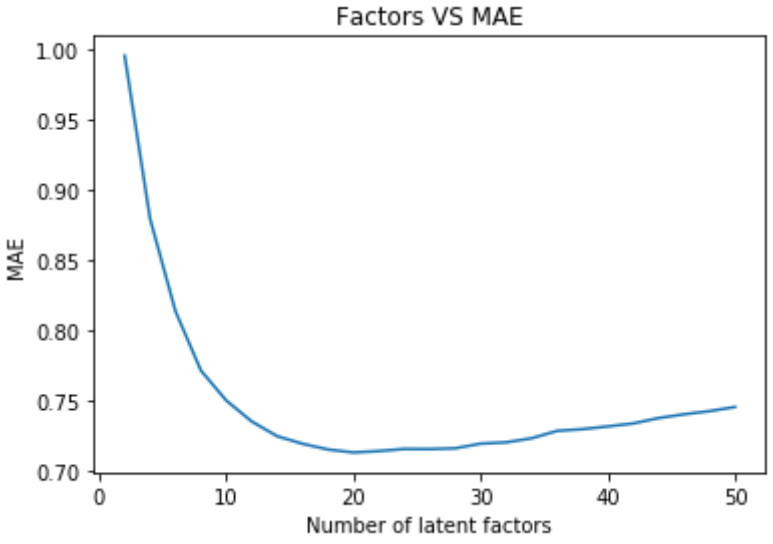

## 2.16 Question 16

If we fix U and look at each line of V, it's the square estimate of linear regression. Therefore, the optimization problem given by equation 5 should be convex with U fixed.

## 2.17 Question 17

We did the prediction using NNMF. We tried the number of latent factors from 2 to 50 with the step size of 2 and 10-fold cross validation. The figures below shows the RMSE value and MAE value vs number of latent factors.




## 2.18 Question 18

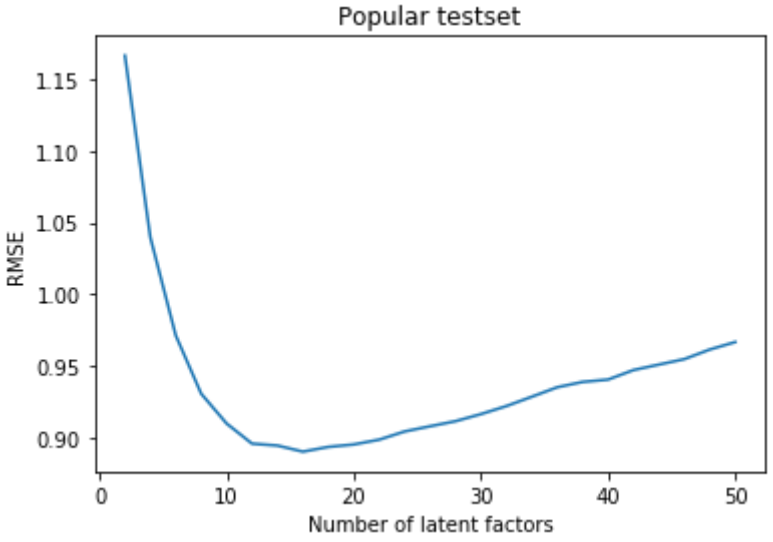
The minium RMSE: 0.9372 when factors = 18

The minimun MAE: 0.713 when factors = 20

The optimal number of latent factors is about 20. Since there are 18 genres in the movie database, the number of latent factors is almost the same as the number of genres.

## 2.19 Question 19

The minimum average RMSE = 0.8903 when factors = 16

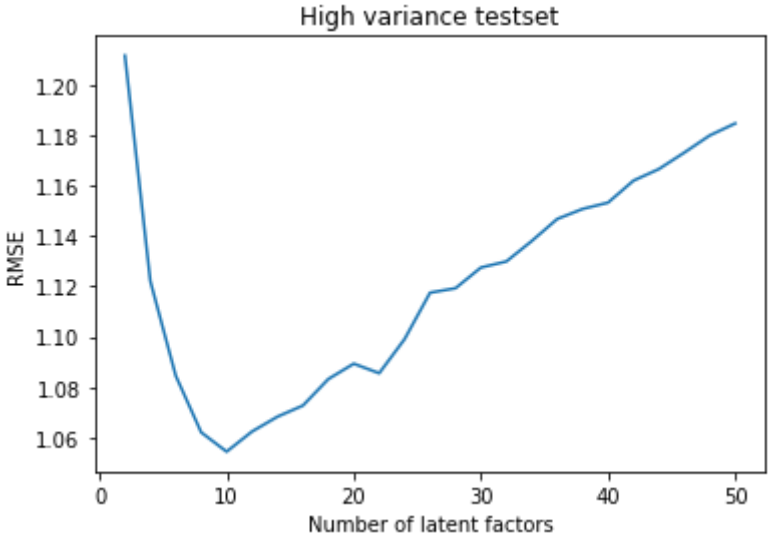



## 2.20 Question 20

The minimum average RMSE = 1.029 when factors = 20


## 2.21 Question 21

The minimum average RMSE = 1.05 when factors = 10




## 2.22 Question 22

AUC:

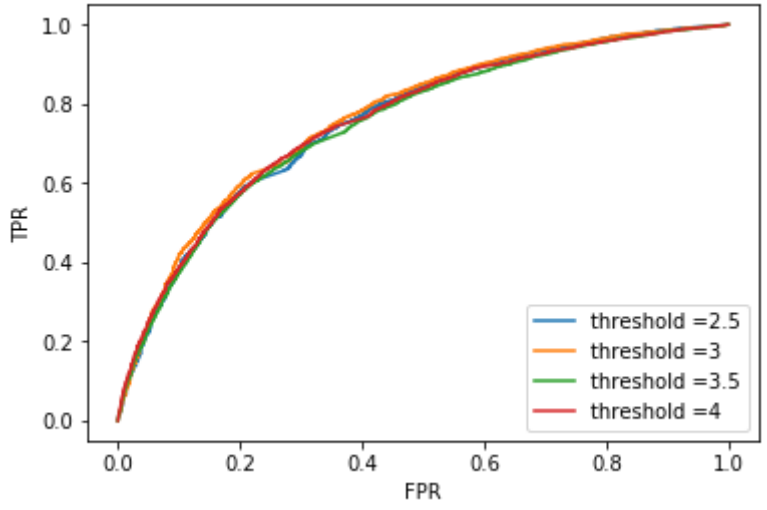
Threshold = 2.5: 0.7572461778596667

**Threshold = 3.0: 0.7662716652722009**

Threshold = 3.5: 0.7508049364414757

Threshold = 4.0  0.7585464956007213

We can see that when threshold = 3.0, the value of AUC is maximized.



## 2.23 Question 23

We use the NMF function in sklearn to decompose the matrix instead of using the package suprise.

If we look at the first 10 columns the original results are listed below), which means the first 10 latent factors, and check the genres of the top 10 movies in each column:

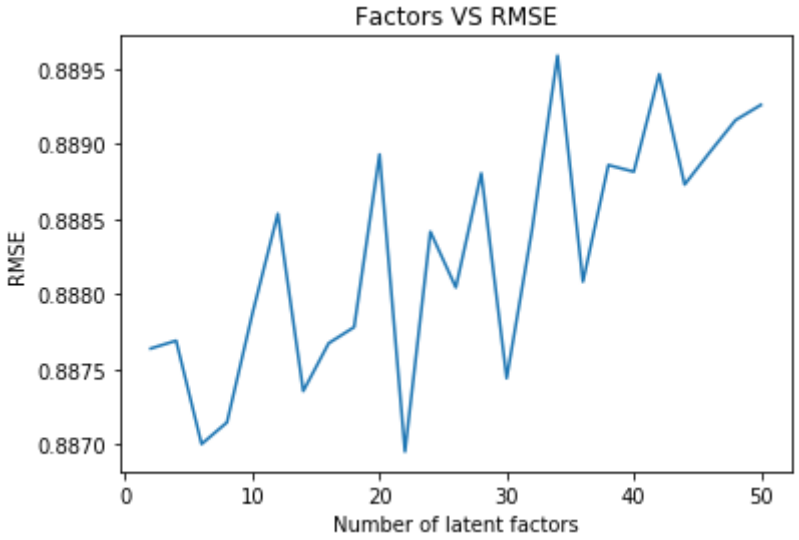
- since most movies belong to the genre drama, so we ignore the effect this genre
- Movies in several columns seem to belong to the same genre, such as: col 1 Fantasy, col 3 Adventure, col 5 Horror/Thriller, col 6 Action/Adventure, col 7 children, col 8 Horror/Thriller, col 9 Action/Adventure
- However, there are also some columns doesn't shown significant patterns, such as col 2,4,10
- We can say that most of the latent factors have close connections with the movie genres while others don't show some connections..

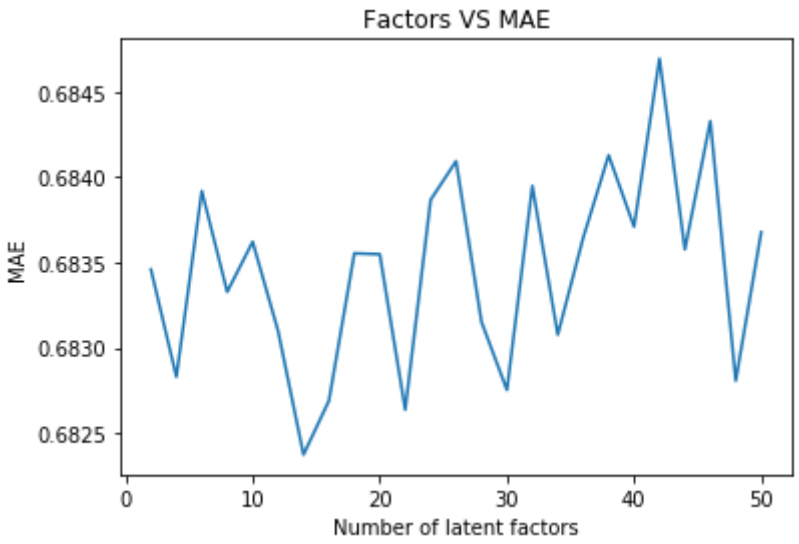
['Adventure|Fantasy', 'Adventure|Fantasy', 'Action|Adventure|Drama|Fantasy',
'Adventure|Animation|Children|Comedy|Fantasy|Romance', 'Adventure|Animation|Children|Comedy|Fantasy',
'Action|Adventure|Comedy|Fantasy', 'Adventure|Animation|Children|Comedy',
'Action|Adventure|Animation|Children|Comedy', 'Crime|Drama', 'Action|Sci-Fi|Thriller']
['Drama|Romance', 'Drama|Mystery', 'Action|Adventure|Mystery|Romance|Thriller', 'Film-Noir|Mystery',
'Comedy|Romance', 'Drama', 'Comedy|Drama|Romance', 'Mystery|Thriller', 'Comedy|War', 'Crime|Drama']
['Comedy|Drama|Romance|War', 'Thriller', 'Adventure|Drama|Western', 'Adventure|Drama|IMAX',
'Comedy|Crime|Drama|Thriller', 'Action|Adventure|Sci-Fi|Thriller',
'Action|Adventure|Comedy|Romance|Thriller', 'Crime|Horror|Thriller', 'Crime|Drama', 'Action|Drama|War']
['Drama', 'Comedy|Drama', 'Adventure|Drama|Western', 'Drama', 'Action|Drama|Romance|War',
'Children|Drama|Fantasy', 'Drama|Romance|War', 'Drama', 'Comedy|Drama|War',
'Adventure|Animation|Comedy|Fantasy|Musical']
['Comedy', 'Adventure|Children|Fantasy|Musical', 'Drama', 'Comedy|Musical|Romance',
'Drama|Musical|Romance', 'Comedy|Drama', 'Drama', 'Drama', 'Drama|Romance', 'Documentary']
['Drama|Horror|Thriller', 'Horror', 'Action|Crime|Thriller', 'Adventure|Drama|Fantasy|Mystery|Sci-Fi',
'Drama|Sci-Fi|Thriller', 'Comedy|Drama|Romance', 'Comedy|Drama', 'Drama|Mystery|Thriller',
'Adventure|Comedy|Drama', 'Crime|Fantasy|Horror']
['Action|Crime|Drama|IMAX', 'Action|Adventure|Sci-Fi', 'Action|Crime|Drama|Mystery|Sci-Fi|Thriller|IMAX',

'Action|Adventure|Sci-Fi|IMAX', 'Action|Adventure|Sci-Fi|IMAX', 'Action|Adventure|Sci-Fi|IMAX',
'Action|Adventure|Drama|Fantasy', 'Adventure|Animation|Children|Romance|Sci-Fi',
'Action|Crime|Mystery|Thriller', 'Adventure|Fantasy']
['Animation|Children|Comedy|Musical|Romance', 'Comedy|Drama', 'Comedy',
'Adventure|Animation|Children|Comedy|Musical', 'Comedy|Romance',
'Animation|Children|Fantasy|Musical|Romance|IMAX', 'Adventure|Animation|Children|Comedy',
'Adventure|Animation|Children|Comedy|Fantasy', 'Comedy|Musical|Romance',
'Children|Comedy|Fantasy|Musical']
['Horror', 'Horror', 'Drama|Horror|Thriller', 'Drama|Fantasy|Horror|Thriller', 'Horror|Thriller', 'Horror',
'Horror|Sci-Fi|Thriller', 'Comedy|Horror|Thriller', 'Drama|Horror|Thriller', 'Drama|Romance']
['Action|Adventure|Sci-Fi', 'Action|Adventure|Sci-Fi', 'Action|Adventure|Sci-Fi',
'Action|Adventure|Horror|Sci-Fi', 'Action|Sci-Fi|Thriller', 'Action|Sci-Fi', 'Horror|Sci-Fi', 'Action|Sci-Fi|Thriller',
'Action|Adventure', 'Adventure|Comedy|Sci-Fi']

## 2.24 Question 24

We did the prediction using MF with bias. We tried the number of latent factors from 2 to 50 with the step size of 2 and 10-fold cross validation. The figures below shows the RMSE value and MAE value vs number of latent factors.






## 2.25 Question 25

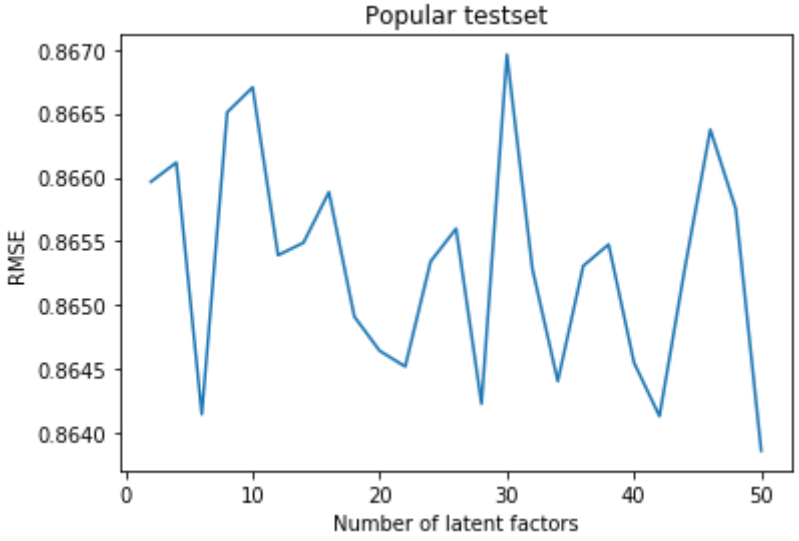
The best RMSE 0.8866 when k = 14

The best MAE 0.6823 when k = 14

The optimal number of latent factors should be 14.

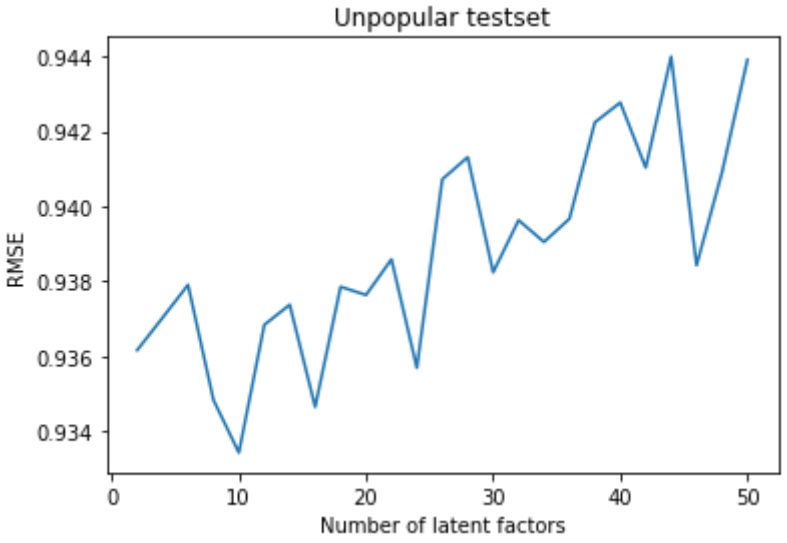
## 2.26 Question 26

Popular subset: The minimal value of RMSE = 0.864 when k = 6



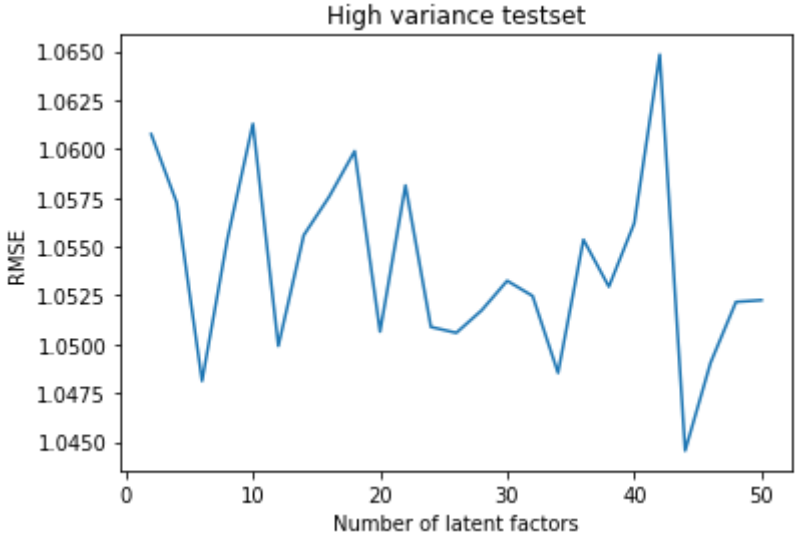
## 2.27 Question 27

Unpopular subset: The minimal value of RMSE = 0.933 when k = 10




## 2.28 Question 28

High variance subset: The minimal value of RMSE = 1.04 when k = 44




## 2.29 Question 29

The value of AUC:

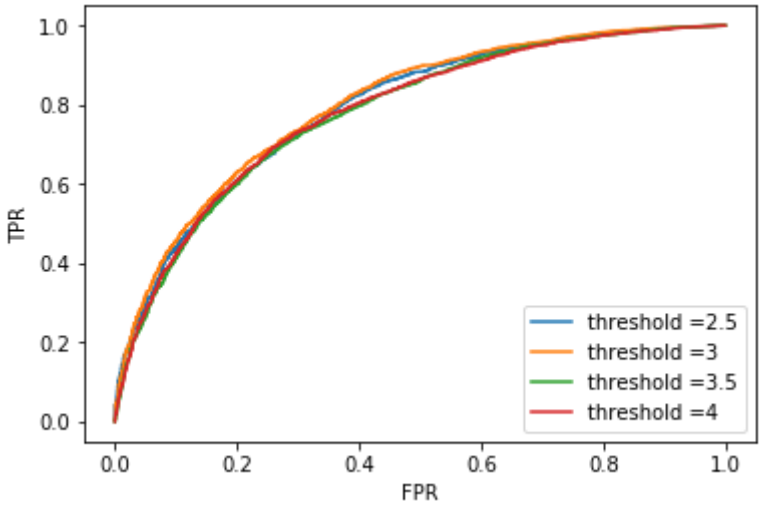
Threshold = 2.5: 0.7914340693097344

**Threshold = 3.0:  0.8000795676022692**

Threshold = 3.5: 0.7800648736108896

Threshold = 4.0  0.7826265064870965

We can see that when threshold = 3.0, the value of AUC is maximized.




## 2.30 Question 30

We use the naive collaborative filtering to do the prediction. After the 10-fold cross validation, the mean RMSE = 0.9624558160124576

## 2.31 Question 31

Popular subset: mean RMSE = 0.94827552

## 2.32 Question 32

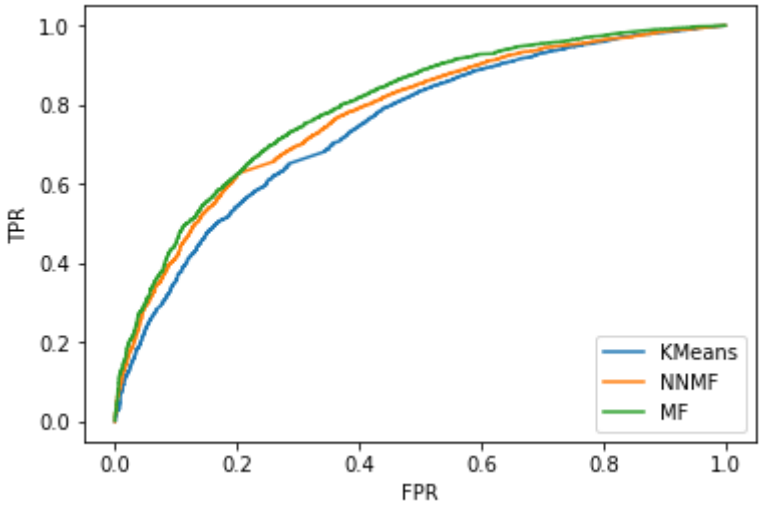
Unpopular subset: mean RMSE = 0.99457736

## 2.33 Question 33

High variance subset: mean RMSE  = 1.13985929

## 2.34 Question 34

The ROC curve:




AUC:
Kmeans: 0.7454362691735876
NMMF: 0.7742141879547884
MF: 0.7952686630231407
We selected the optimal value we found in the previous questions: for k-means, k = 22, for NNMF, k = 20, for MF, k = 14
From the figure we can see that the MF has the best AUC value, which is 0.7952

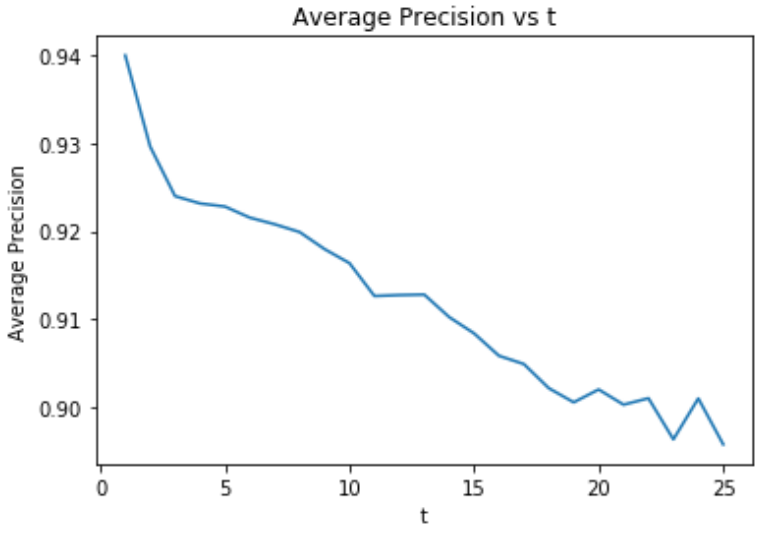
## 2.35 Question 35

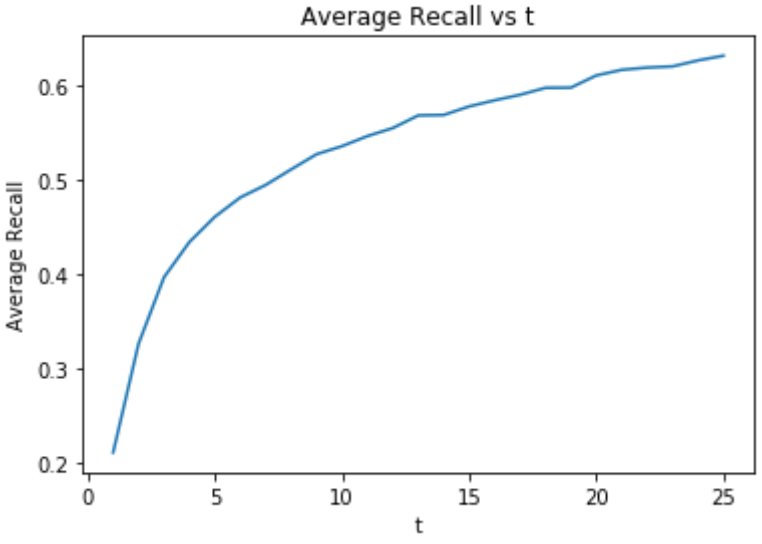
Precision is defined as the ratio between the number of correctly recommended items (because the we know the ground truth that the user has liked this item), and the total number of recommended items. This metric measures how efficient our recommendations are as ideally, we want to only recommend liked items over all our recommendations i.e. keeping the number of recommendations as low as possible.

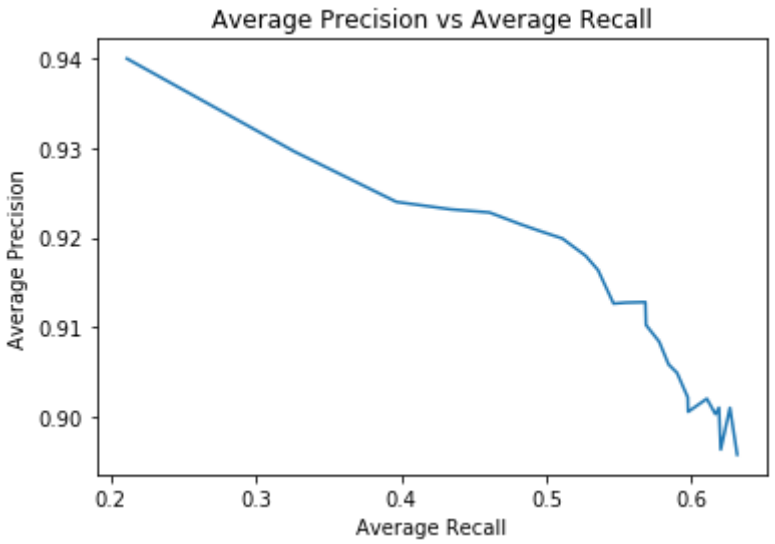
On the other hand, recall is the ratio between the same number of correctly recommended items, and the total number of ground truth items that the user has liked previously. This metric promotes the full coverage of the total ground truth items, as we no longer care how efficient our recommendations are but instead focus on recommending as many ground truth items as possible.

## 2.36 Question 36

The plots are as following, we can find out that average precision will go down when t increases, and average recall instead will go up when t increases, and the average precision and average recall has a negative correlation:

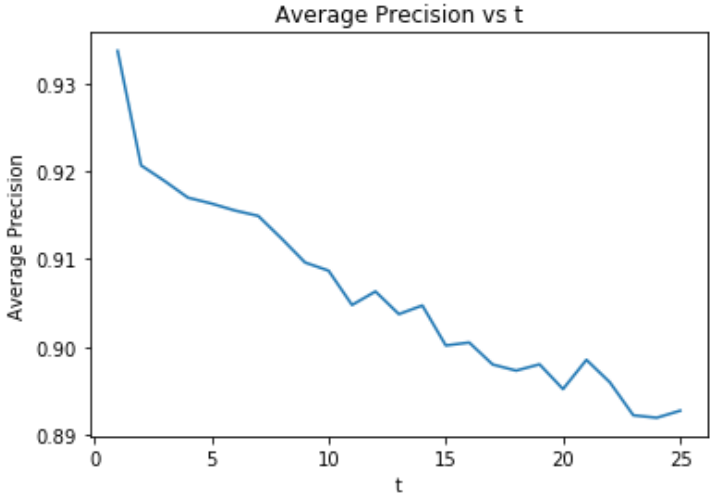


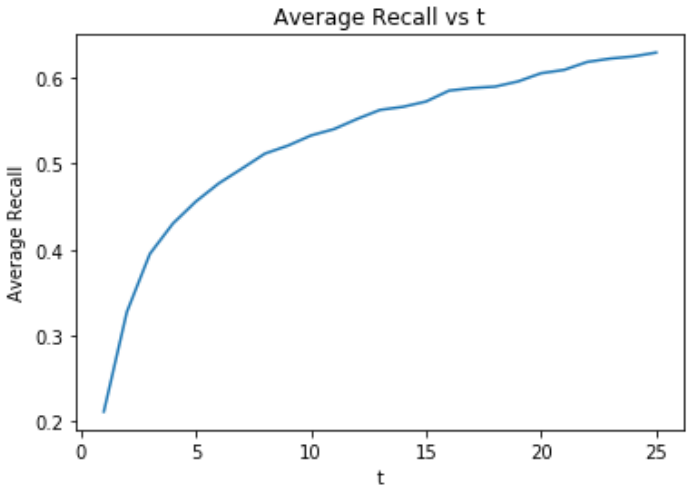



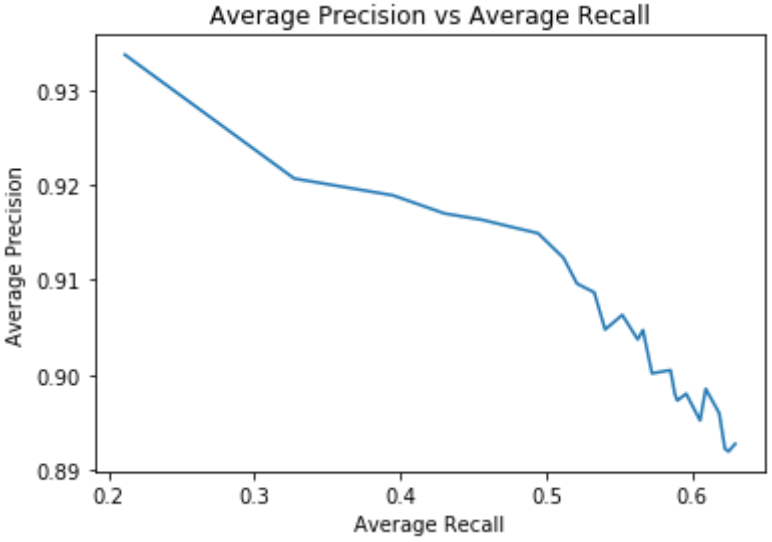



## 2.37 Question 37

The plots are as following, the basic logic of each graph is the same as above:

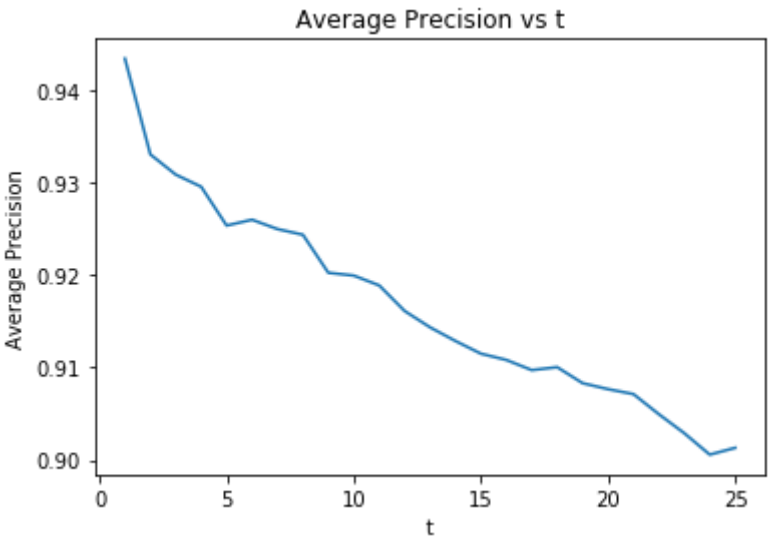


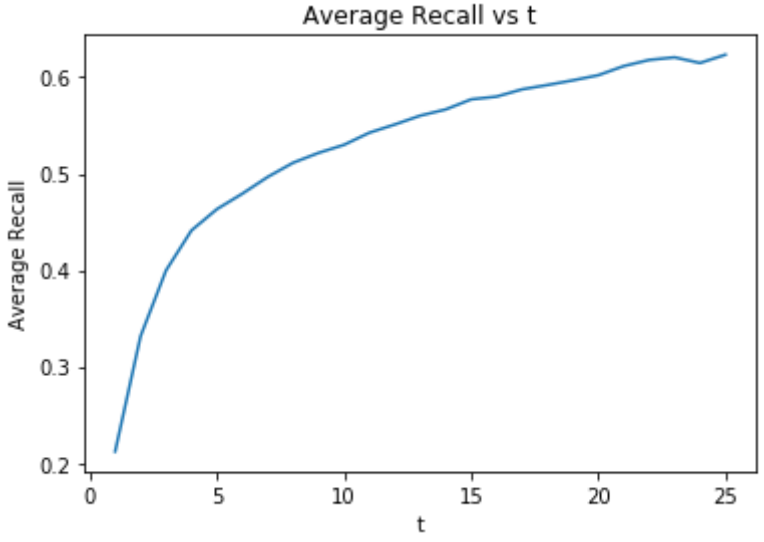


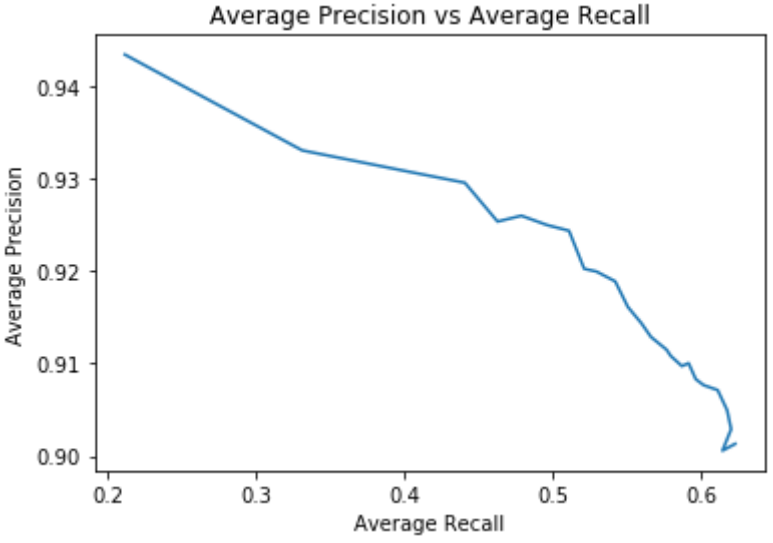




## 2.38 Question 38

The plots are as following, the basic logic of each graph is the same as above:


Average Precision vs t
Average Precision
0.94
0.93
0.92
0.91
0.90
0
5
10
15
20
25
t


Average Recall vs t
Average Recall
0.6
0.5
0.4
0.3
0.2
0
5
10
15
20
25
t


Average Precision vs Average Recall
Average Precision
0.94
0.93
0.92
0.91
0.90
0.2
0.3
0.4
0.5
0.6
Average Recall

## 2.39 Question 39

The ROC curve is as following, we also can see from precision-recall curves above that all three filters perform similarly, k-NN filter and NNMF filter performed approximately the same on precision, while NNMF performed better on recall: