Question 1A:

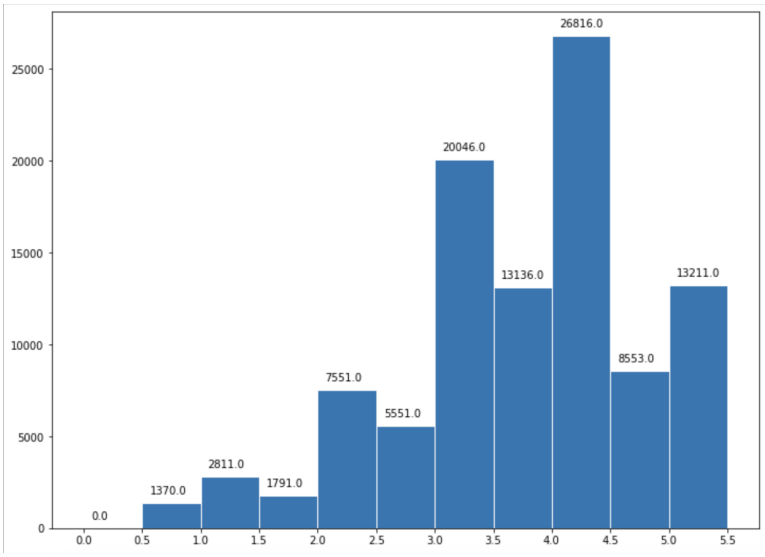
The total number of available ratings is 100836

The total number of possible ratings is $9742 \times 610 = 5942620$

Then we have $Sparsity = \frac{100836}{5942620} \approx 0.016968$

Question 1B:

(the frequency of the rating values)

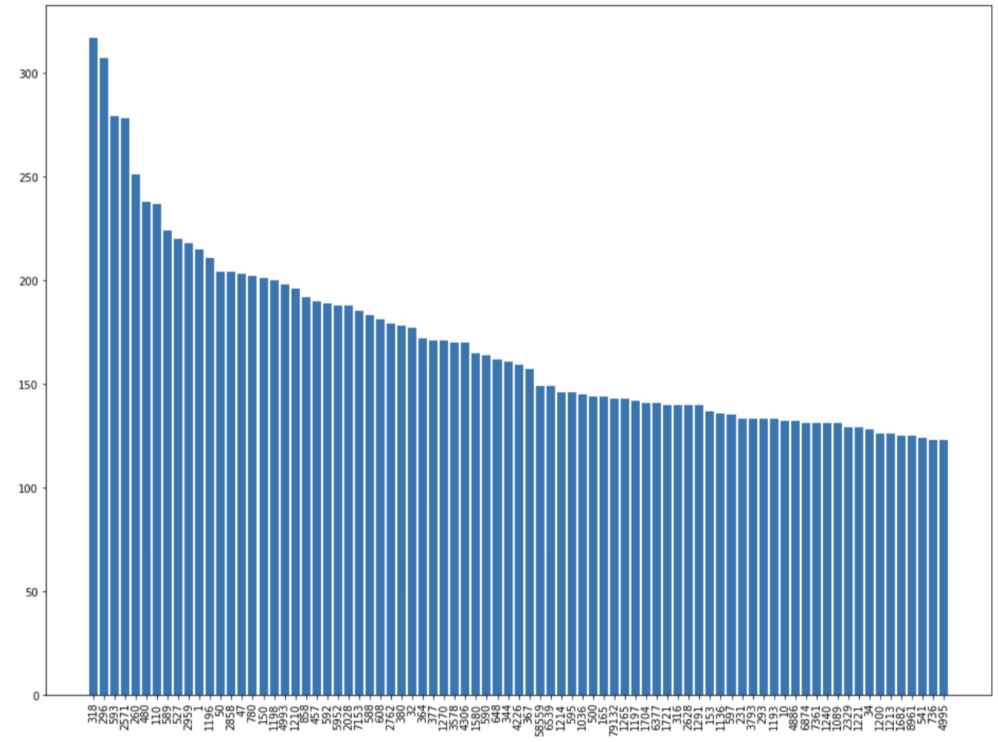


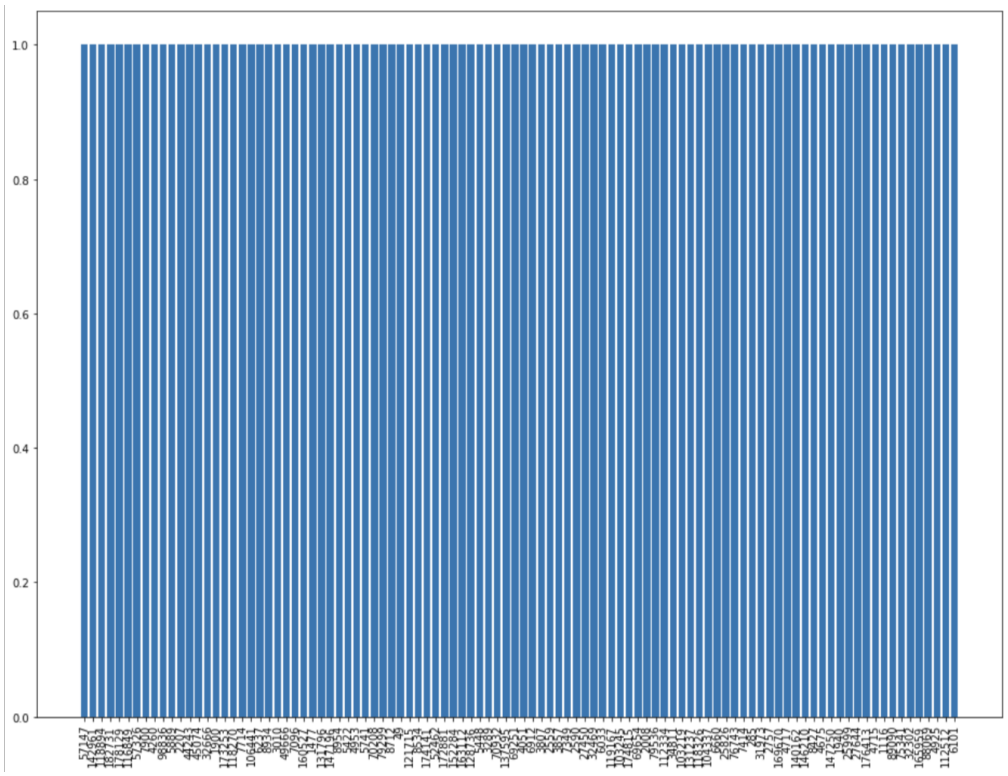
Comments on the shape of the histogram:

1. Most ratings are greater than or equal to 3
2. The most frequent rating in the dataset is 4
3. The least frequent rating in the dataset is 0.5

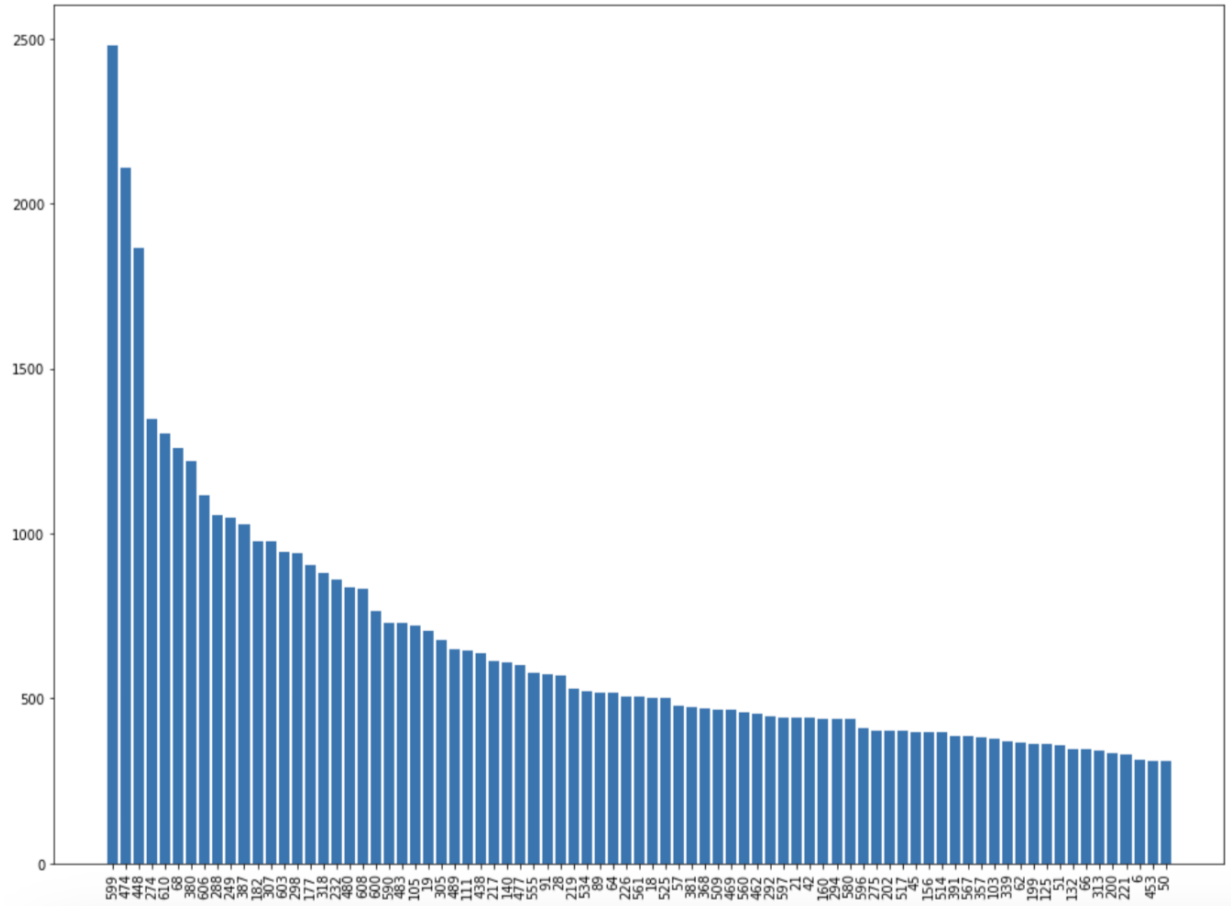
Question 1C:

The number of ratings received among movies is in a monotonically decreasing trend. In the plots above, the first one shows the first 80 movies (index and the # of received ratings) that with the highest number of ratings received, and the second one shows the 80 movies (index and the # of received ratings) that with the lowest number of ratings received (only have one rating).


300
250
200
150
100
50
0


1.0
0.8
0.6
0.4
0.2
0.0

Question 1D:

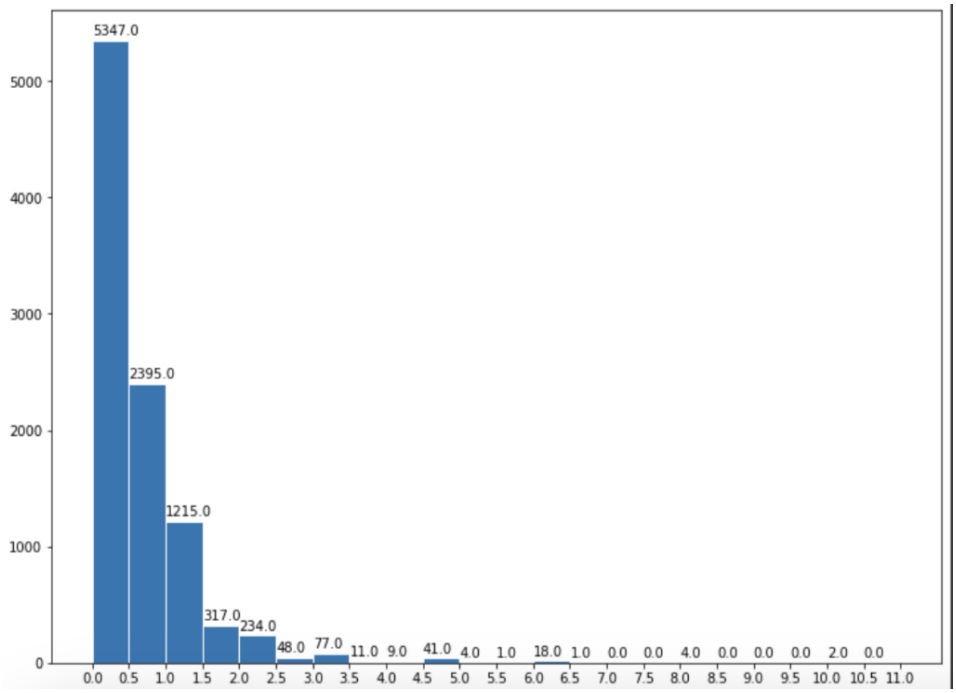


The above plot shows the first 80 users and the number of ratings given by that user, overall the plot is in a monotonically decreasing trend.

Question 1E:

From the plot of question C, we can tell that most of the movies have received a limited amount of ratings. (The largest number of ratings received by a movie is 329 and the smallest number of ratings is 1.) This means most movies have very few ratings, but a small number of movies received the majority of the ratings (Same for the distribution in question D). Therefore, a lot of cell (user-movie) in matrix R is not available (no ratings), which means R has a low sparsity(i.e the ratings matrix is very sparse), and this low sparsity can lead to overfitting problems if the matrix is directly used without preprocessing or regulations.

Question 1F:



From the histogram here, we can find that the majority of the movies have a rating variance less than 2.5. This means the ratings for each movie in the dataset are relatively consistent and trustworthy.

Question 2A:

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

Question 2B:

The intersection of I_u and I_v simply means the set of item indices which both user u and v have given ratings. The intersection can be empty as it is such a big data set, and this can happen when two people never give a rating to the same item. However, we don't need to care about such pairing because an empty set provides no information to infer their similarities.

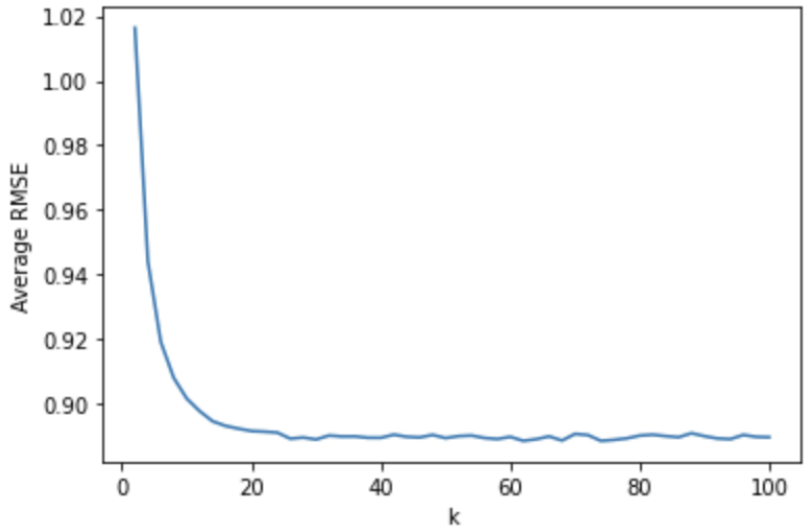
Question 3:

The prediction function is designed to estimate the rating user u would give to item j based on the rating user v gave. The mean-centering of the raw ratings $r_{vj} - \mu_v$ can represent the preference of user v. If users either give very high or low ratings on items, we can not distinguish the degree of preference only based on their average ratings. Subtracting the mean rating of

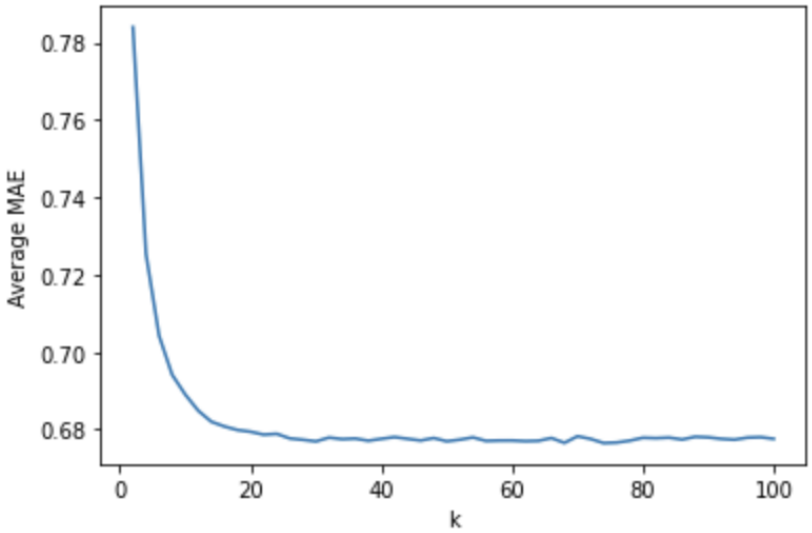
each user from their ratings can reduce the bias, and so that we can obtain a more accurate representation of how the user rates this item relative to their own average.

Question 4:

`average RMSE (Y-axis) against k (X-axis)`



`average MAE (against) k (X-axis)`

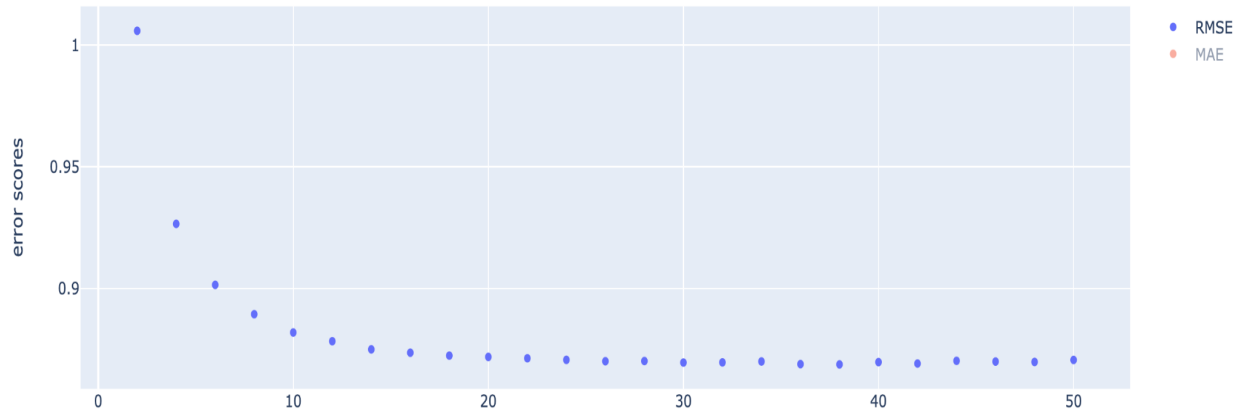


Question 5:

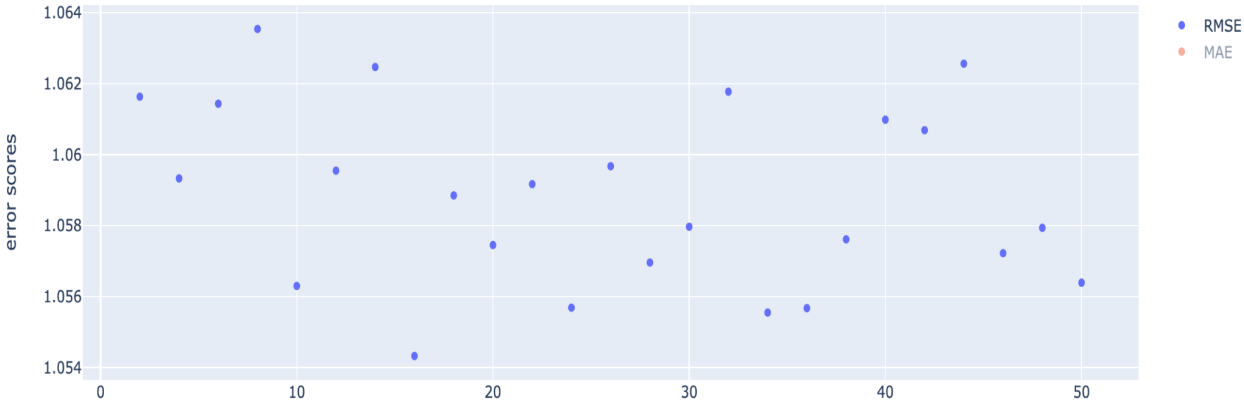
From the plots above, we can find the minimum k is about 20, and
The steady status value of average RMSE is ≈ 0.8915
The steady status value of average MAE is ≈ 0.6792

Question 6:

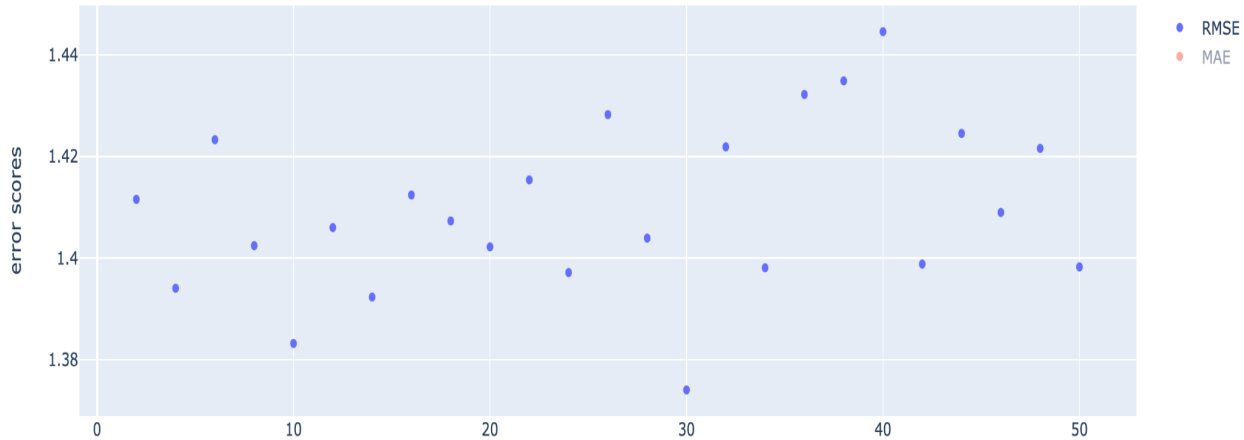
RMSE: For the popular dataset: the minimum average RMSE is about 0.86.



For the unpopular dataset: the minimum average RMSE is about 1.05.

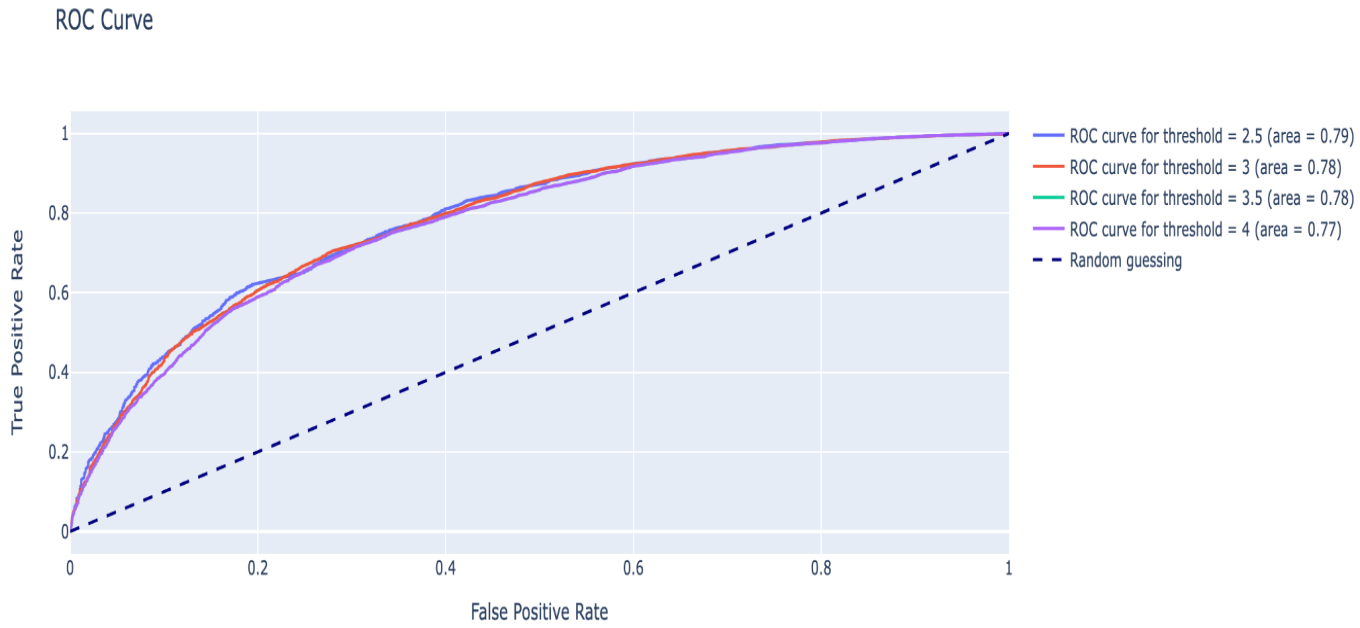


For the high variance dataset: the minimum average RMSE is about 1.37.



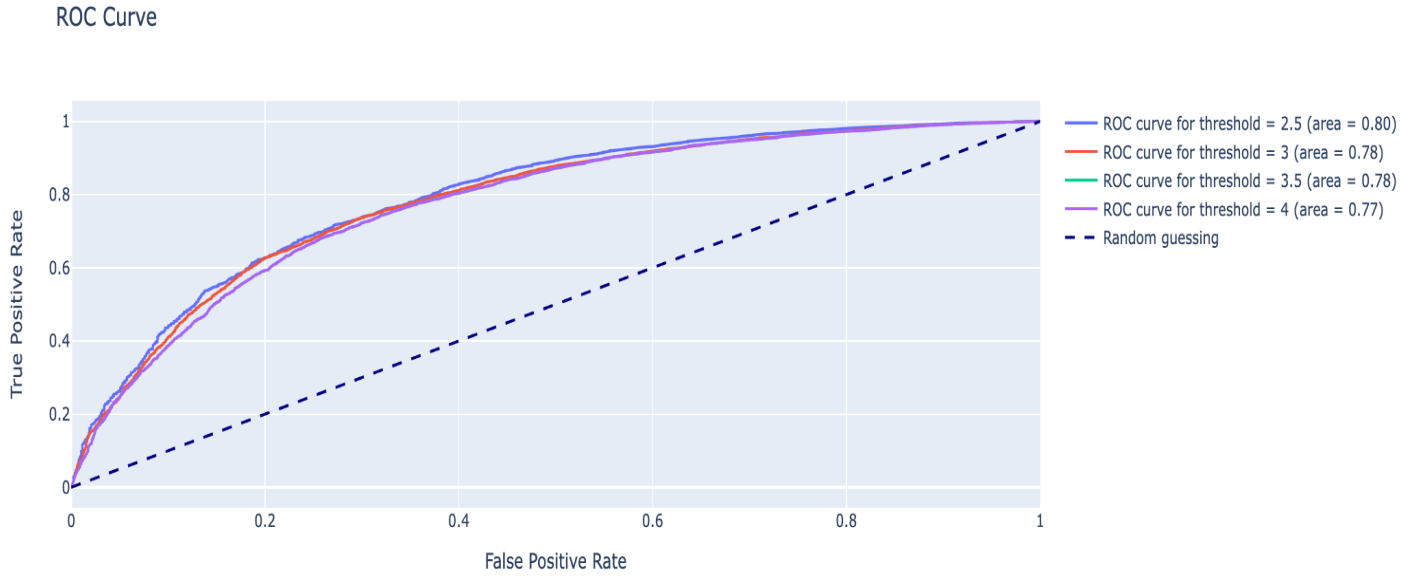
ROC: For the original untrimmed dataset:

When the threshold is 2.5, we have an AUC score of 0.79.
When the threshold is 3, we have an AUC score of 0.78.
When the threshold is 3.5, we have an AUC score of 0.78.
When the threshold is 4, we have an AUC score of 0.77.



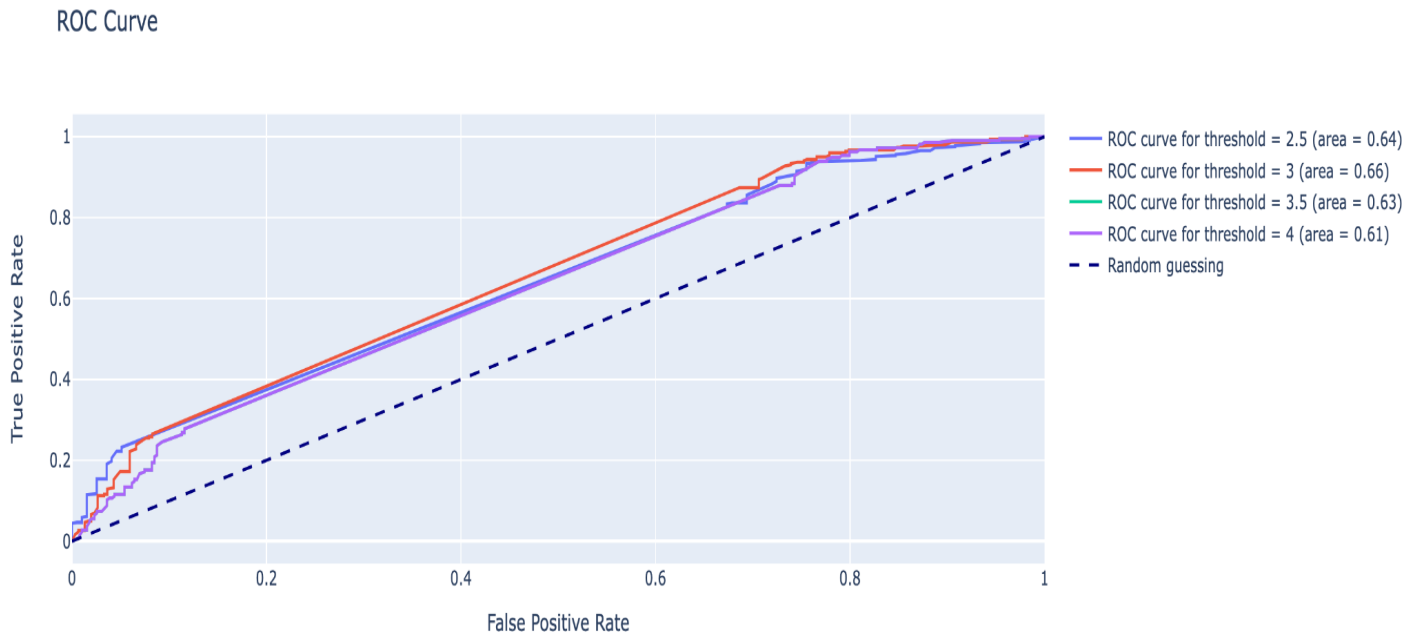
For the popular dataset:

When the threshold is 2.5, we have an AUC score of 0.8.
When the threshold is 3, we have an AUC score of 0.78.
When the threshold is 3.5, we have an AUC score of 0.78.
When the threshold is 4, we have an AUC score of 0.77.



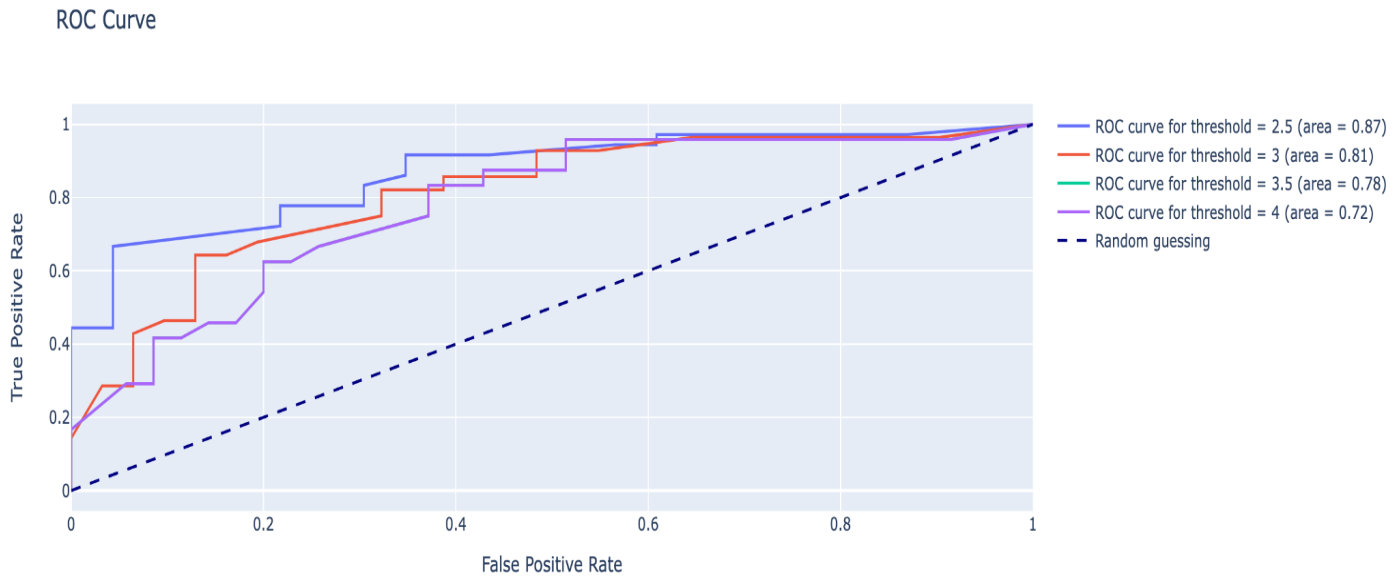
For the unpopular dataset:

When the threshold is 2.5, we have an AUC score of 0.64.
When the threshold is 3, we have an AUC score of 0.66.
When the threshold is 3.5, we have an AUC score of 0.63.
When the threshold is 4, we have an AUC score of 0.61.



For the high variance dataset:

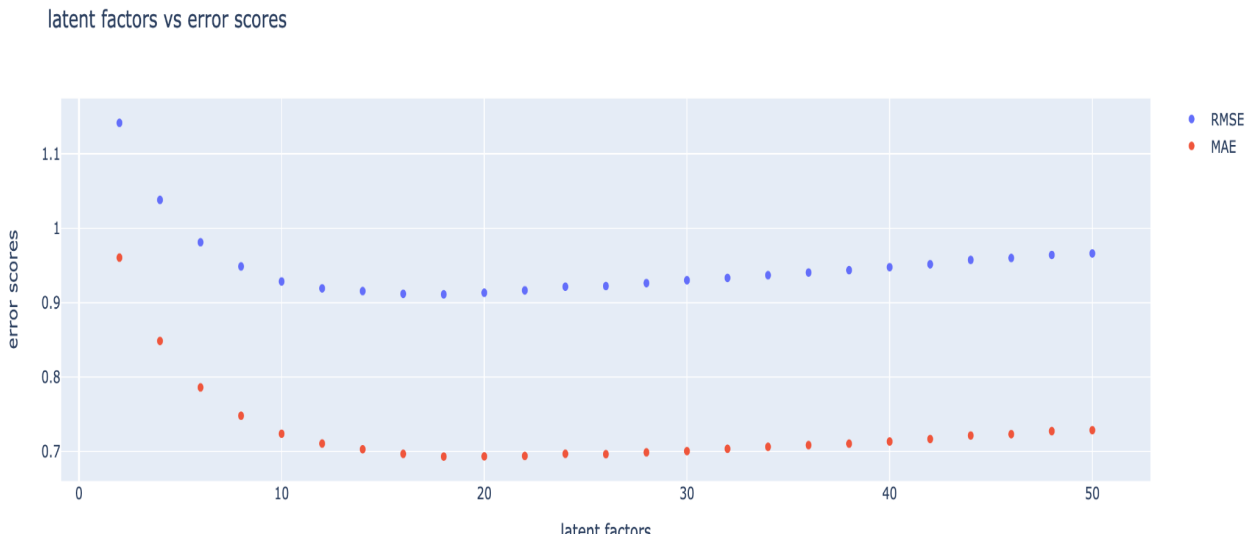
When the threshold is 2.5, we have an AUC score of 0.87.
When the threshold is 3, we have an AUC score of 0.81.
When the threshold is 3.5, we have an AUC score of 0.78.
When the threshold is 4, we have an AUC score of 0.72.



Question 7:
The optimization problem given by the equation is not convex. For a fixed U, the least-squares formulation is:

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

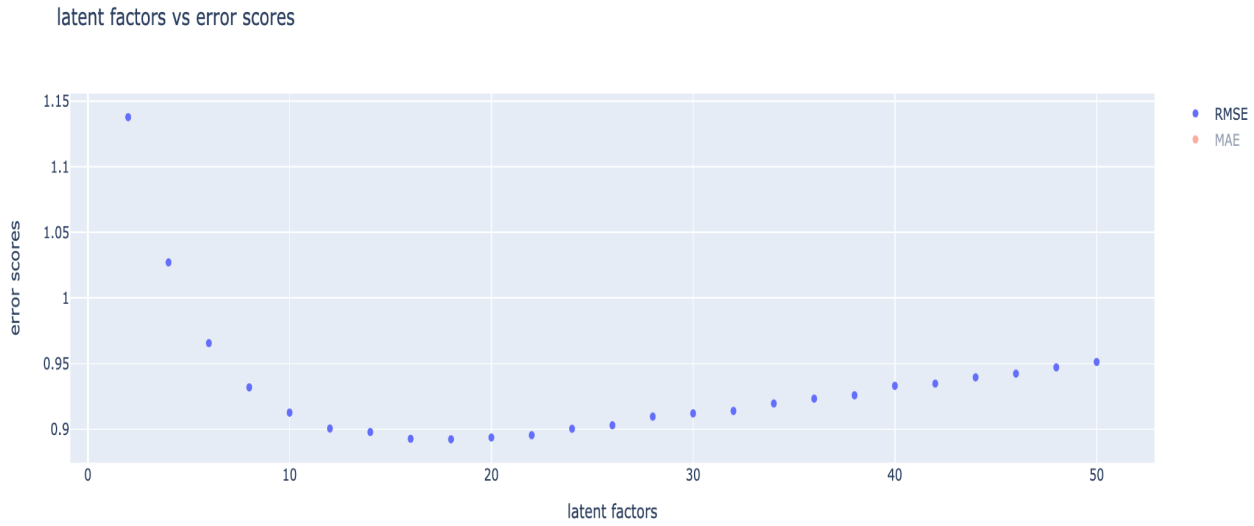
Question 8:
A:



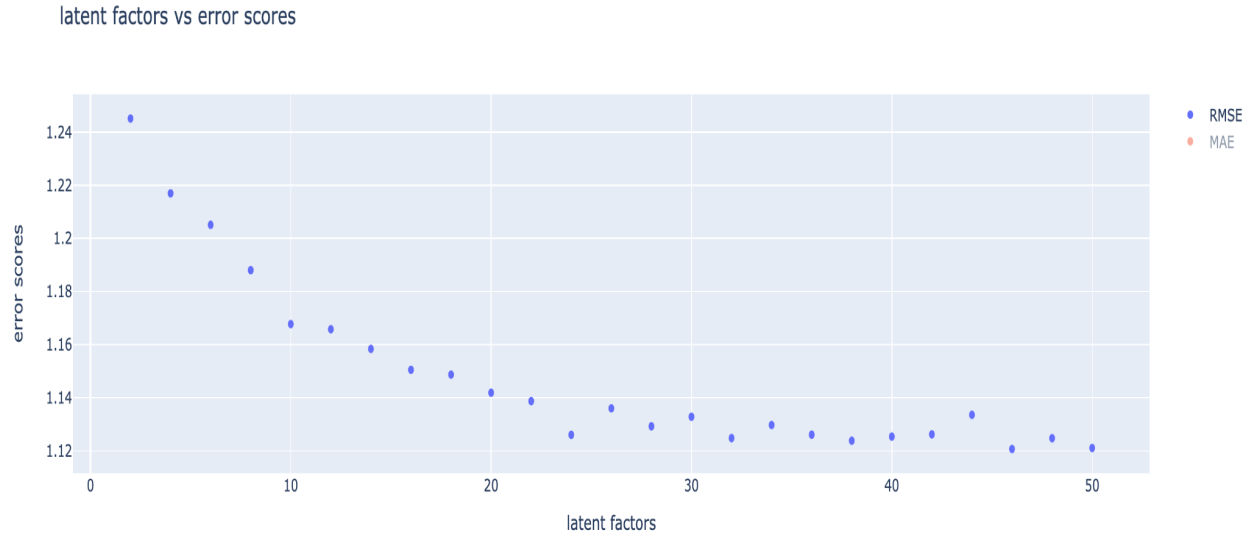
B:
The minimum RMSE is 0.91. The minimum MAE is 0.69. The optimal number of latent factors is 18 which is about the same as the number of movie genres.

C:

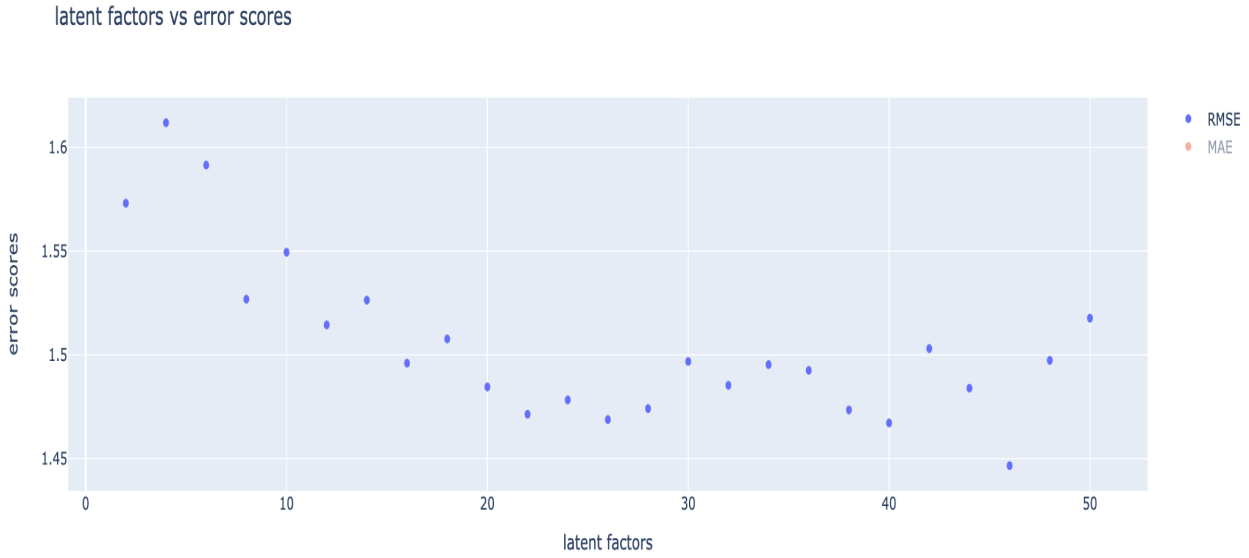
For the popular dataset: the minimum average RMSE is about 0.89.



For the unpopular dataset: the minimum average RMSE is about 1.12.

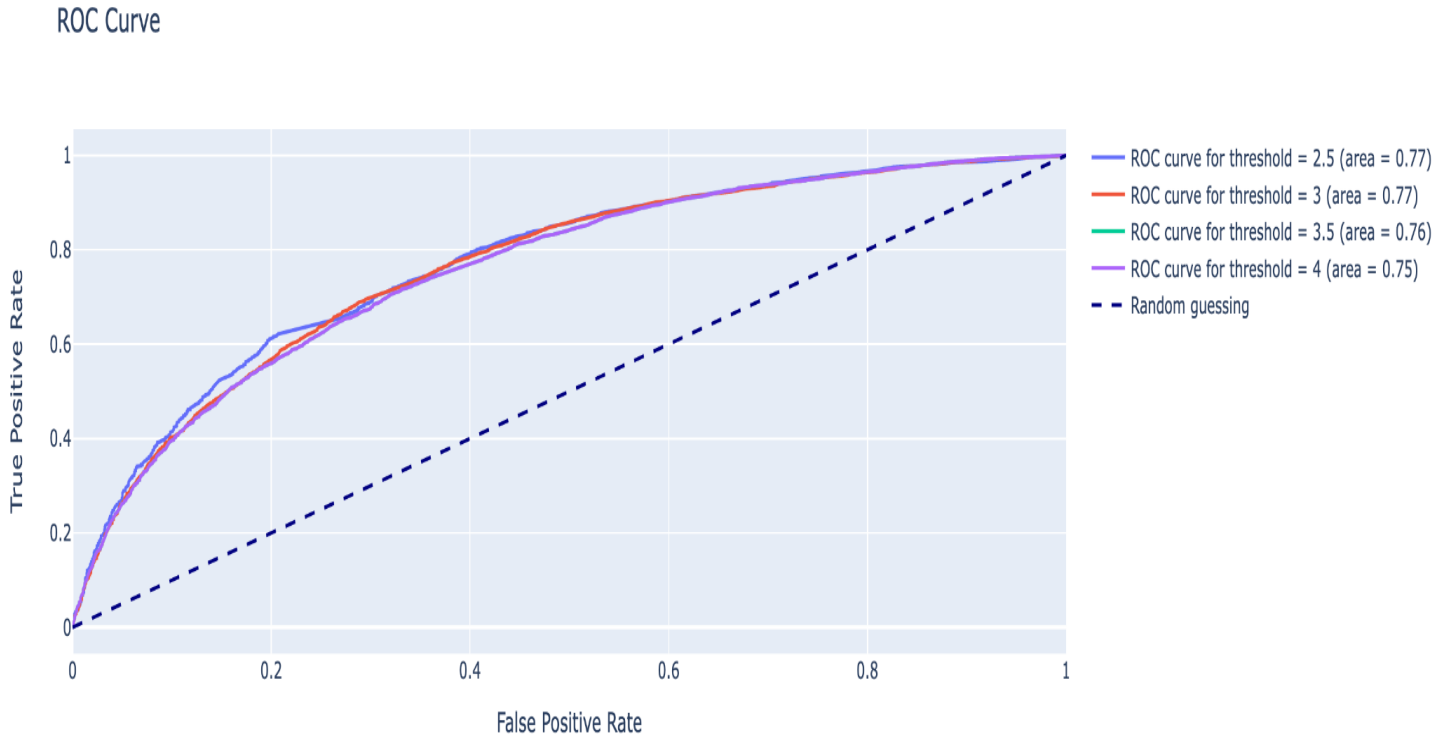


For the high variance dataset: the minimum average RMSE is about 1.44.



ROC:

When the threshold is 2.5, we have an AUC score of 0.77.
When the threshold is 3, we have an AUC score of 0.77.
When the threshold is 3.5, we have an AUC score of 0.76.
When the threshold is 4, we have an AUC score of 0.75.



Question 9:
Column1: {'Comedy': 8, 'Action': 5, 'Romance': 3, 'Drama': 2, 'Children': 1, 'Crime': 1, 'Adventure': 1, 'Sci-Fi': 1, 'Thriller': 1, 'Documentary': 1}

Column2: {'Drama': 7, 'Comedy': 4, 'Children': 3, 'Sci-Fi': 3, 'Animation': 2, 'Romance': 2, 'Musical': 1, 'Thriller': 1, 'Action': 1, 'Adventure': 1, 'Crime': 1, 'Fantasy': 1}

Column3: {'Comedy': 7, 'Drama': 4, 'Action': 2, 'Adventure': 1, 'Children': 1, 'Sci-Fi': 1}

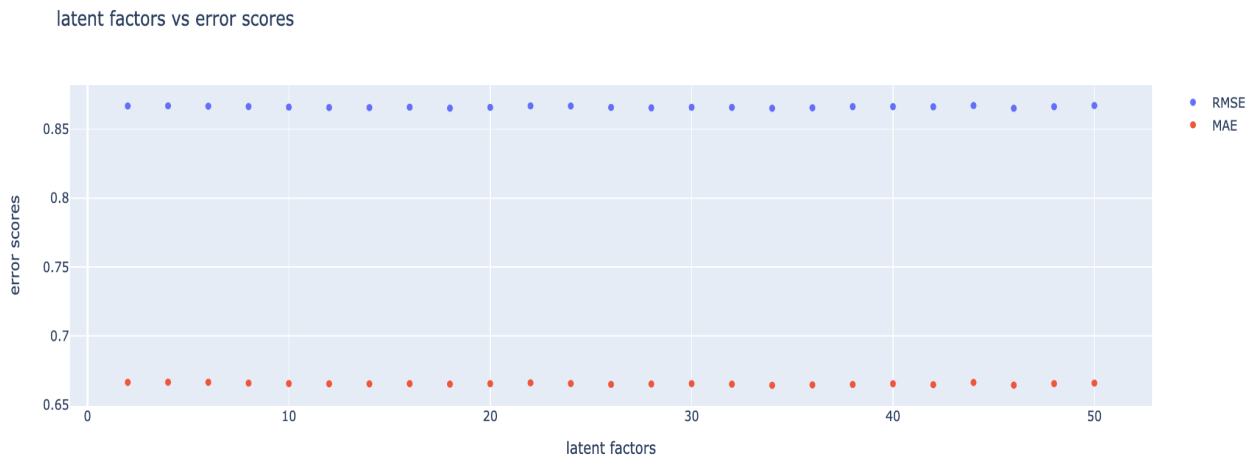
Column4: {'Comedy': 6, 'Drama': 4, 'Action': 2, 'Sci-Fi': 2, 'Thriller': 1, 'Horror': 1}

Column5: {'Comedy': 6, 'Drama': 3, 'Action': 3, 'Crime': 3, 'Romance': 2, 'Sci-Fi': 1, 'Thriller': 1, 'Fantasy': 1, 'Horror': 1}

(We count the genres for top 10 movies of all 20 columns, you can find the complete result in our source code)

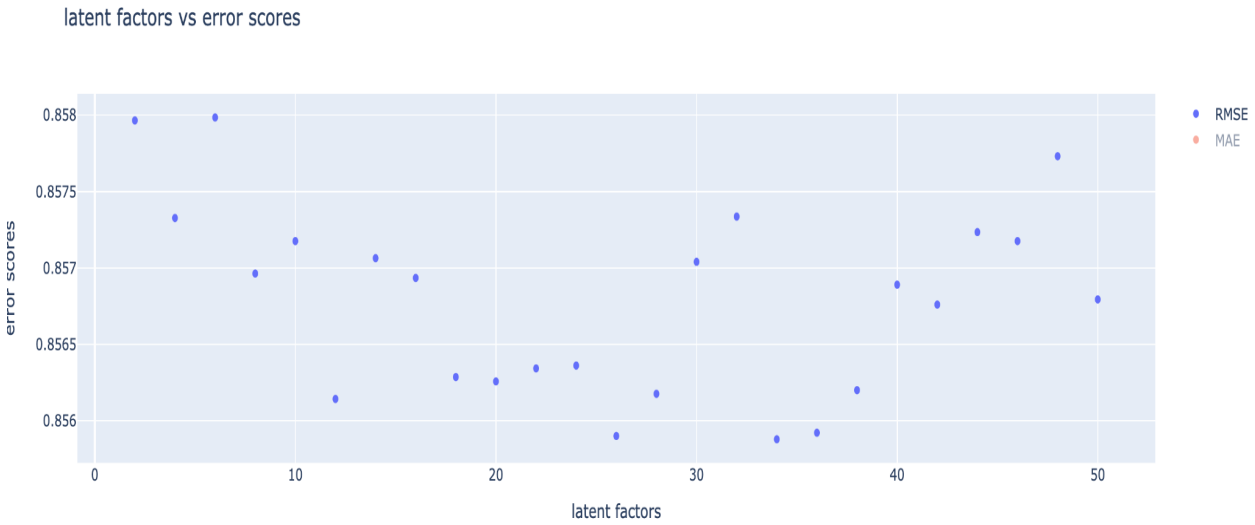
We find the corresponding movieId based on the sorted V matrix. After that, we collect the genres of all ten movies and get the frequency of each genre. In the dictionary output above, we can clearly see that most of the top 10 movies belong to a particular or a small collection of genres. The top ten movies of each column represents a particular combination of genres. For example, one of the columns represents the movie genre combination of "comedy + action" because there are 8 movies that have 'comedy' and 5 movies that have 'action' as keywords in genre. Because most columns have their own particular genre combination, there is a strong connection between the latent factors and the movie genres.

Question 10:
A:

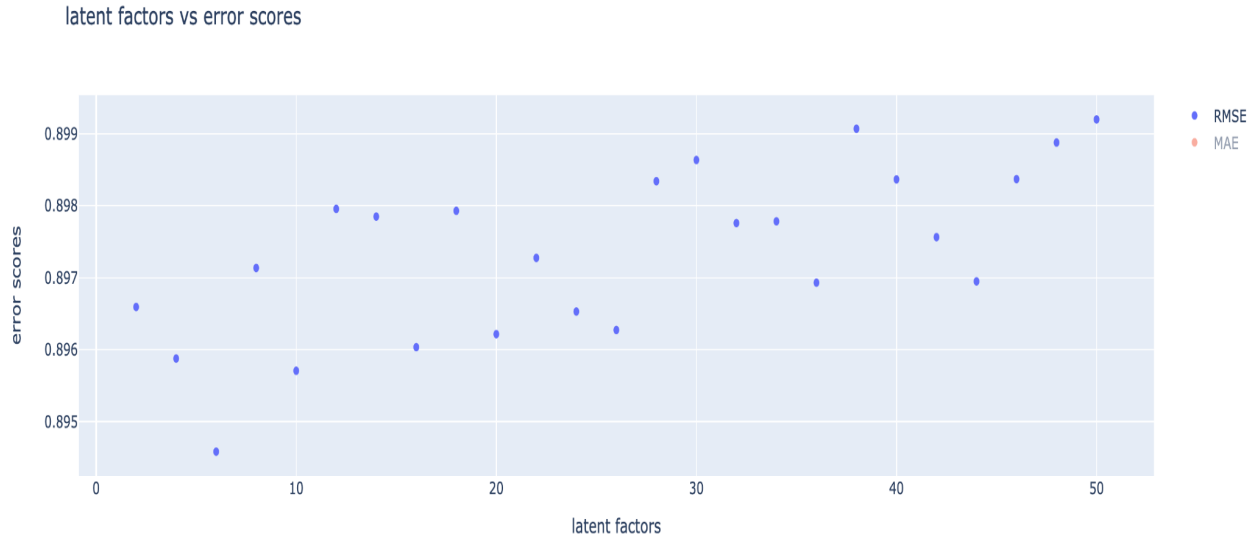


B:
The minimum RMSE for the original dataset is 0.86. The minimum MAE is about 0.66. The optimal number of latent factors is 16 which is about the same as the number of genres 18.

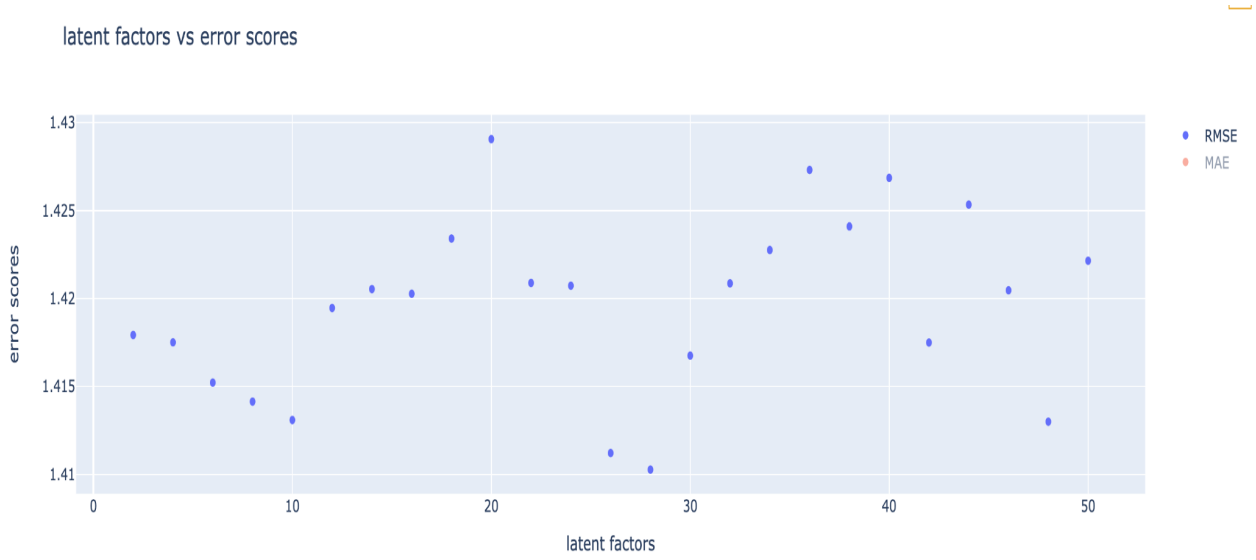
C:
For the popular dataset: the minimum average RMSE is about 0.85.



For the unpopular dataset: the minimum average RMSE is about 0.89.

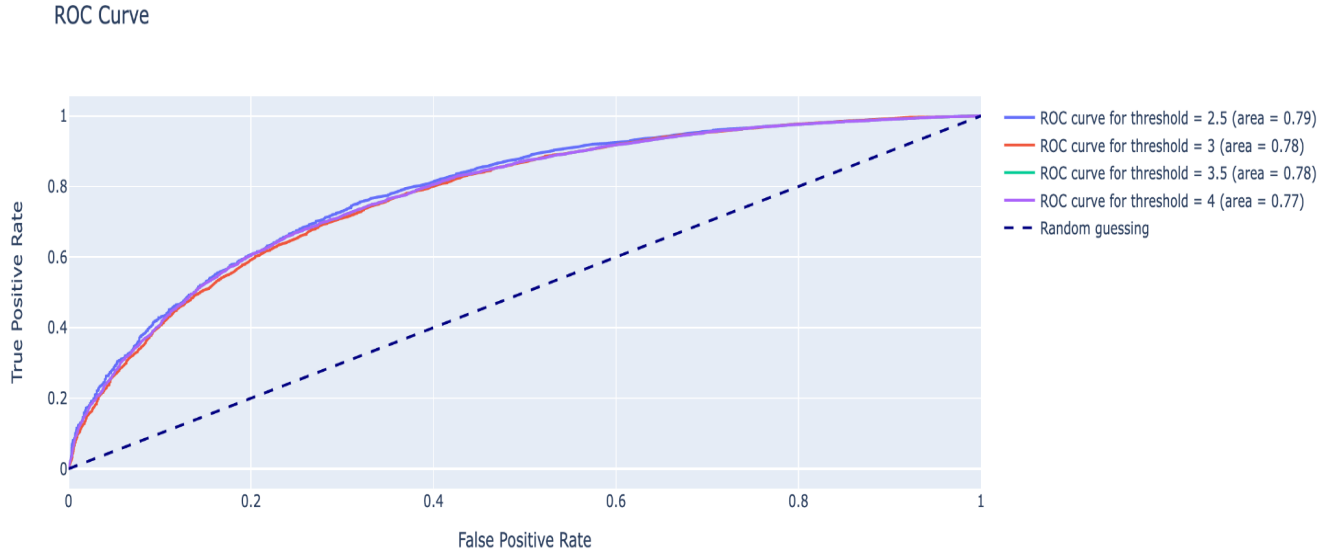


For the high variance dataset: the minimum average RMSE is about 1.41.



ROC:

When the threshold is 2.5, we have an AUC score of 0.79.
When the threshold is 3, we have an AUC score of 0.78.
When the threshold is 3.5, we have an AUC score of 0.78.
When the threshold is 4, we have an AUC score of 0.77.



Question 11:

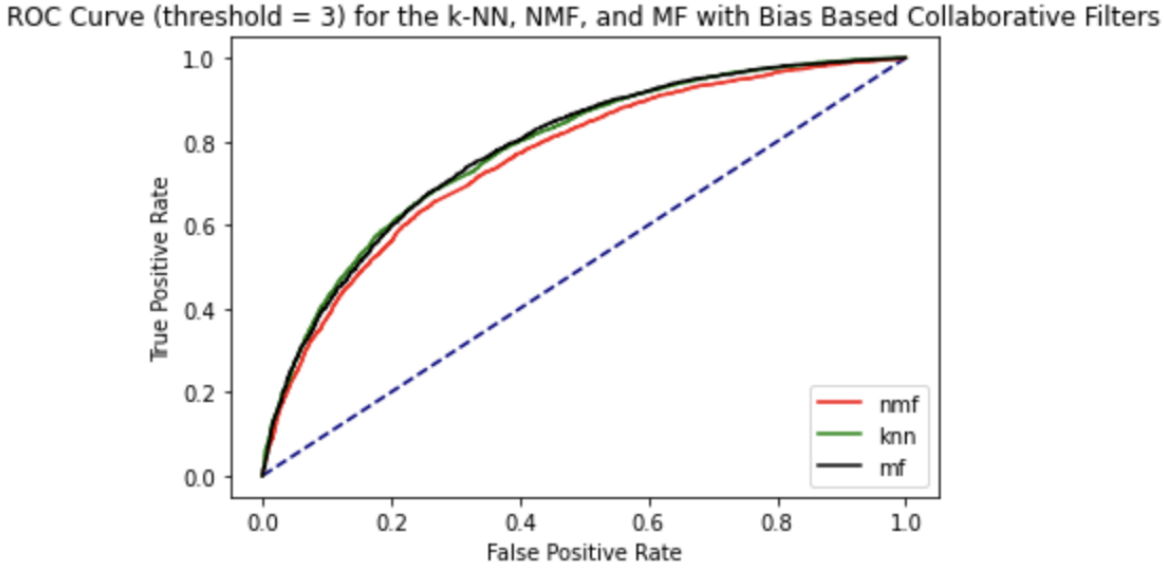
The average RMSE for NCF is: `0.9347079916774117`

The average RMSE for NCF with popular movie trimming is: `0.9323018781445956`

The average RMSE for NCF with unpopular movie trimming is: `0.9709674271855377`

The average RMSE for NCF with high-variance trimming is: `1.3741094082762655`

Question 12:



When the threshold is 3, from the plot, we can see that based on the ROC curves and the area under the curve, the NMF-based collaborative filter has the worst performance among three models and the MF-based collaborative filter has the best performance.

Question 13:

Precision: $\frac{|S(t) \cap G|}{|S(t)|}$.

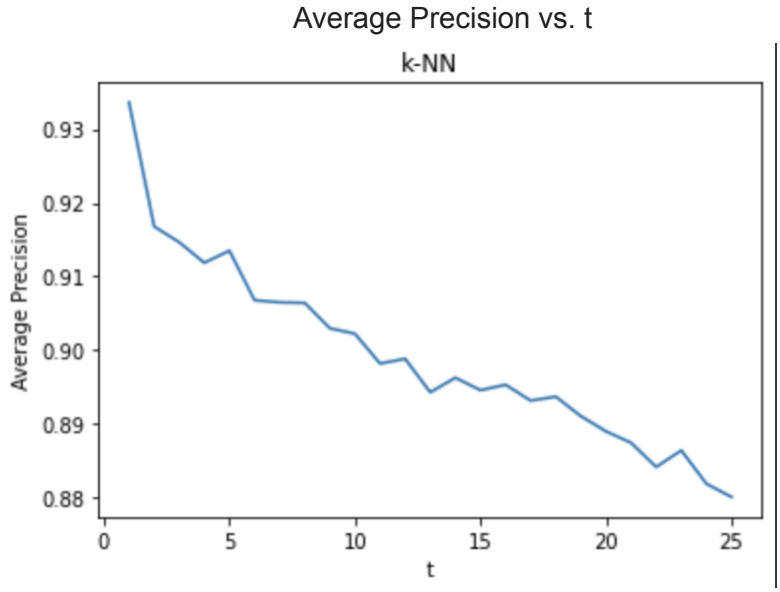
With the given notations about $S(t)$ and G, we have $|S(t) \cap G|$ represents the number of items in the recommended items set that the user likes, and $|S(G)|$ represents the total number of items in the recommended items set. Thus, the precision means how likely the movie in the recommended moving set would be liked by the user. (the fraction of items that the user liked in the recommended items set out of the number of items in the recommended items set).

Recall: $\frac{|S(t) \cap G|}{|G|}$.

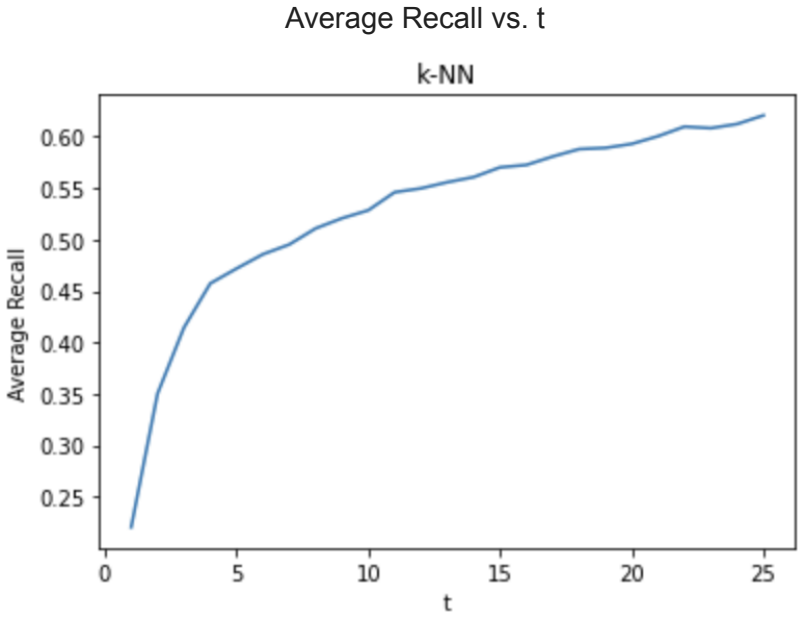
$|G|$ represents the number of all movies that the user likes. Thus, the recall means how likely a movie the user likes would be in the recommended items set. (the fraction of items that the user liked in the recommended set out of the number of all items that the user likes).

Question 14:

k-NN (k=20) collaborative filtering technique with 10-fold cross validation

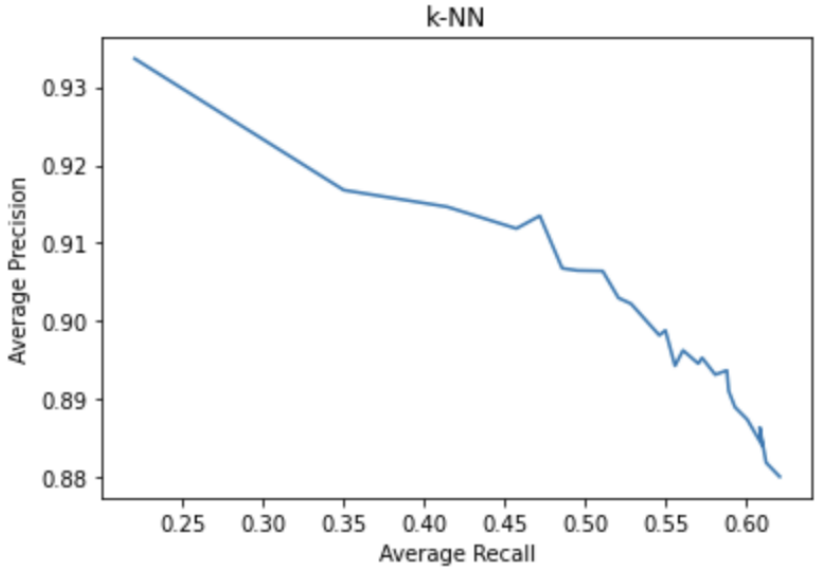


From the plot above, we can see that as t increases, the average precision decreases (in a general way, sometimes, the average precision slightly increases but then decreases again).



From the plot above, we can see that as t increases, the average recall increases.

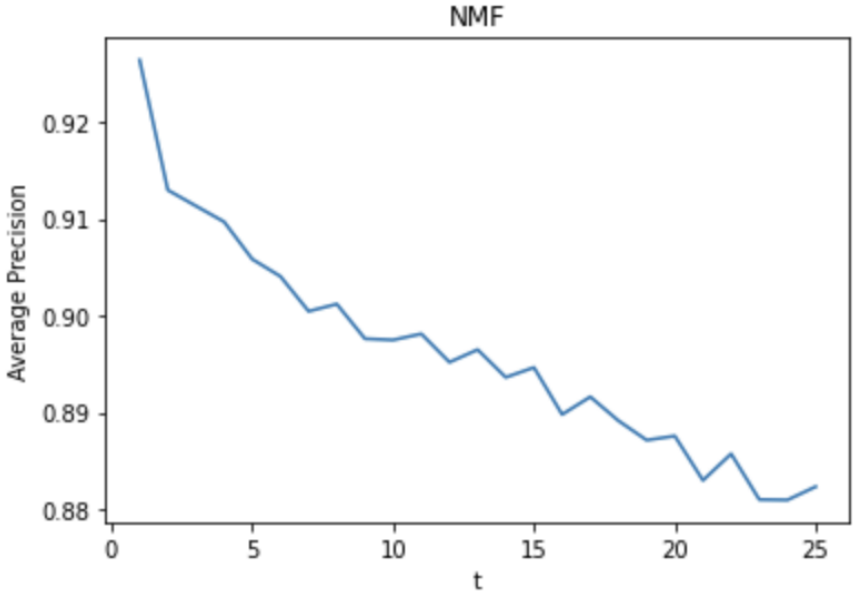
Average Precision vs. Average Recall



From the plot above, we can see that as average recall increases, the average precision decreases in a general way. This shows that there is a tradeoff between average recall and average precision.

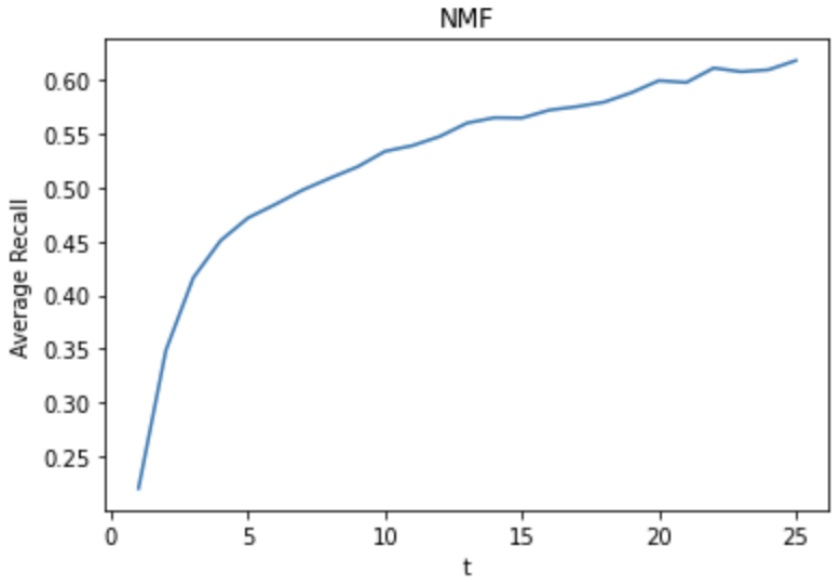
NMF (n_factors = 16)

Average Precision vs. t



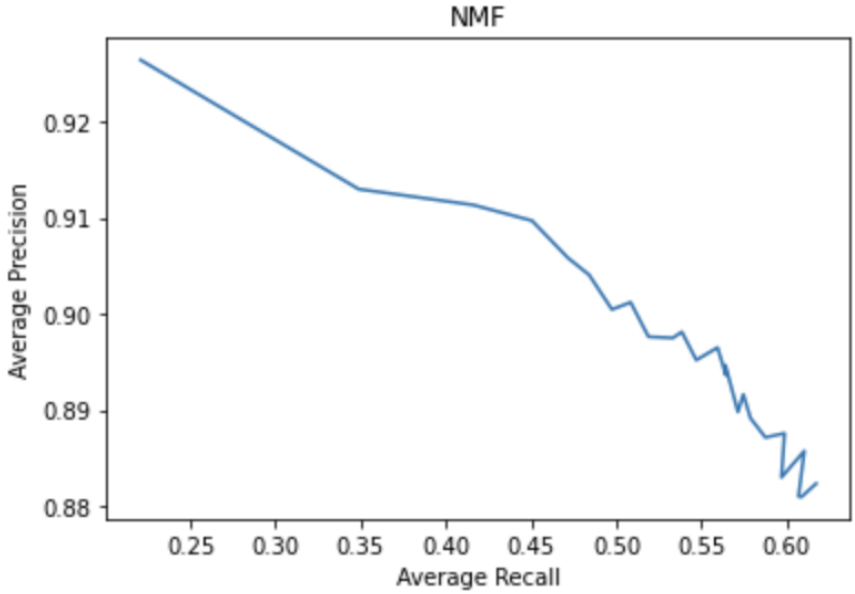
From the plot here, we can also see that as t increases, the average precision decreases in a general way.

Average Recall vs. t



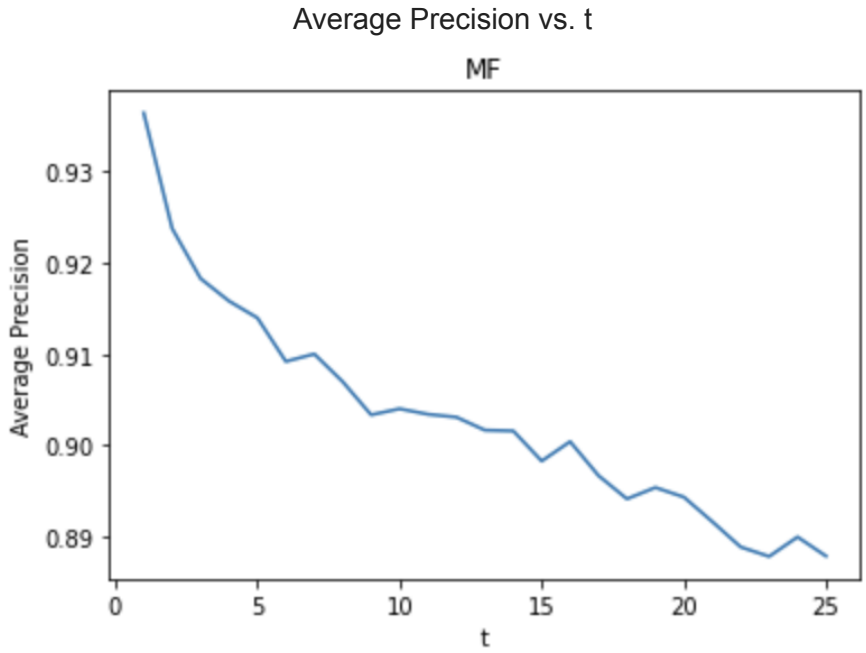
From the plot here, we can see that as t increases, the average recall also increases.

Average Precision vs. Average Recall

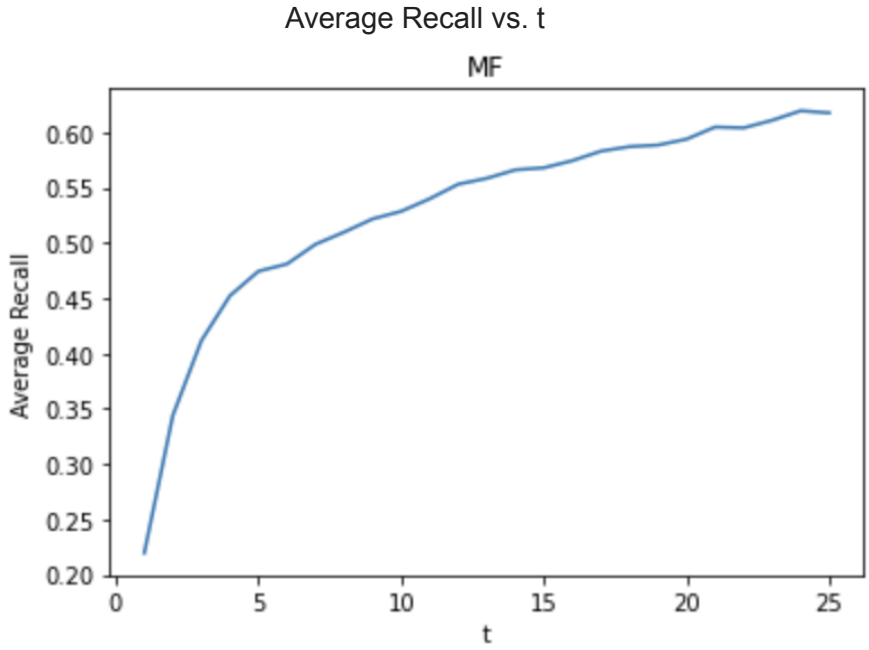


From the plot here, we can see that as average recall increases, the average precision decreases. This shows a tradeoff between average recall and average precision.

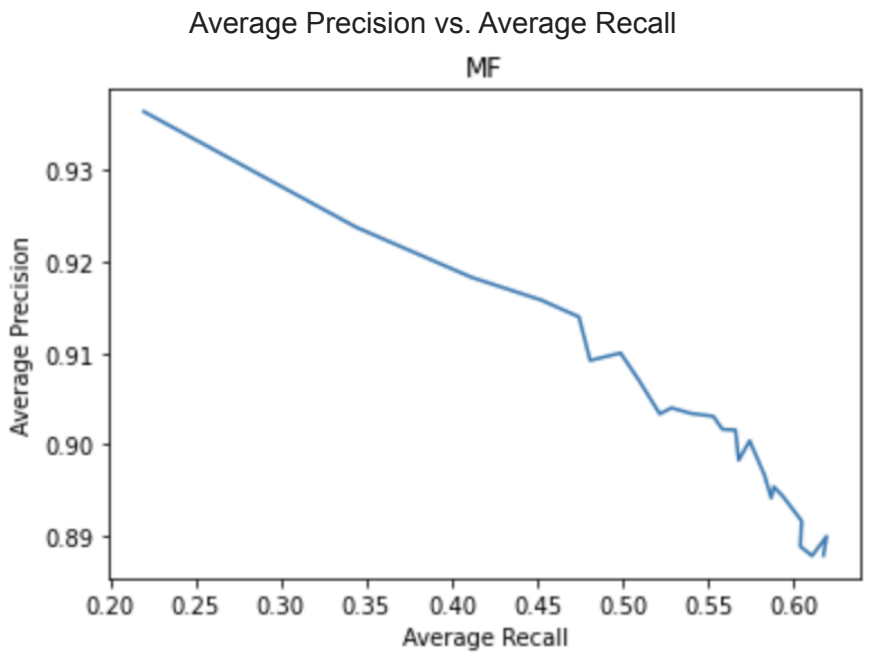
MF (with n_factors = 16)



From the plot here, we can see that as t increases, the average precision decreases in a general way.

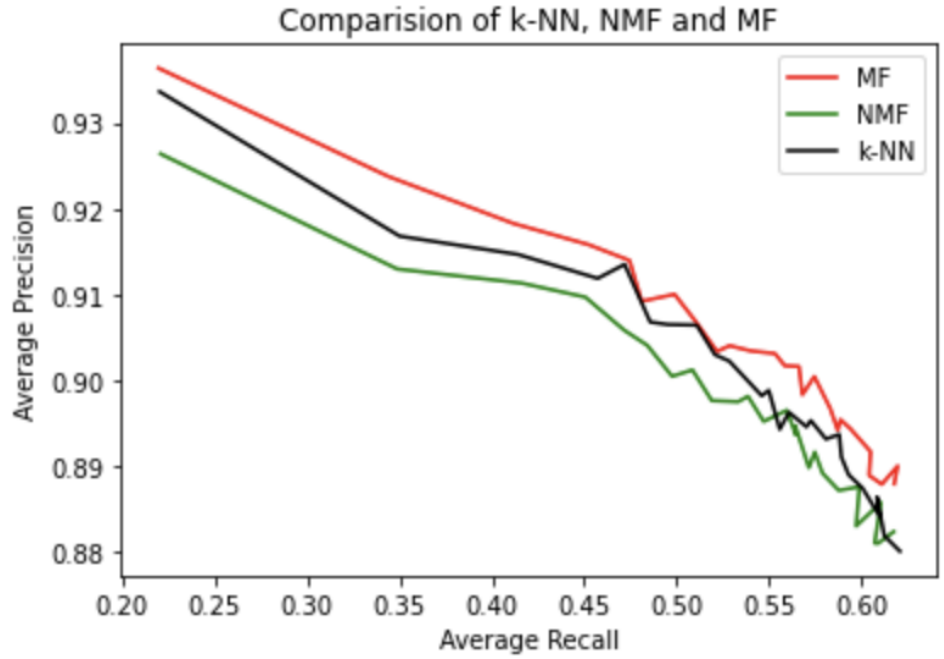


From the plot here, we can see that as t increases, the average recall also increases.



From the plot here, we can see that as average recall increases, the average precision decreases. This shows a tradeoff between average recall and average precision.

Plot the best precision-recall curves obtained for the three models in the same figure. Use this figure to compare the relevance of the recommendation list generated using k-NN, NMF, and MF with bais predictions.



From the above plot, we can find that as average recall increases, the average precision generally decreases for the three models. And MF has the best performance since the red line

is always higher than the other two (the average precision for MF is generally higher than the average precision for the other two models when the average recall is the same).