EE 219 Project 3 Report: Collaborative Filtering

Jessica Fu (805034901) & Jasmine Moreno (705035581)
Winter 2018

---

1 Introduction

The rapidly increasing frequency of online electronic and business transactions pushes the development and improvement of recommender systems technology. Its goal is to recommend items to a user or customer. These systems can use user-item interactions (such as ratings) or information about the user (such as profiles or keywords). Collaborative filtering uses user-item interactions.

2 Collaborative filtering models

Collaborative filtering models are used to make recommend an item to users. The tricky part is knowing how to handle the data since most user only viewed a small fraction of the movies. There are multiple ways to implement a collaborating filters like calculating the similarity between users to recommend the movies.

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 filtering

3 MovieLens dataset

In this part, we analyze and visualize some properties of the MovieLens dataset . The dataset contains four Comma Separated Values (.csv) files. We only used the ratings.csv and movies.csv files. Using these two files, we created a rating matrix R (size 671x9125) since there are 671 users and 9125 movies.

Question 1

Q: Compute the sparsity of the movie rating dataset, where sparsity is defined by equation below (it has been modified to the TA’s response on Piazza):

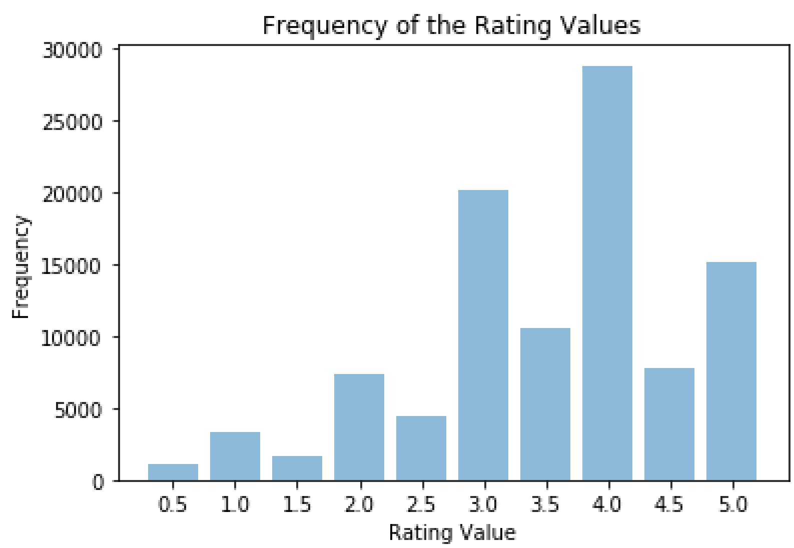
$$Sparsity = 1 - \frac{Total\ number\ of\ available\ ratings}{Total\ number\ of\ possible\ ratings}$$

A: From the ml-latest-small dataset's README document, we know there is 9125 movies. There are a total of 100004 ratings from 671 users. That means the total number of available ratings are 100004 and the total number of possible ratings are 671 users x 9125 movies = 6122875. Using the equation above, we calculated the sparsity to be 0.98367.

Question 2

Q: Plot a histogram showing the frequency of the rating values. To be specific, bin the rating values into intervals of width 0.5 and use the binned rating values as the horizontal axis. Count the number of entries in the ratings matrix R with rating values in the binned intervals and use this count as the vertical axis. Briefly comment on the shape of the histogram.

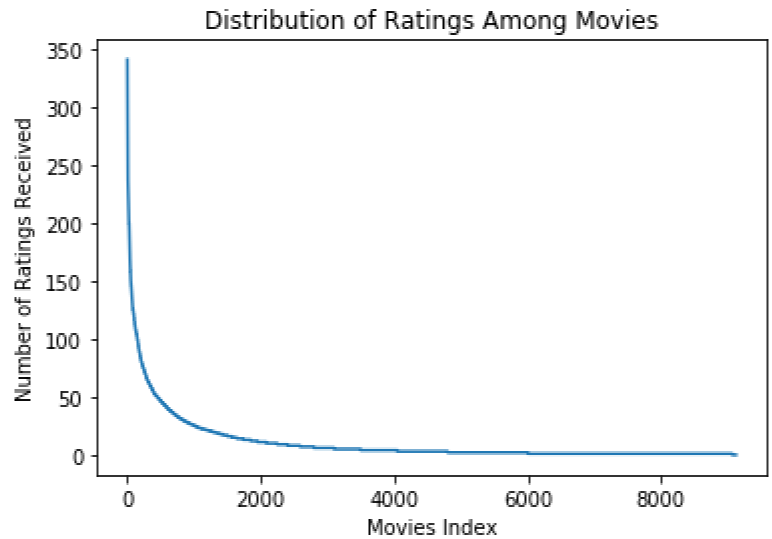
A: Below we have the histogram of the movie ratings. As we can see the frequency is higher when the rating value is greater than 3. This probably shows that people tend to rate more if they liked a movie or tend to watch movies they would like.



Question 3

Q: Plot the distribution of ratings among movies. To be specific, the X-axis should be the movie index ordered by decreasing frequency and the Y -axis should be the number of ratings the movie has received.

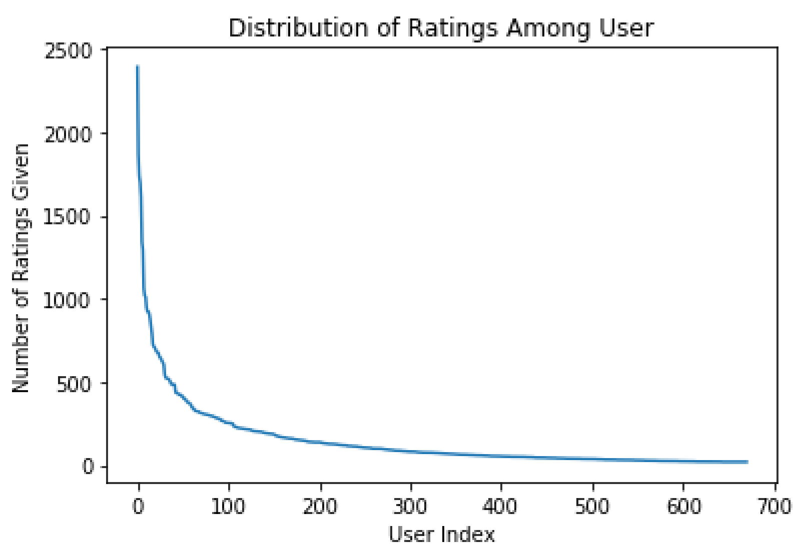
A: As we can see in the "Distribution of Ratings Among Movies" graph, most movies are not rated. This makes sense since the sparsity value, which we calculate in question 1, shows that the ratings matrix will be mostly zeros.



Question 4

Q: Plot the distribution of ratings among users. To be specific, the X-axis should be the user index ordered by decreasing frequency and the Y -axis should be the number of movies the user have rated.

A: From the “Distribution of Ratings Among User” graph, we can see that there are lots of users who rated a small amount of movies. It still makes sense because one person might not have the time to watch all 9125 movies.



Question 5

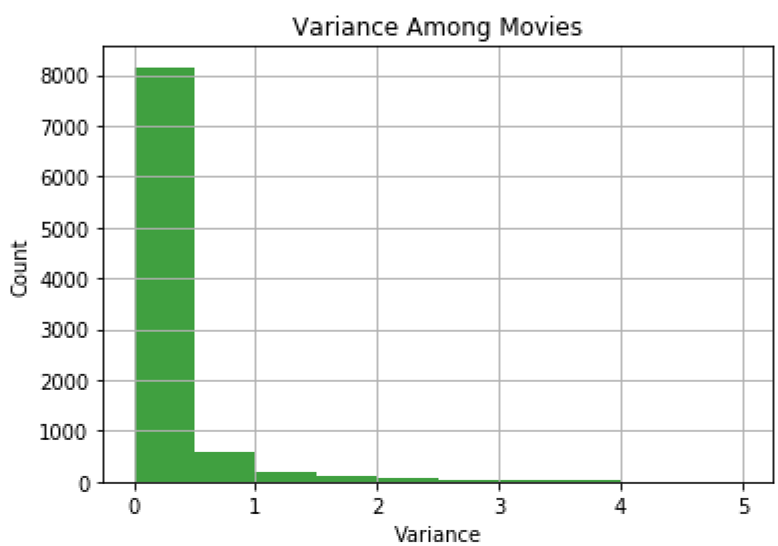
Q: Explain the salient features of the distribution found in question 3 and their implications for the recommendation process.

A: As we can seen from our graph in question 3, there is a spike in the in the beginning implying that a lot of users watched that movie[s]. It seems that the current recommendation process recommends popular movies to users. This makes our dataset very sparse which means we need to use other filtering methods to recommend movies.

Question 6

Q: Compute the variance of the rating values received by each movie. Then, bin the variance values into intervals of width 0.5 and use the binned variance values as the horizontal axis. Count the number of movies with variance values in the binned intervals and use this count as the vertical axis. Briefly comment on the shape of the histogram

A: In the histogram below, we can see that the variance value range 0 to 0.5 has a significantly higher count than the other variance value range. This is due to the fact that our R matrix is sparse, so most movies have zero ratings which means their variance should be low as well.



4 Neighborhood-based collaborative filtering

In this part, we designed and tested a k-NN collaborative filter. In the following sections, we will explore user-based neighborhood models, pearson-correlation coefficient and how they both relate to the k-NN collaborative filter.

4.1 User-based neighborhood models

For our neighborhood-based collaborative filtering, we will be using a user-based method. We identify similar users (i.e. users who have similar rating on the same items) to predict the rating on the item. We compute the similarity of the target user to all the other users. In the next section, we will go over how to compute the similarity between users.

4.2 Pearson-correlation coefficient

Pearson-correlation coefficient is how we compute the similarity between the rating vectors of the target user *u* and other user *v.* Some notation of this is:

- I_u : Set of item indices for which ratings have been specified by user u
- I_v : Same as last variable but for user v
- μ_u : Mean rating for user u
- r_{uk} : Rating of user u for item k

The Pearson-correlation coefficient between users u and v is defined by equation:

$$Pearson(u,v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)(r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}}$$

Question 7

Q: Write down the formula for μ_u in terms of I_u and r_{uk}

A: The formula for the mean rating is

$$\mu_u = \frac{1}{n} \sum_{k \varepsilon I_u} r_{uk} \ ,$$

where n is the number of ratings of user u for all items, and r_{uk} is the rating of user u for item k

Question 8

Q: In plain works, explain the meaning of $I_u \cap I_v$. Can $I_u \cap I_v = \varnothing$
(Hint: Rating matrix R is sparse)

A: $I_u \cap I_v$ is the intersection of two sets I_u and I_v . This means that $I_u \cap I_v$ contains all elements of I_v that also belong to I_u but no other elements. $I_u \cap I_v$ can equal to nothing because the matrix R is sparse meaning there are not alot of movie rated. There can be an empty set since there is a possibility of two users not watching the same set of movies.

4.3 k-Nearest neighborhood (k-NN)

k-Nearest neighbor of user u, denoted by P_u, is the set of k users with the highest Pearson-correlation coefficient with user u.

4.4 Prediction Function

The predicted rating of user u for item j, denoted by $\widehat{r_{uj}}$, is given by equation:

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u} Pearson(u,v)(r_{vj} - \mu_v)}{\sum_{v \in P_u} |Pearson(u,v)|}$$

Question 9

Q: Can you explain the reason behind mean-centering the raw ratings ($r_{vj} - \mu_v$) in the prediction function? (Hint: Consider users who either rate all items highly or rate all items poorly and the impact of these users on the prediction function)

A: There are some people who might rate a movie high or low (5 or 0.5) which can interfere with the the prediction. So the raw ratings ($r_{vj} - \mu_v$) is there to get rid of those bias ratings.

4.5 k-NN collaborative filter

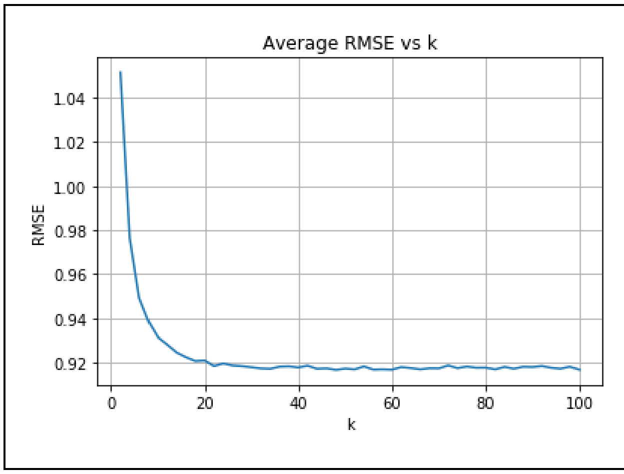
In this section, we will be using the built-in python functions for our predictions.

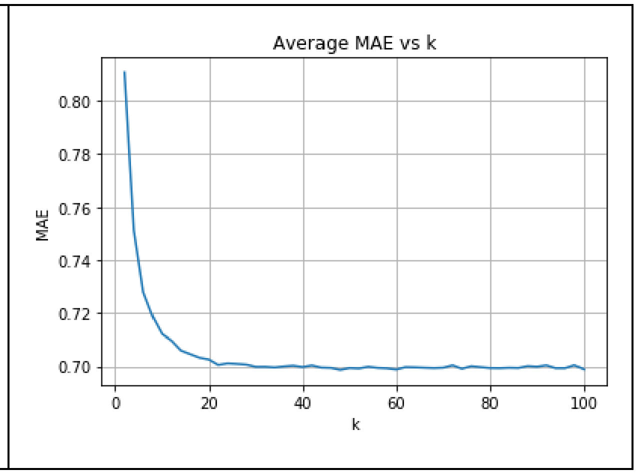
4.5.1 Design and test via cross-validation

Question 10

Q: Design a k-NN collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross validation. Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis).

A:





Question 11

Q: Use the plot from question 10, to find a ’minimum k’. Note: The term ’minimum k’ in this context means that increasing k above the minimum value would not result in a significant decrease in average RMSE or average MAE. If you get the plot correct, then ’minimum k’ would correspond to the k value for which average RMSE and average MAE converges to a steady-state value. Please report the steady state values of average RMSE and average MAE

A: As we can see in both the RMSE and MAE graph, the averages begin to converge around k equals 22.

4.6 Filter performance on trimmed test set

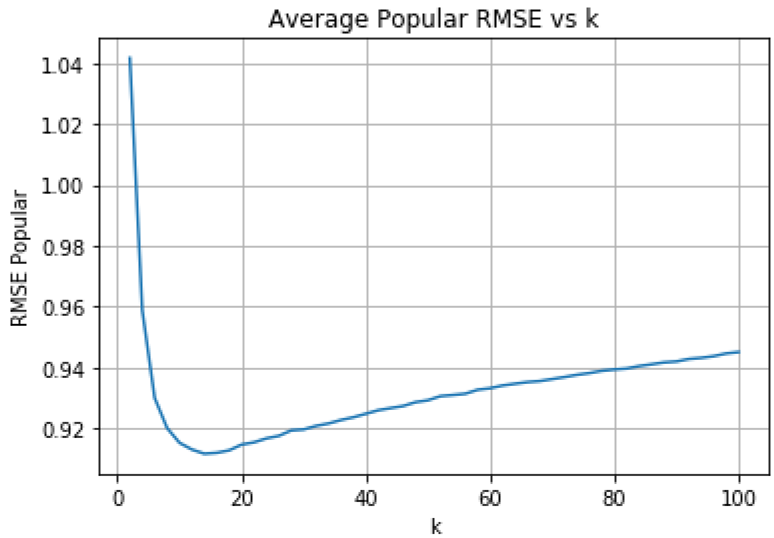
In this part, we trim our test set in the following way:

- Popular movie trimming: If a movie in the test set has received more than 2 rating in the entire dataset, then we predict the rating. Otherwise we do not predict the rating.
- Unpopular movie trimming: If a movie in the test set has receive less than or equal to 2 ratings in the entire test set, then we predict the rating. Otherwise we do not predict the rating.
- High variance movie trimming: If a movie in the testset has variance (of the rating values received) of 2 or greater AND received at least 5 ratings in the entire dataset, then we predict the rating. Otherwise, we do not predict the rating.

Question 12

Q: Design a k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE

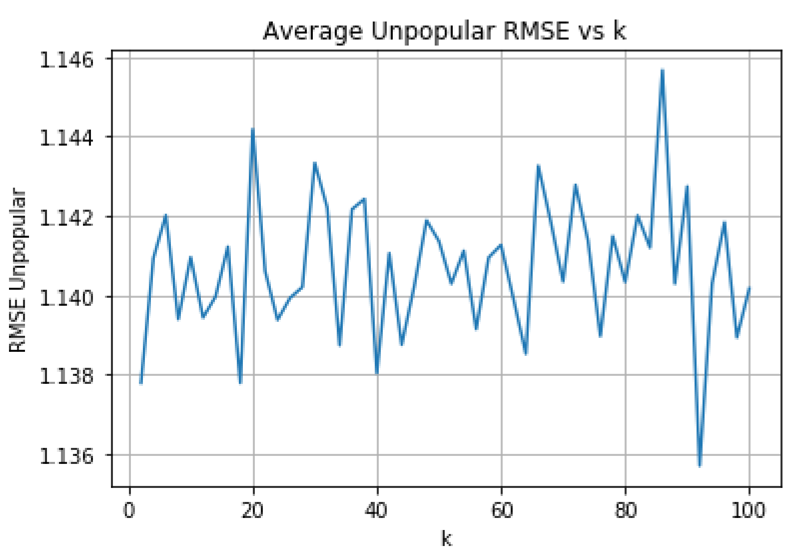
A: We can see from the graph below that the popular RMSE graph is convex.The minimum average RMSE is 0.91173583



Question 13

Q: Design a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate it's performance using 10-fold cross validation.Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE

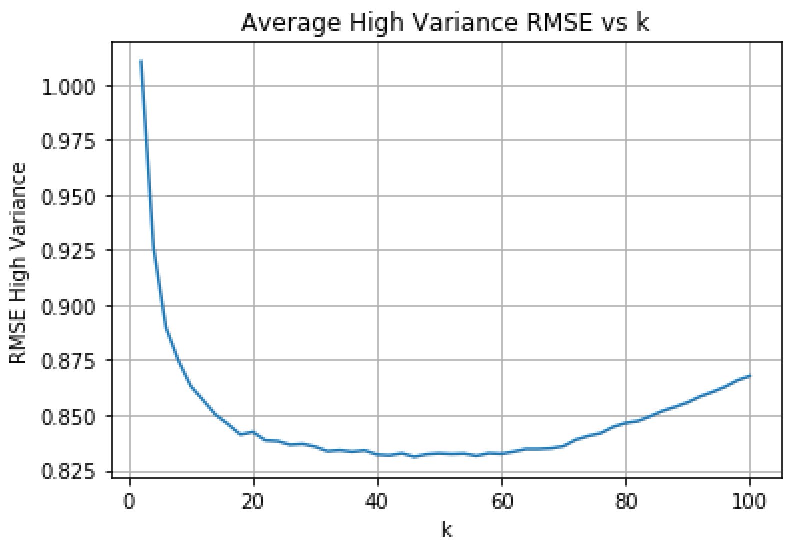
A: From the graph, we can see that the graph is not smooth. But the RMSE average values do stay in the 0.001 precision. This makes sense because the unpopular movies do not have that many ratings so it is probably more difficult to predict. The minimum average RMSE is 1.1356988



Question 14

Q: Design a k-NN collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

A: As we can see, the graph is convex. The minimum average RMSE is 0.83105120



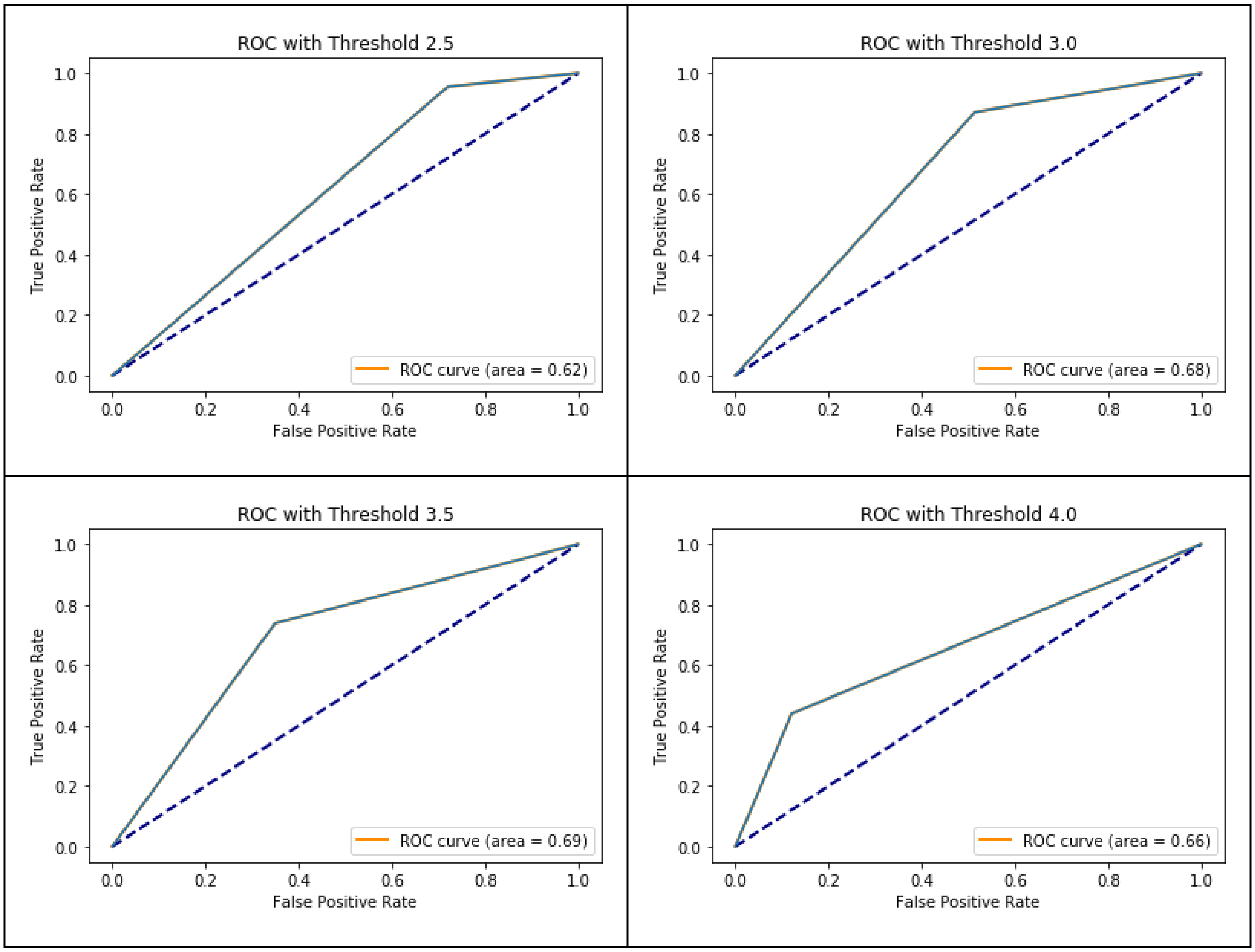
4.6.1 Performance evaluation using ROC curve

ROC is to identify binary classifier. In this section, we will binarize our ratings to determine if a user likes or dislikes a movie. We binarized our ratings with four different thresholds (2.5, 3, 3.5, 3). If the observed rating is below the rating, we set it to 0. If it is above or equal to the rating, we set it to 1. Having a value 1 implies the user likes the movie and 0 implies they dislike it.

Question 15

Q: Plot the ROC curves for the k-NN collaborative filter designed in question 10 for threshold values [2.5,3,3.5,4]. For the ROC plotting use the k found in question 11. For each of the plots, also report the area under the curve (AUC) value.

A: The AUC value is provided on the graph. We can see that the best performance happens when the threshold is 3 or 3.5.



5 Model-based collaborative filtering

Model-based collaborative filtering uses machine learning techniques to predict users' ratings of unrated items. In this part, we will explore latent factor based models for collaborative filtering.

5.1 Latent factor based collaborative filtering

Latent factor based models estimate the missing values of rating matrix R. This predicts new items the user may like based on the idea that significant rows and columns of the R matrix are correlated.

The matrix is approximated with

$$\underset{U,V}{\text{minimize}} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2 + \lambda \|U\|_F^2 + \lambda \|V\|_F^2$$

This optimization problem has a cost function with weight W_{ij} and regularization. Regularization is used to prevent overfitting. The regularization parameter λ controls the weight of the regularization and is always non-negative. The weight makes sure only known rates are taken in account.

$$W_{ij} = \begin{cases} 1, r_{ij} \text{ is known} \\ 0, r_{ij} \text{ is unknown} \end{cases}$$

This project will be exploring Non-negative matrix factorization (NNMF) and Matrix factorization with bias (MF with bias).

5.2 Non-negative matrix factorization (NNMF)

The non-negative matrix factorization is used for rating non-negatives matrices using non-negative U and V values. This is shown in the formula below.

$$\begin{aligned} &\underset{U,V}{\text{minimize}} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2 + \lambda \|U\|_F^2 + \lambda \|V\|_F^2 \\ &\text{subject to} \quad U \geq 0, V \geq 0 \end{aligned}$$

We can use algorithms like stochastic gradient descent (SGD) or alternating least-squares (ALS) to solve the above equation. ALS converges faster and is more stable than SVD, but we use SVD because the Python package used to build the NNMF-based collaborative Filter has only implemented SVD.

Question 16

Q: Is the optimization problem given by equation 5 convex? Consider the optimization problem given by equation 5. For U fixed, formulate it as a least-squares problem.
A: Yes, it is convex. Since equation 5 is a minimization problem, it makes sense for the equation to be to be convex. The equation needs to be convex in order to have a global minimum value. Least square problems are also convex. The least square formulation is below. The least square formulation will be:

$$min_H \{\|A - UV^T\|_F^2 + \lambda\|V^T\|_F^2$$

5.2.1 Prediction function

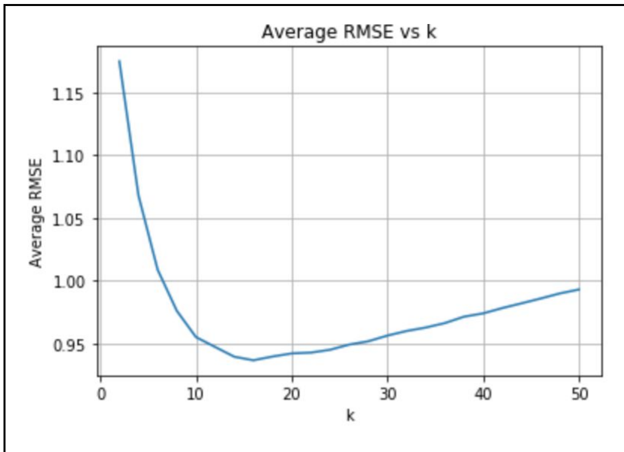
Use values from U and V to predict the ratings (user = i, item = j).

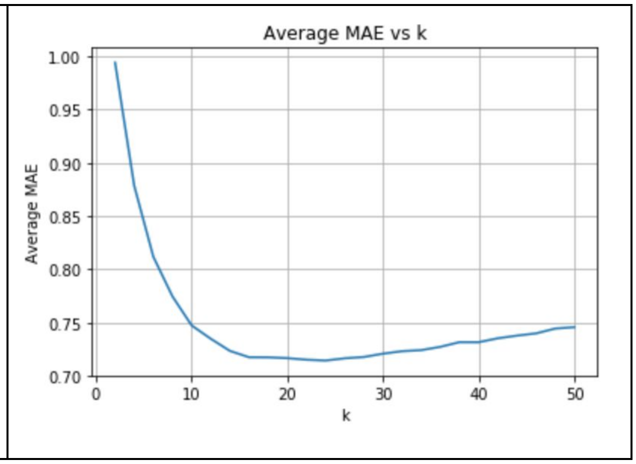
$$\hat{r}_{ij} = \sum_{s=1}^{k} u_{is} \cdot v_{js}$$

5.2.2 Design and test via cross-validation

Question 17

Q: Design a NNMF-based collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross-validation. Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. Plot the average RMSE (Y-axis) against k (X-axis) and the average MAE (Y-axis) against k (X-axis). For solving this question, use the default value for the regularization parameter.
A: The minimum average RMSE is 0.936827102876 at k = 16. The minimum average MAE is 0.714334521549 at k = 24.





Question 18

Q: Use the plot from question 17, to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE. Is the optimal number of latent factors same as the number of movie genres?
A: Minimum average RMSE is 0.936827102876 at k = 16. MAE average at k = 16 is 0.71743227389.
Minimum average MAE is 0.714334521549 at k = 24. RMSE average at k = 24 is 0.94517638334.

Since the averages for MAE is a difference by 0.001 verses RMSE the averages for RMSE is a difference changed by 0.01, we picked the optimal to be k = 16. The optimal number does not match the number of genres (18 genres), but this is quite close. This difference may have been due to noise.

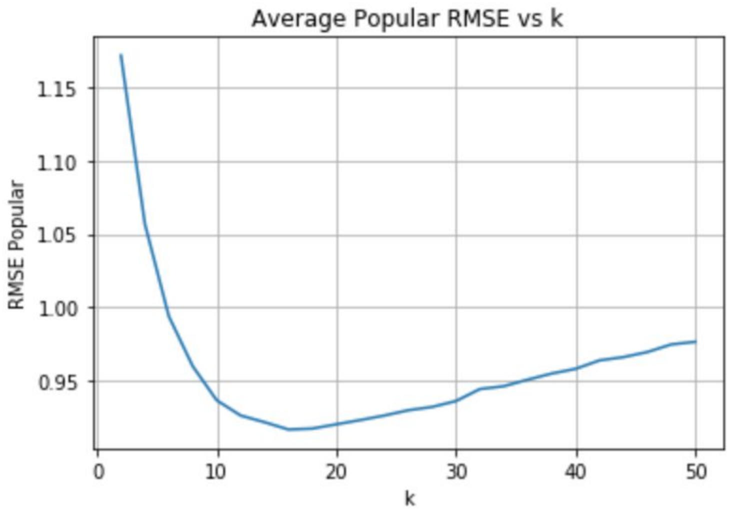
5.2.3 NNMF filter performance on trimmed test set

In this part, we test the performance of the filter in predicting the ratings of the movies in the trimmed test set

Question 19:

Q: Design a NNMF collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

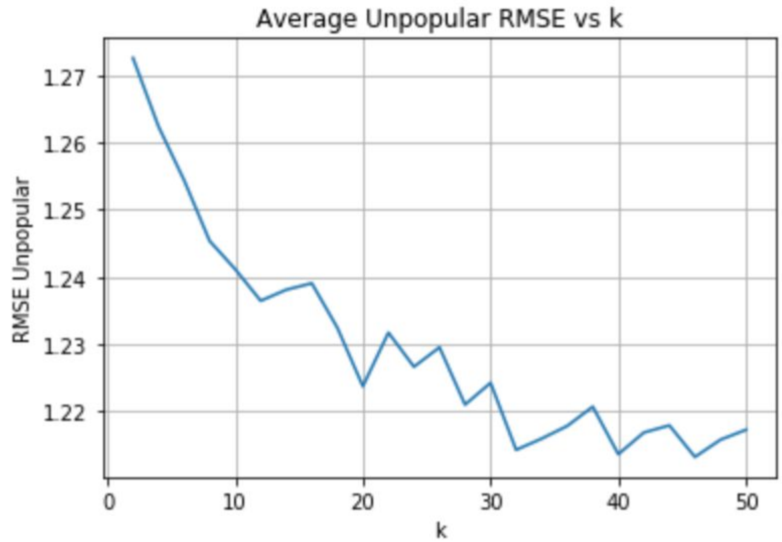
A: The result is a convex plot. Minimum average RMSE is 0.916175602019 at k = 16. We can confirm this in the plot below.



Question 20:

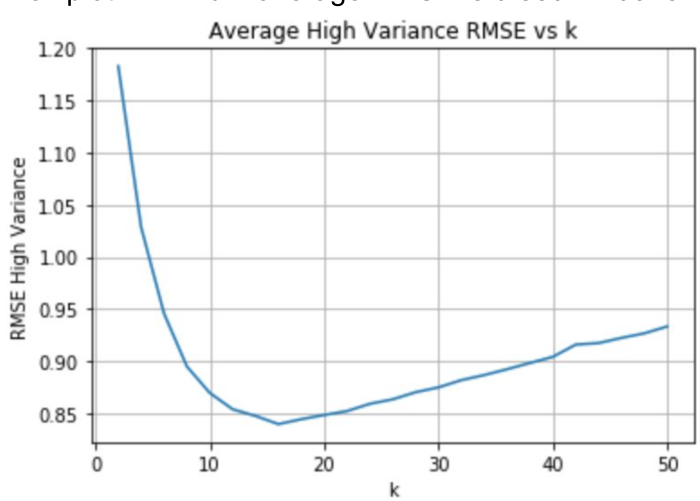
Q: Design a NNMF collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

A: This results in a plot with a generally negative slope. Minimum average RMSE is 1.21321146118 at k = 46. The average unpopular RMSE values are higher than the popular RMSE. This is because the unpopular matrix is more sparse (assuming unrated movies are considered unpopular).



Question 21:

Q: Design a NNMF collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.
A: The result is a convex plot. Minimum average RMSE is 0.839741052528 at k = 16.

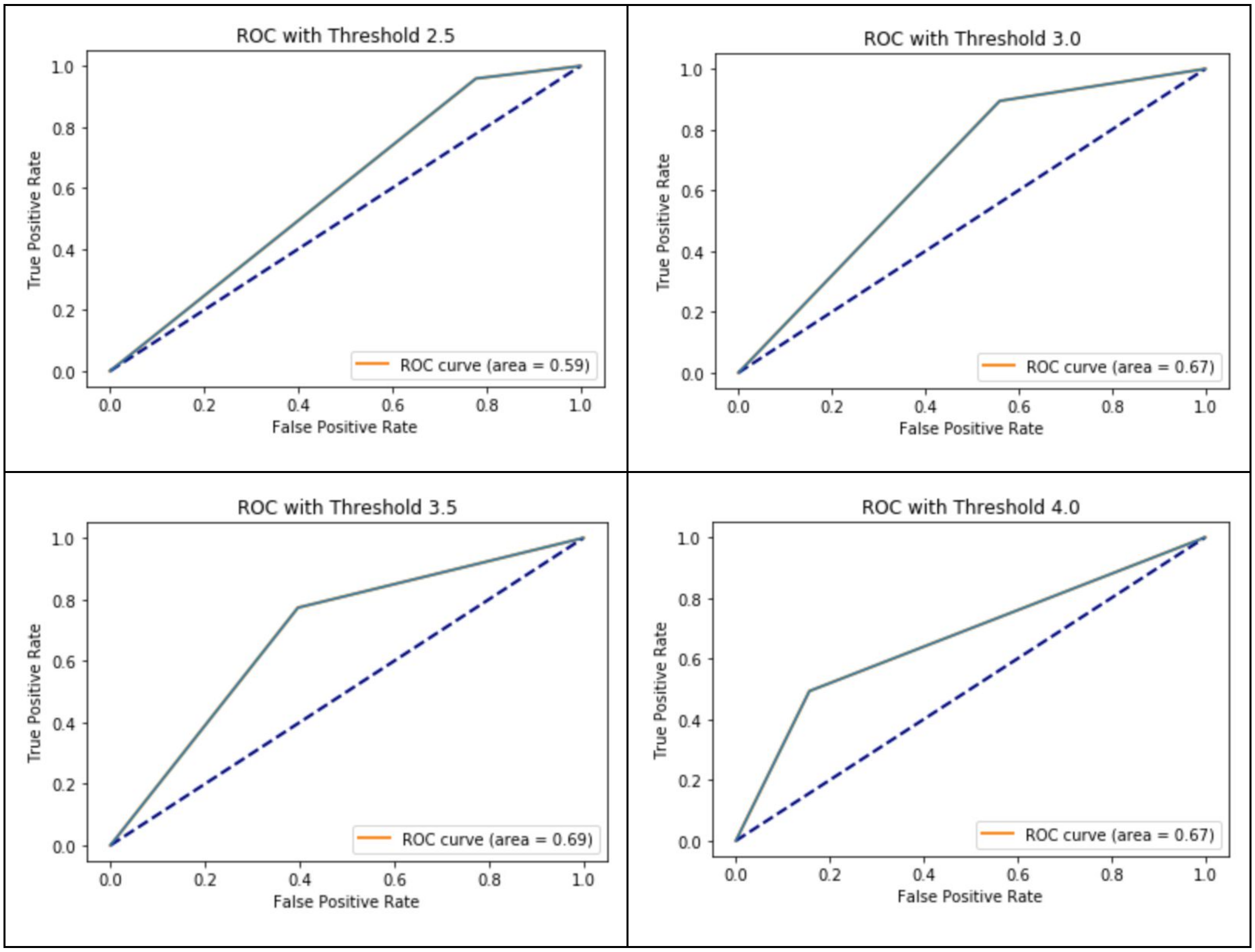


5.2.4 Performance evaluation using ROC curve

In this part, we evaluate the performance of the NNMF-based collaborative filter using the ROC curve.

Question 22:

Q: Plot the ROC curves for the NNMF-based collaborative filter designed in question 17 for threshold values [2.5, 3, 3.5, 4]. For the ROC plotting use the optimal number of latent factors found in question 18. For each of the plots, also report the area under the curve (AUC) value.
A: According to the AUC values, threshold 3.5 results in the best ROC. All the AUC values are similar, but threshold 3.5 had the value closest to 1.



5.2.5 Interpretability of NNMF

NNMF is useful due to its high level of interpretability which gives understanding user-item interactions. Here, we explore the connection between the latent factors and the movie genres.

Question 23:

Q: Perform Non-negative matrix factorization on the ratings matrix R to obtain the factor matrices U and V , where U represents the user-latent factors interaction and V represents the movie-latent factors interaction (use k = 20). For each column of V, sort the movies in descending order and report the genres of the top 10 movies. Do the top 10 movies belong to a particular or a small collection of genre? Is there a connection between the latent factors and the movie genres?

A: We can see the top 10 movies of the first four latent factors below. They all hold higher numbers of movies with genres Drama, Comedy, Romance, and Thriller. This can lead to assume that these types of movie rank high in these latent factors. We can also see certain genres do not appear for some latent factors. For example, Musical ranks high in Column 4 and appears in all columns except Column 3. Some genres don’t show up at all, like Western. Many genres are actually a combination of multiple genres, which can mean the broader the genre, the high chance it has to rank higher in the latent factors.

Column 1	Column 2
'Horror|Thriller' 'Adventure|Comedy|Thriller' 'Crime|Drama|Film-Noir|Mystery' 'Drama' 'Comedy|Fantasy|Musical|Romance’ 'Comedy' 'Documentary|War' 'Action|Fantasy|Horror|Thriller' 'Action|Adventure|War' 'Animation|Comedy|Musical'	'Action|Adventure|Sci-Fi' 'Animation' 'Action|Adventure|Children|Fantasy' 'Comedy|Drama' 'Documentary|Musical' 'Action|Adventure|Fantasy|Romance|IMAX' 'Drama' 'Drama|Fantasy|Mystery|Romance' 'Crime|Drama|Thriller' 'Comedy|Horror|Romance'

Column 3	Column 4
'Comedy|Drama|Fantasy|Romance' 'Drama' 'Comedy|Drama|War' 'Drama' 'Crime|Drama|Mystery|Romance|Thriller' 'Children|Comedy' 'Drama|War' 'Drama|Romance' 'Action|Drama|Romance' 'Action|Mystery|Thriller'	'Musical' 'Adventure|Children|Drama' 'Thriller' 'Comedy|Musical|Romance' 'Children|Comedy|Fantasy' 'Comedy' 'Comedy|Horror' 'Action|Comedy' 'Drama|Romance' 'Horror|Thriller'

5.3 Matrix factorization with bias (MF with bias)

In MF with bias, we modify the cost function from before by adding bias term for each user and item. By adding this, we get the equation:

$$\underset{U,V,b_u,b_i}{\text{minimize}} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(r_{ij}-(UV^T)_{ij})^2 + \lambda\|U\|_F^2 + \lambda\|V\|_F^2 + \lambda\sum_{u=1}^{m} b_u^2 + \lambda\sum_{i=1}^{n} b_i^2$$

where b_u is the bias of user u and b_i is the bias of item i

5.3.1 Prediction function

The solution to the optimization problem in the previous equation is

$$\hat{r}_{ij} = \mu + b_i + b_j + \sum_{s=1}^{k} u_{is} \cdot v_{js}$$

where μ is the mean of all ratings, b_i is the bias of user i, b_j is the bias of item j, r_{ij} is the predicted rating of user i for item j

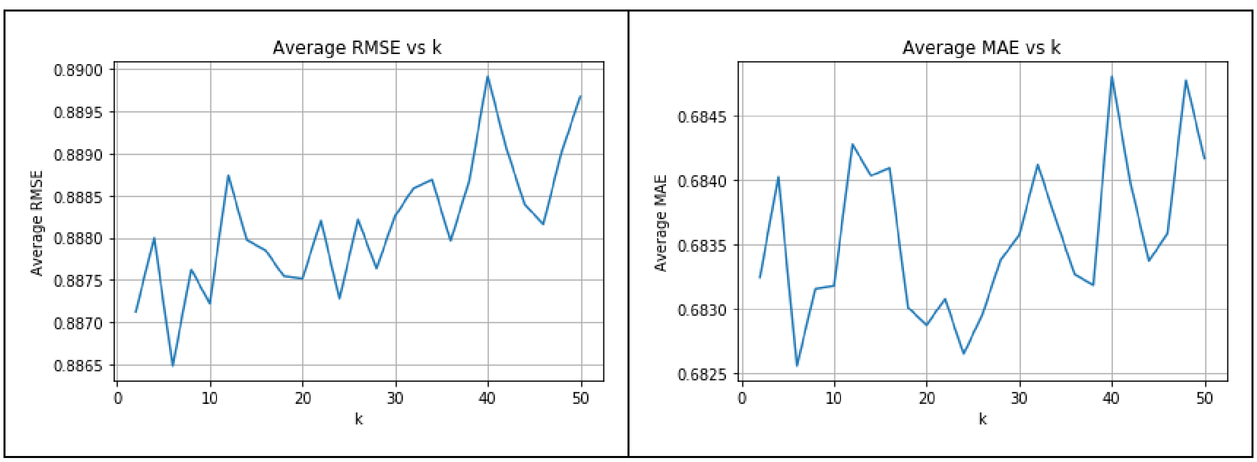
5.3.2 Design and test via cross-validation

In this part, we design and tested a MF with bias collaborative filter.

Question 24

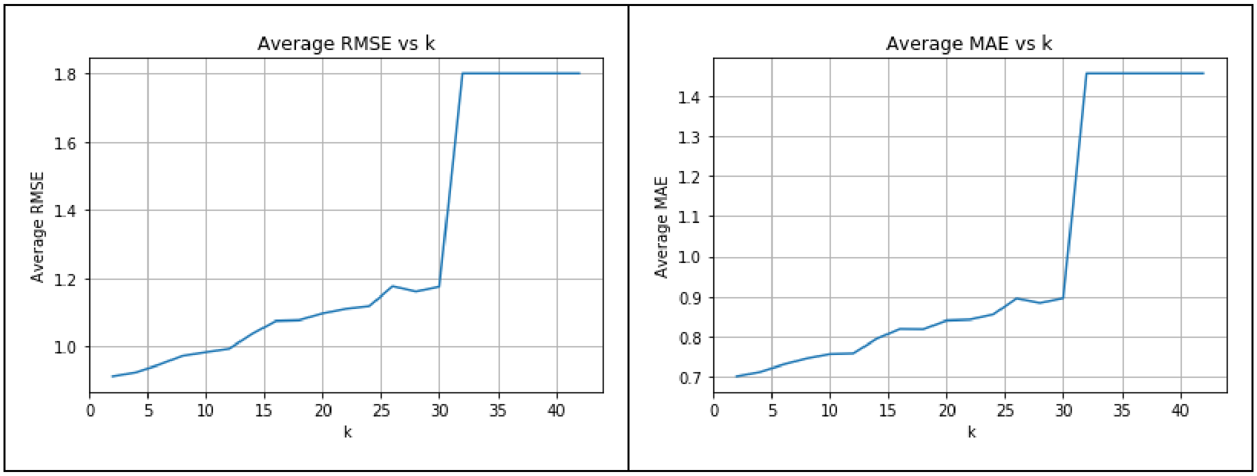
Q: Design a MF with bias collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate it's performance using 10-fold cross-validation. Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. Plot the average RMSE (Y-axis) against k (X-axis) and the average MAE (Y-axis) against k (X-axis). For solving this question, use the default value for the regularization parameter.

A: Below is the graph of the average RMSE and MAE. The minimum RMSE is 0.8864 at k = 6. The minimum MAE is 0.6825 at k = 6.



<u>Further Analysis</u>

In addition, we tested the graph with an initial mean. The initial mean is the mean of the ratings in the R matrix. We saw that the graph is smoother, but the runtime took +10 hours. We found that the minimum RMSE is 0.9117 at k = 2 and minimum MAE is 0.700323 at k = 2. Below are the graphs.



Question 25

Q: Use the plot from question 24, to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE.

A: Again, the minimum RMSE is 0.8864 at k = 6, and the minimum MAE is 0.6825 at k = 6. Therefore, we picked k = 6 to be the optimal number of latent factors.

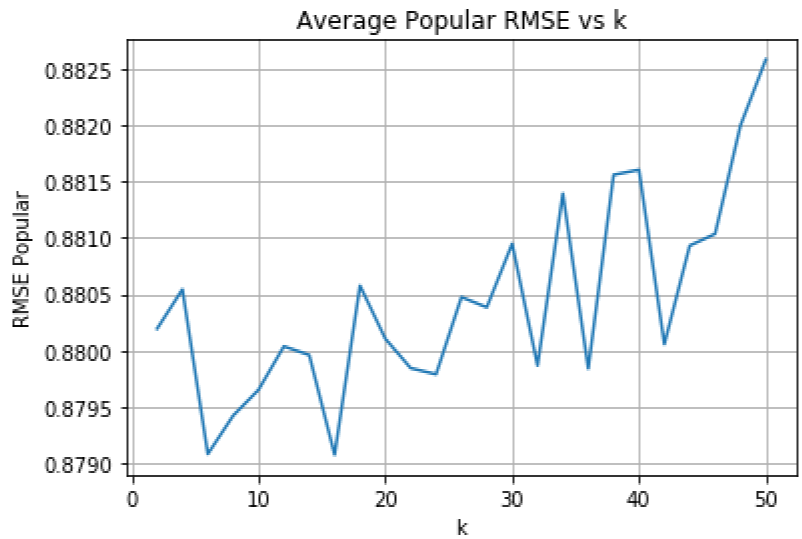
5.3.3 MF with bias filter performance on trimmed test set

We will use the same trimming operations before for popular, unpopular, and high variance. Look at section 4.6 to see the trimming criteria.

Question 26

Q: Design a MF with bias collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE

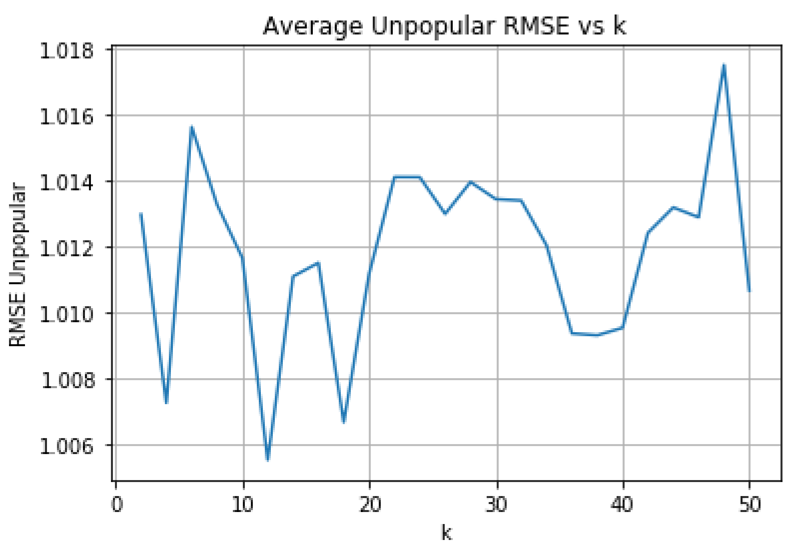
A: From the graph below, we see that the graph fluctuates. However, we can see it increasing when k increases as well. The minimum average RMSE is 0.87907 at k = 16.



Question 27

Q: Design a MF with bias collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE

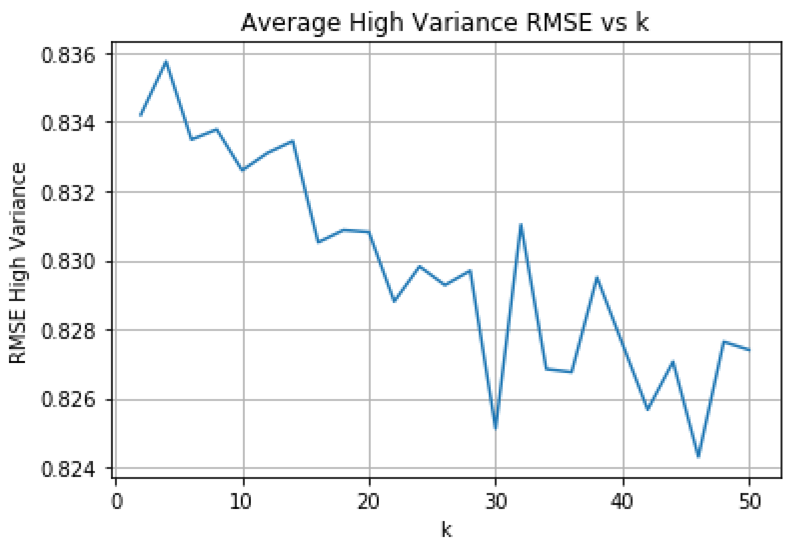
A: Again, just like the popular RMSE graph, we can see that the data fluctuates. This time it seems like thater is not really a pattern when the k value is low. But if you increase the k above 20, then the RMSE gets higher. The minimum average RMSE is 1.0055 and happens at k = 12.



Question 28

Q: Design a MF with bias collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation.Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

A: For the high variance graph, we can see that as k increases, the RMSE value decreases. The minimum average RMSE is 0.8243 at k = 46.



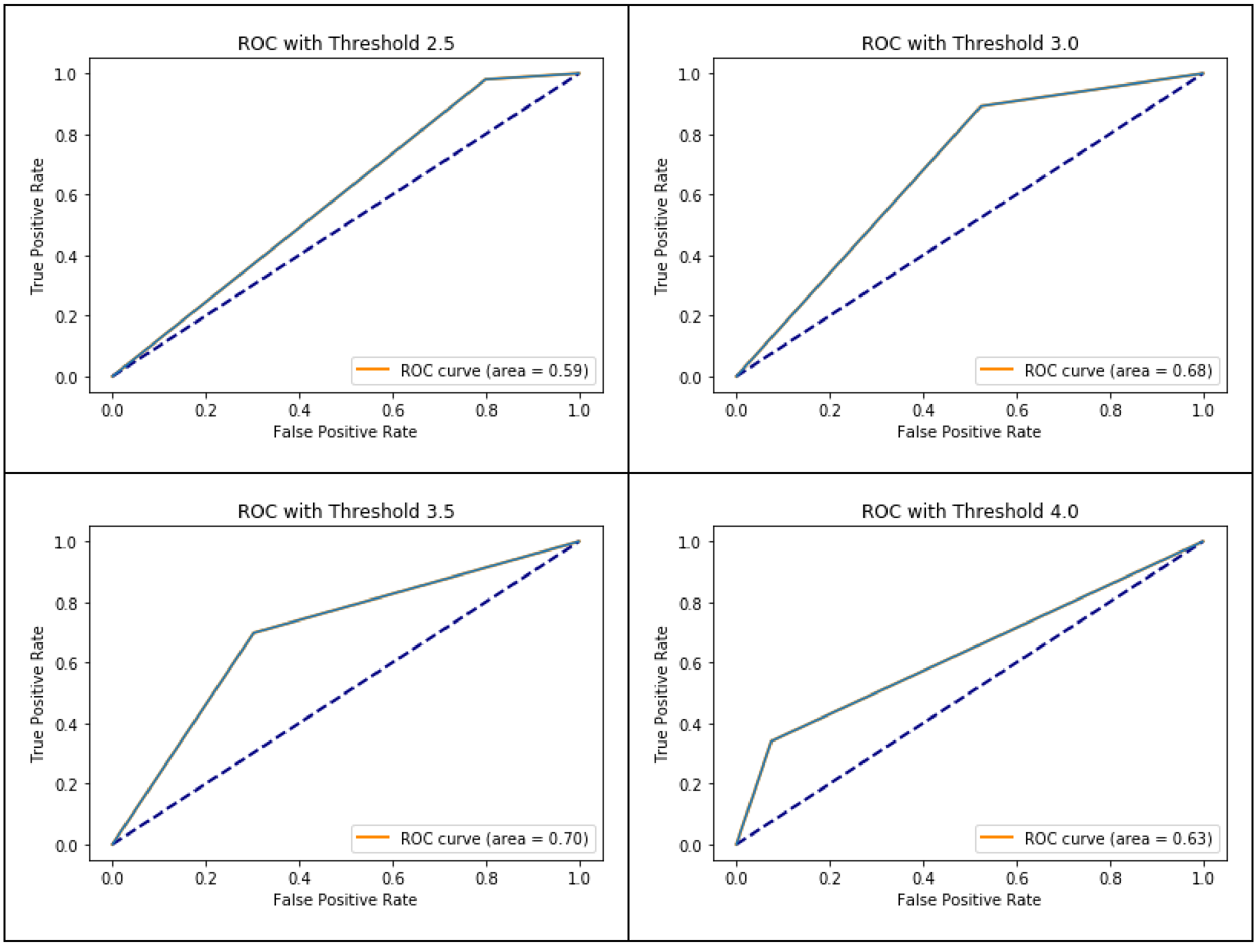
5.3.4 Performance evaluation using ROC curve

In this part, we will evaluate the performance of the MF with bias collaborative filter using the ROC curve.

Question 29

Q: Plot the ROC curves for the MF with bias collaborative filter designed in question 24 for threshold values [2.5,3,3.5,4]. For the ROC plotting use the optimal number of latent factors found in question 25. For each of the plots, also report the area under the curve (AUC) value.

A: From the graphs we can see that the performance of the MF with bias is pretty good. The performance is best when we make our threshold 3.5. On the graph you will find the AUC value.



6 Naive collaborative filtering

In this part, we will implement a naive collaborative filter to predict the ratings of the movies. A naive collaborative filter uses the mean of user's movie ratings and uses the mean as the prediction for an item. That means that there is no training data for the section.

6.1 Prediction function

The predicted rating of user i for item j, denoted by $\hat{r}_{ij}$ is given by the equation:

$$\hat{r}_{ij} = \mu_i$$

where μ_i is the mean rating of user i.

6.2 Design and test via cross-validation

Question 30

Q: Design a naive collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

A: The average overall RMSE is 3.4464. This is a pretty high error, but we expect that because the filter does not use any other data but the mean to predict.

6.3 Naive collaborative filter performance on trimmed test set

We will use the same trimming operations before for popular, unpopular, and high variance. Look at [section 4.6] to see the trimming criteria.

Question 31

Q: Design a naive collaborative filter to predict the ratings of the movies in the popular movie trimmed test set and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

A: The overall average RMSE for popular movies is 3.396423. Again, this value is pretty high but better than the average RMSE value for all the testset.

Question 32

Q: Design a naive collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

A: The overall average RMSE for unpopular movies is 3.16296. This value is best out of the trimming section.

Question 33

Q: Design a naive collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.

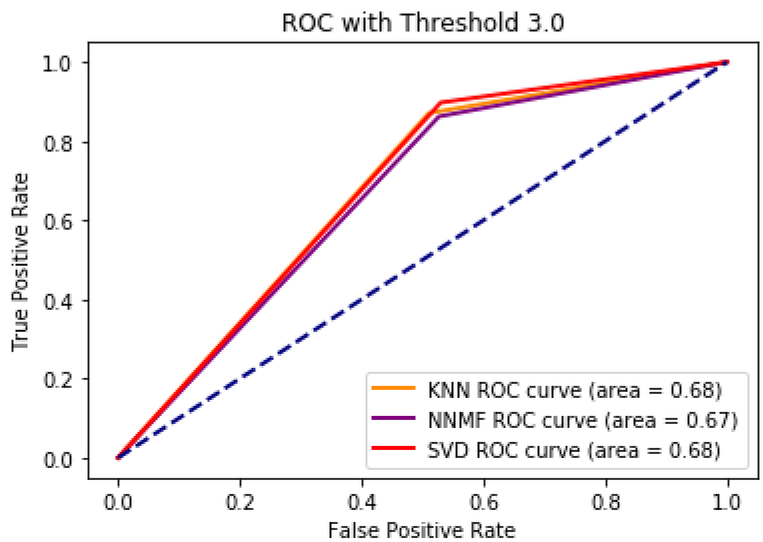
A: The overall average RMSE for high variance movies is 3.33464.

7 Performance comparison

Question 34

Q: Plot the ROC curves (threshold = 3) for the k-NN, NNMF, and MF with bias based collaborative filters in the same figure. Use the figure to compare the performance of the filters in predicting the ratings of the movies.

A: As we can see in the graph below, all three filters have similar performance in predicting the with NNMF being slightly worst.



8 Ranking

In this part, we will be solving the ranking version of the problem. The ranking version of problem recommends the top k items for a particular user. To solve this problem, we first solve the prediction problem and then rank the predictions. We already solved the ranking problems in the previous sections.

8.1 Ranking predictions

The ranking can be done in the following manner:

- For each user, compute its predicted ratings for all the items using one of the collaborative filtering techniques. Store the predicted ratings as a list L.
- Sort L in descending order, the item with the highest predicted ratings appear first and the item with the lowest predicted ratings appear last.

- Select the first t-items from the sorted list to recommend to the user

8.2 Evaluating ranking using precision-recall curve

In this part, we use the precision-recall to evaluation the relevance of the ranked list. In this we have some notation:

- S(t): The set of items of size t recommended to the user. In this recommended set, ignore (drop) the items for which we don't have a ground truth rating.
- G: The set of items liked by the user (ground-truth positives)

Then we get the formula for precision and recall:

$$Precision(t) = \frac{|S(t) \cap G|}{|S(t)|} \tag{12}$$

$$Recall(t) = \frac{|S(t) \cap G|}{|G|} \tag{13}$$

Question 35

Q: Precision and Recall are defined by the mathematical expressions given by equations 12 and 13 respectively. Please explain the meaning of precision and recall in your own words.

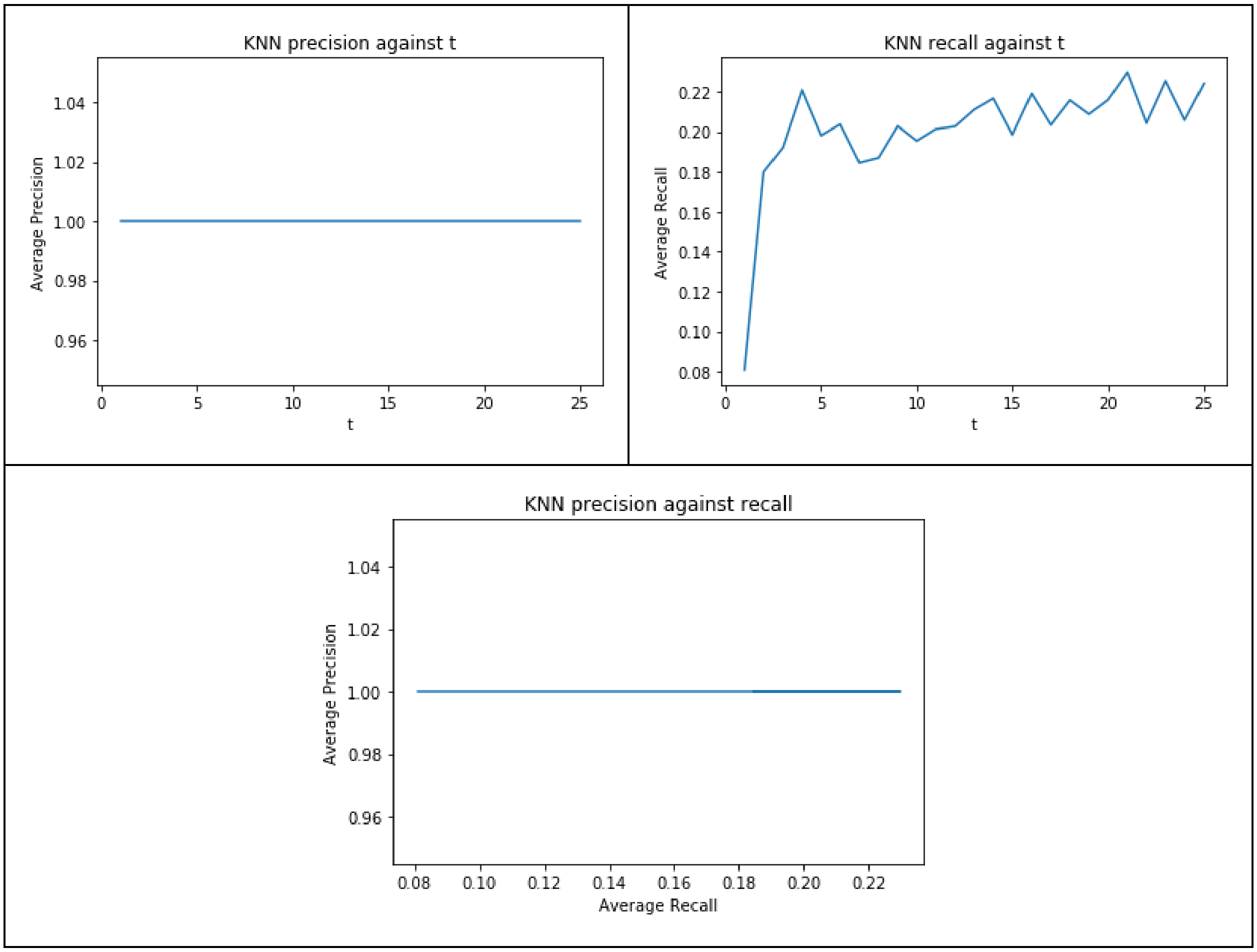
A: For the equation 12, we are comparing the movies that intersect both S(t) and G only and no other element to S(t). We get a value indicating how many good recommendations were in S(t).

Similarly, equation 13 compares the movies that intersect both S(t) and G only and no other element to G. We get a value indication how many of the recommendations did we get compared to the ground truth values.

Question 36

Q: Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using k-NN collaborative filter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use the k found in question 11 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.

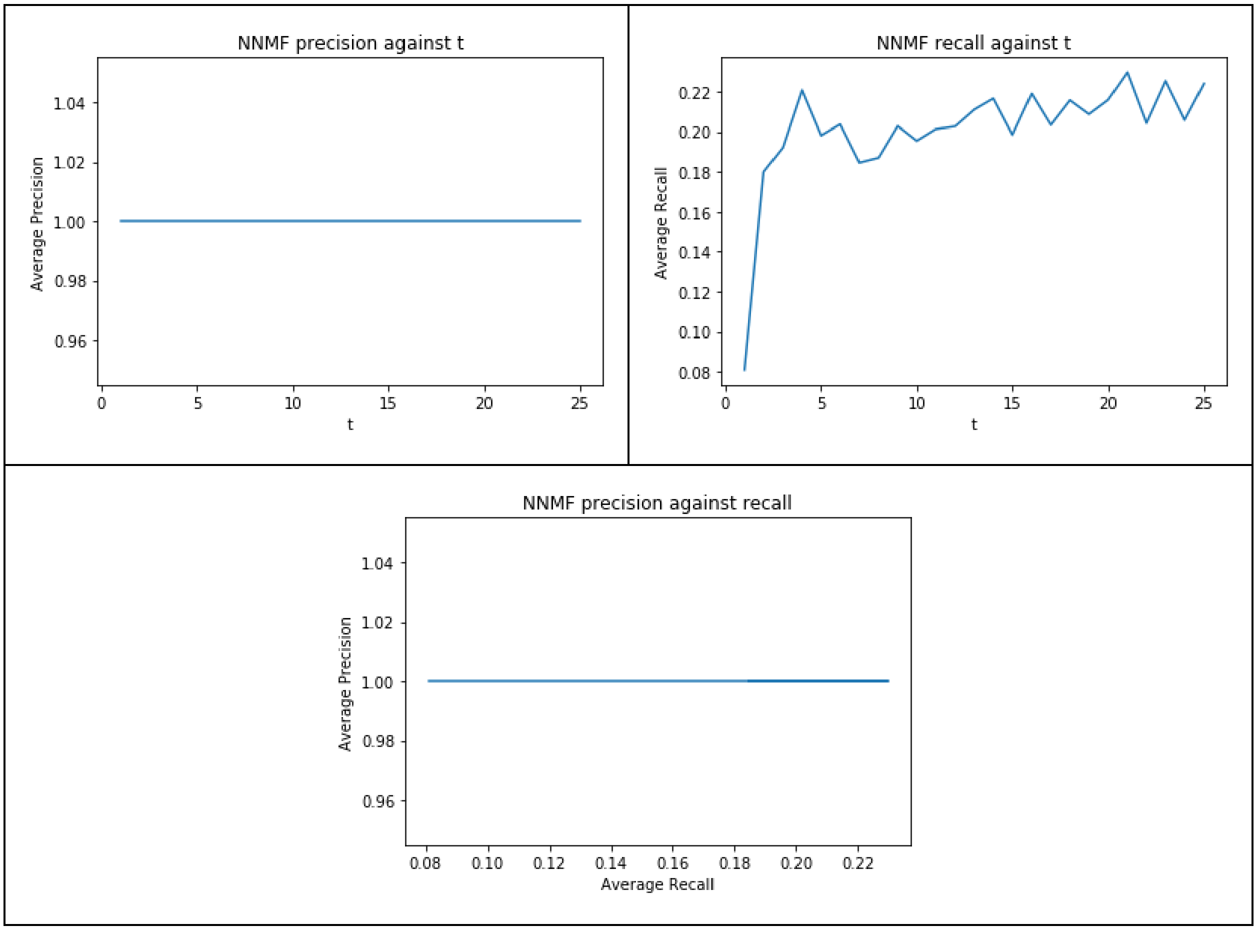
A: We can see that the “KNN precision against t” graph, it is a constant value which means that we get really good predictions since it is all the ground truth. The “KNN recall against t” graph just shows the percentage of the moves we recommended. The “KNN precision against recall’ is similar shaped to the first graph. Overall the ranking is great.



Question 37

Q: Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using NNMF-based collaborative filter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use optimal number of latent factors found in question 18 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.

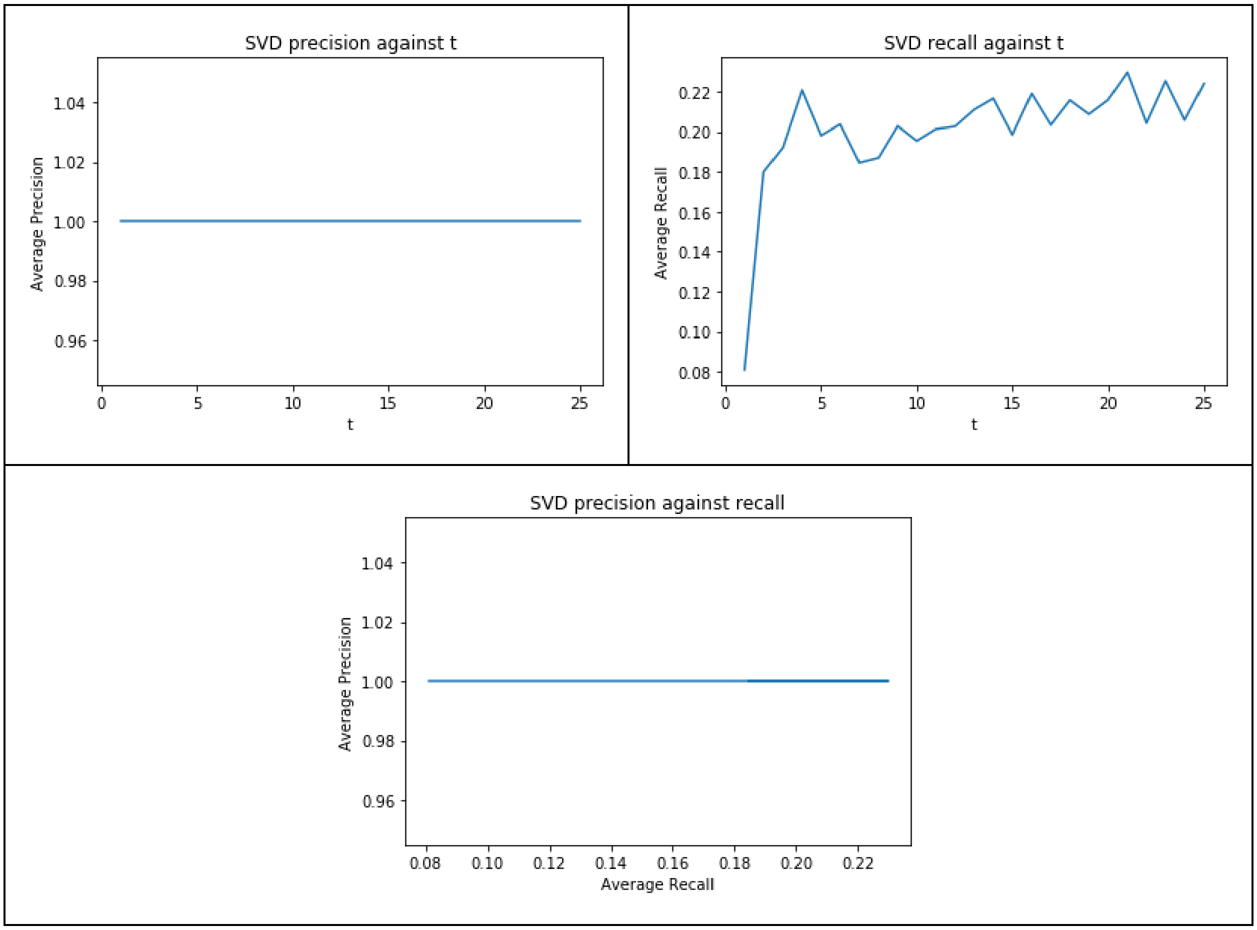
A: Same as the last question: We can see that the “KNN precision against t” graph, it is a constant value which means that we get really good predictions since it is all the ground truth. The “KNN recall against t” graph just shows the percentage of the moves we recommended. The “KNN precision against recall’ is similar shaped to the first graph. Overall the ranking is great.



Question 38

Q: Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using MF with bias-based collaborative filter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use optimal number of latent factors found in question 25 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.

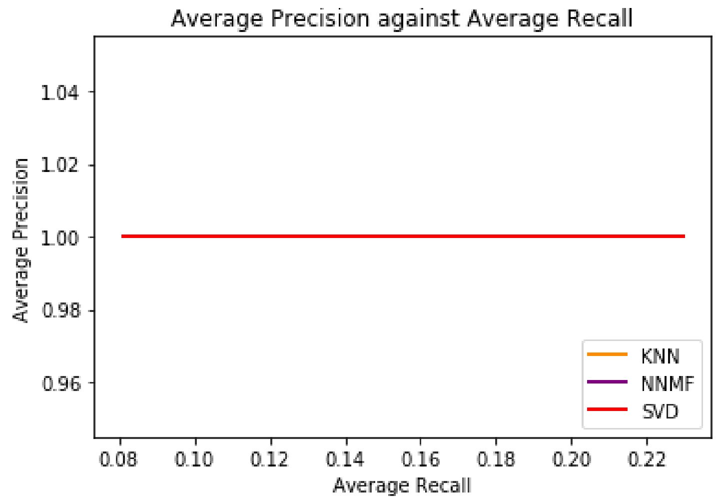
A: Same as the last question: We can see that the “KNN precision against t” graph, it is a constant value which means that we get really good predictions since it is all the ground truth. The “KNN recall against t” graph just shows the percentage of the moves we recommended. The “KNN precision against recall’ is similar shaped to the first graph. Overall the ranking is great.



Question 39

Q: Plot the precision-recall curve obtained in questions 36,37, and 38 in the same figure. Use this figure to compare the relevance of the recommendation list generated using k-NN, NNMF, and MF with bias predictions.

A: As we can see, all three collaborative filtering to recommend movies work really well. All the recommendations fall into the ground truth which shows all filters work well.


Average Precision against Average Recall
1.04
1.02
1.00
0.98
0.96
Average Precision
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
Average Recall
KNN
NNMF
SVD