Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839

ECE 219 Project 3 Report

In this project, we are tasked with creating a recommendation system to estimate the ratings of the movies in a given dataset.

The first step is to create an R matrix containing the users as rows and movies as columns with the values of the matrix determined by the rating user at the ith row gave for the movie at the jth column.

PART 3

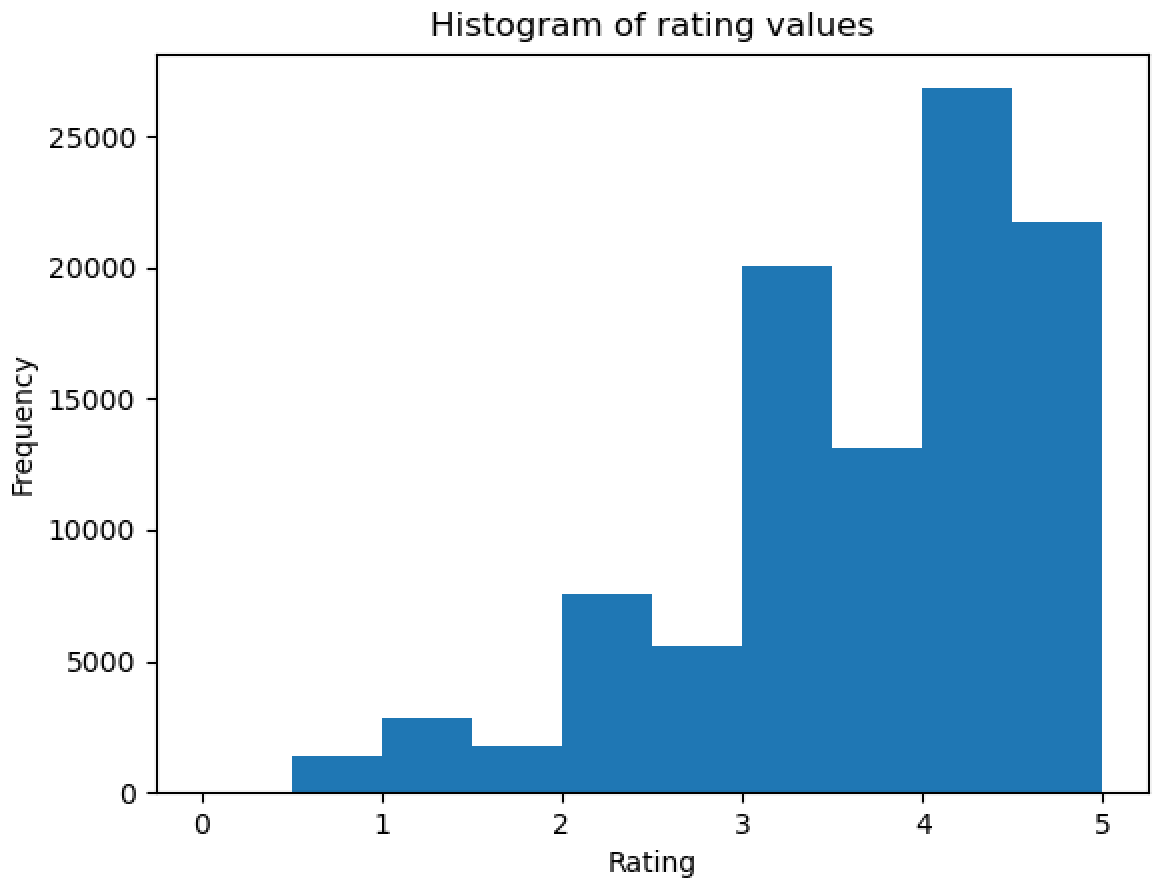
Question 1

A) By utilizing the sparsity formula given as:

$$Sparsity = \frac{\#\, available\, ratings}{\#\, possible\, ratings}$$

We can calculate the sparsity of our data. The result is found as: **0.0169996.**

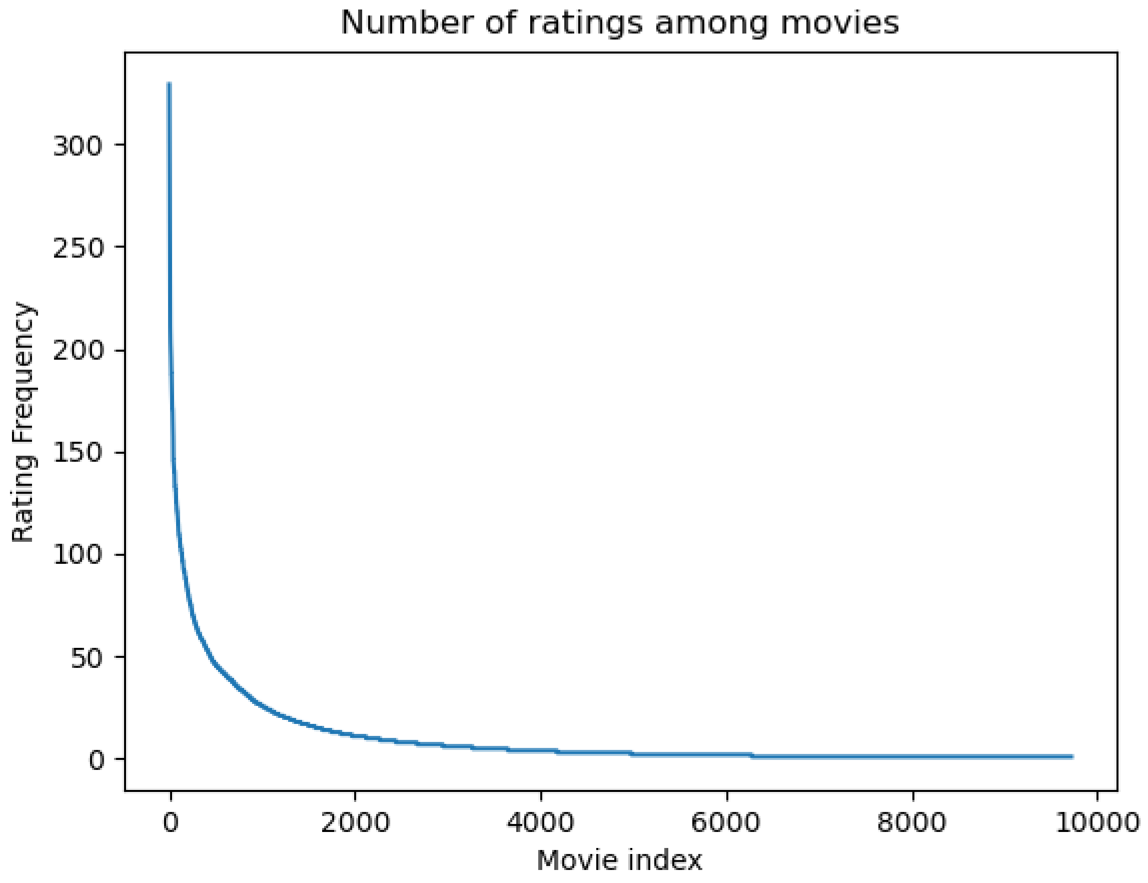
B) We can plot the desired histogram as follows:



From the histogram, we can determine that the majority of the ratings are high numbers, between 3 and 5 with the most amount of ratings between 4 and 4.5. This is expected as most users examine the movie before they watch them to make sure that they will like the movie. Since the users rate the movies after they have seen them, it is more likely for them to score the movies higher.

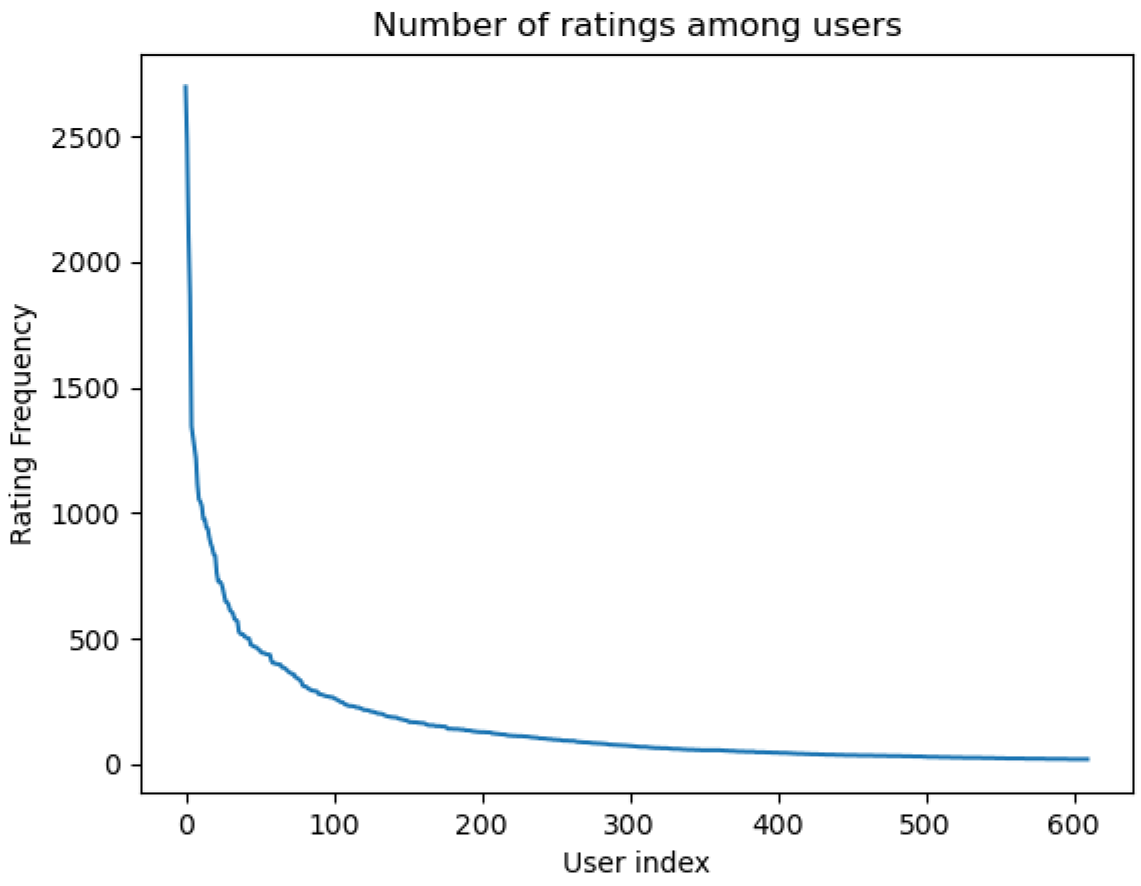
C)

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



As can be seen, there is indeed a monotonically decreasing trend between the frequency of a movie and the movie index after the movies have been sorted by their frequencies.

D)



As before, there is a monotonically decreasing trend between the frequency a user has rated a movie and the user index after the users have been sorted by their frequencies.

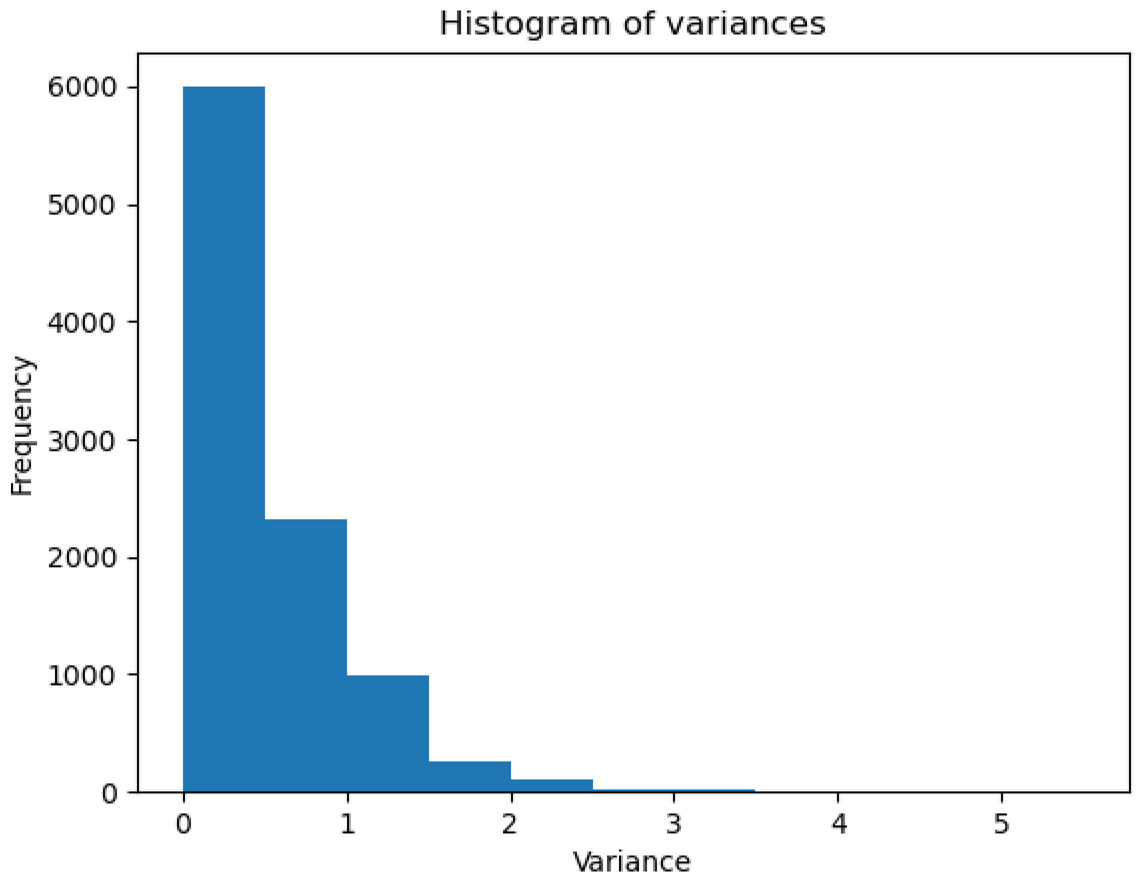
E)

By examining the figure in Part C, we can see that a small number of movies have received the majority of the ratings, which means that most of the data in our dataset is related to a small amount of movies. Furthermore, from the figure in part D, we can understand that the majority of the ratings have been given by a small number of users. The combination of these effects implies that the dataset is quite sparse.

If we try to fit a model to this set as it is, the model would overfit to the data entries corresponding to frequent users and movies. Therefore, our model would not be able to accurately function when tested on movies that have not been rated frequently and when tested on data entries given by users who have not been adequately represented in the dataset. This would lead to bad generalization due to the bias present in the dataset. To overcome this issue, we need to make use of regularization techniques.

F)

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



As can be seen from the graph, most of the variance of the movie ratings is low (between 0 and 1). This implies that the ratings of a movie given by a number of users do not fluctuate much, indicating that any results reached for a movie can be applicable to the behavior of a high number of users.

PART 4

In this part of the project, we are tasked with designing a neighbor based collaborative filtering system.

Question 2

A)

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

B)

$I_u \cap I_v$ corresponds to all the movies that have been rated by both user u and user v. The intersection can be null if the two users have not rated the same movie. Due to the sparsity of the dataset, this situation is quite likely.

k- Nearest Neighbors filtering

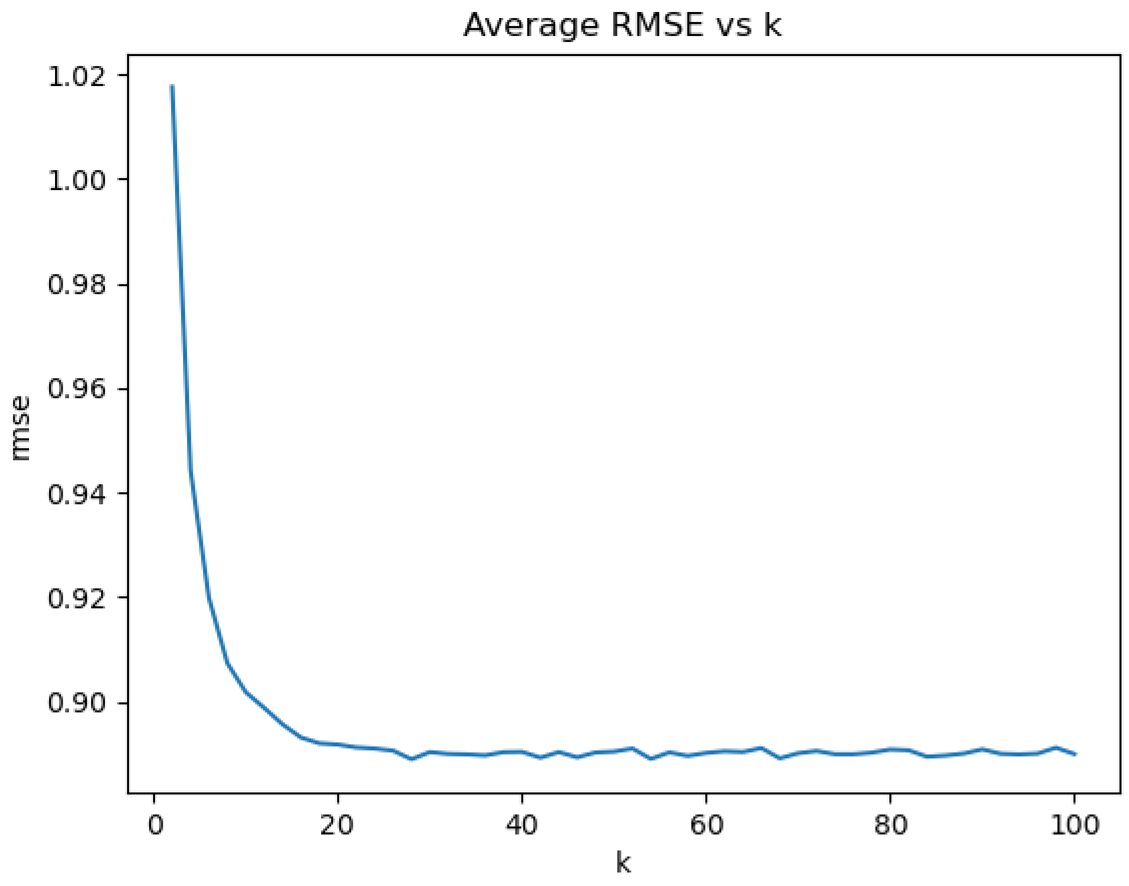
Question 3

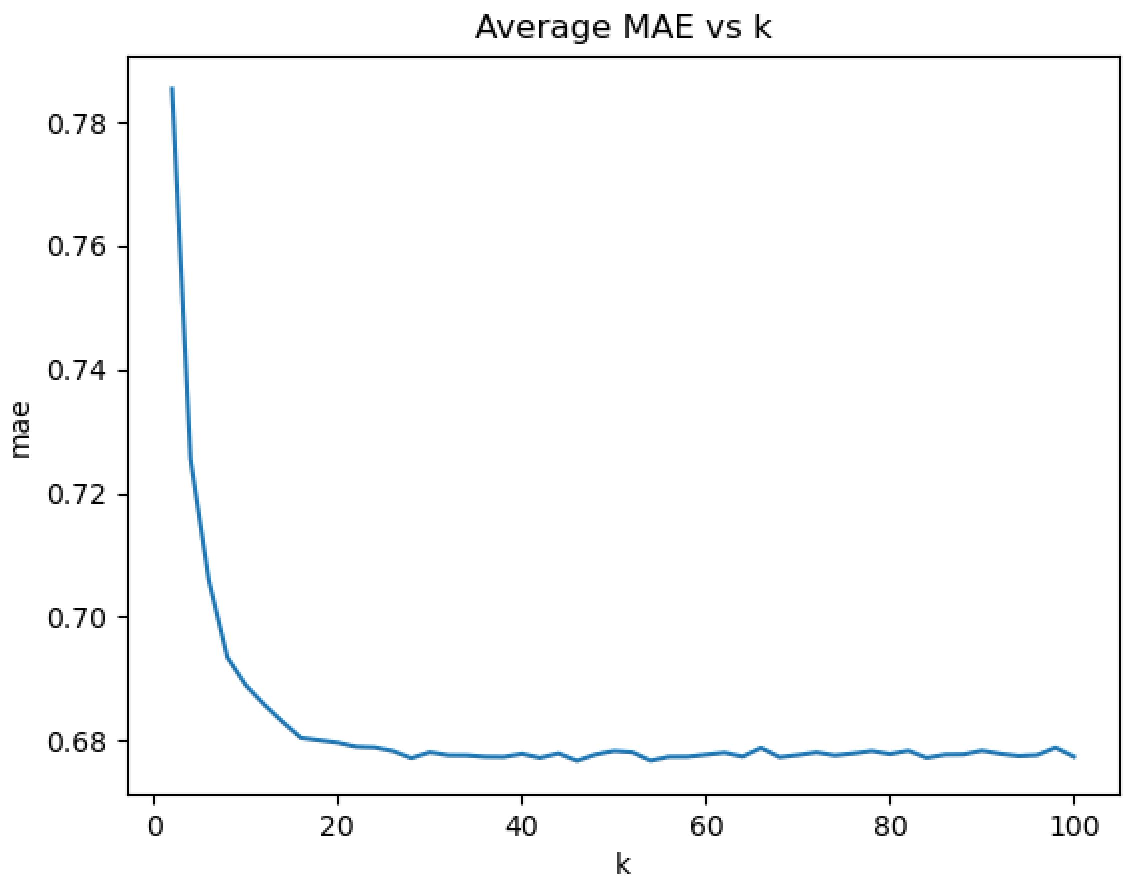
Mean centering allows us to reduce the bias in our system and to account for users that have ratings which are too disparate from the average of the other users. Consider a user who has rated all movies with the highest score. If we use these ratings as they are, then they will introduce bias to our dataset which will make it more difficult to have a model that can generalize well. The same can also be said to users that have rated all movies very poorly. By mean centering, we reduce the effect such users will have on the data and reduce the bias in our dataset.

Question 4

After designing a KNN that uses the Pearson correlation coefficient as the distance measure and evaluating the KNN with different K values as specified, we get the following plots:

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839





Question 5

By examining the figures, we can determine that both the average RMSE and the average MAE reach their steady state values around when k=20. For this k value, the average rmse is **0.8911** and the average mae is **0.6796**.

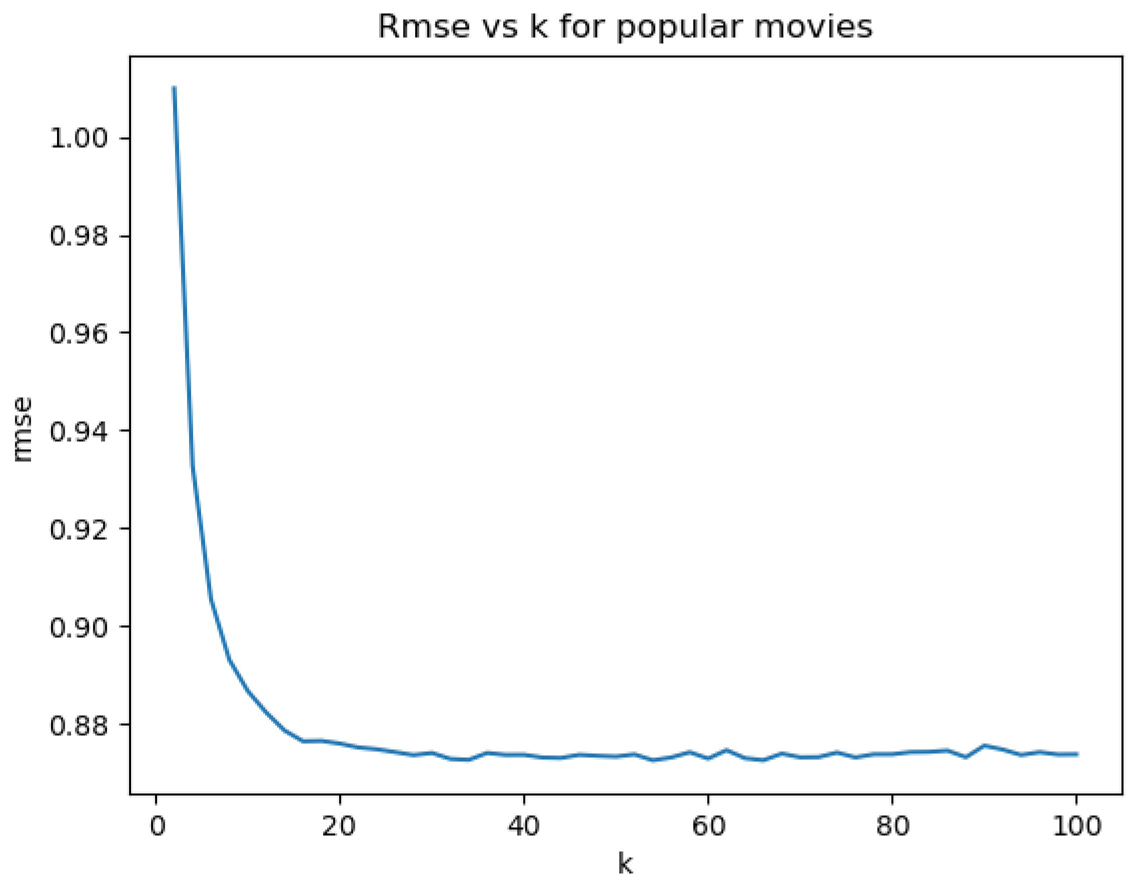
Question 6

Now, let us examine the performance of the KNN collaborative filtering when the test set is trimmed with respect to certain conditions.

Popular Movie Trimming

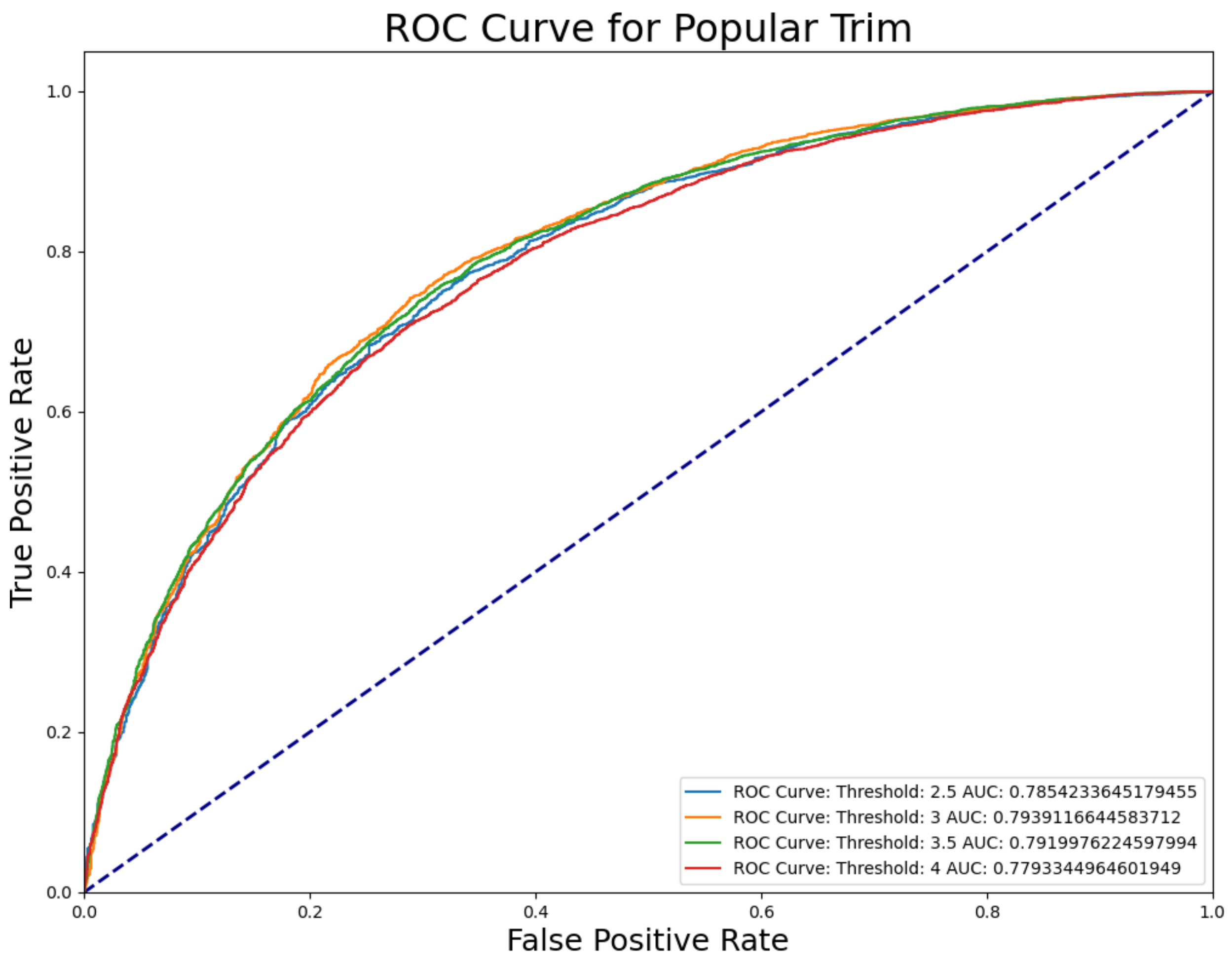
For this trimming, we only consider data entries in the test set that have received more than 2 ratings. The progression of the average RMSE with respect to different k values is as follows:

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



The minimum RMSE is observed to be **0.87258** when k is **54**.

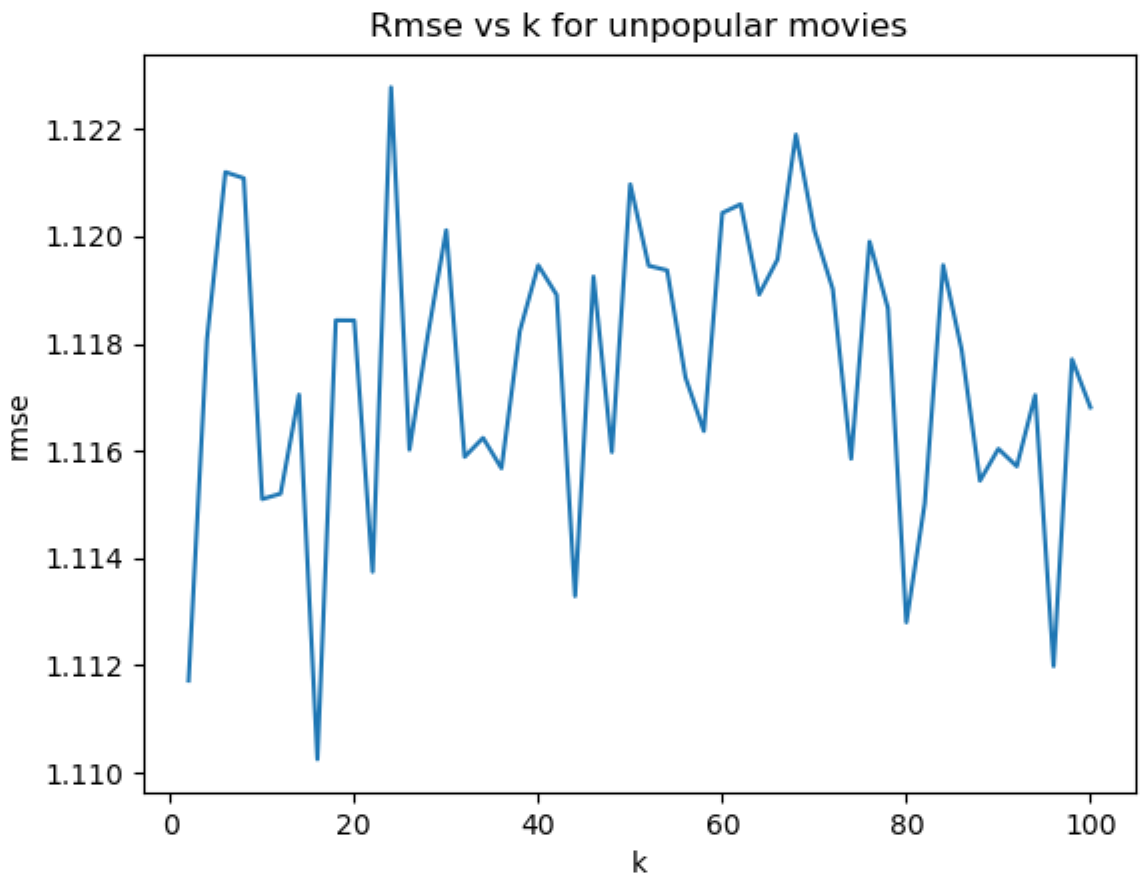
For this optimal k value, let us plot the ROC curve for different thresholds:



As can be seen, the highest area under the curve (AUC) is given when the threshold is 3.

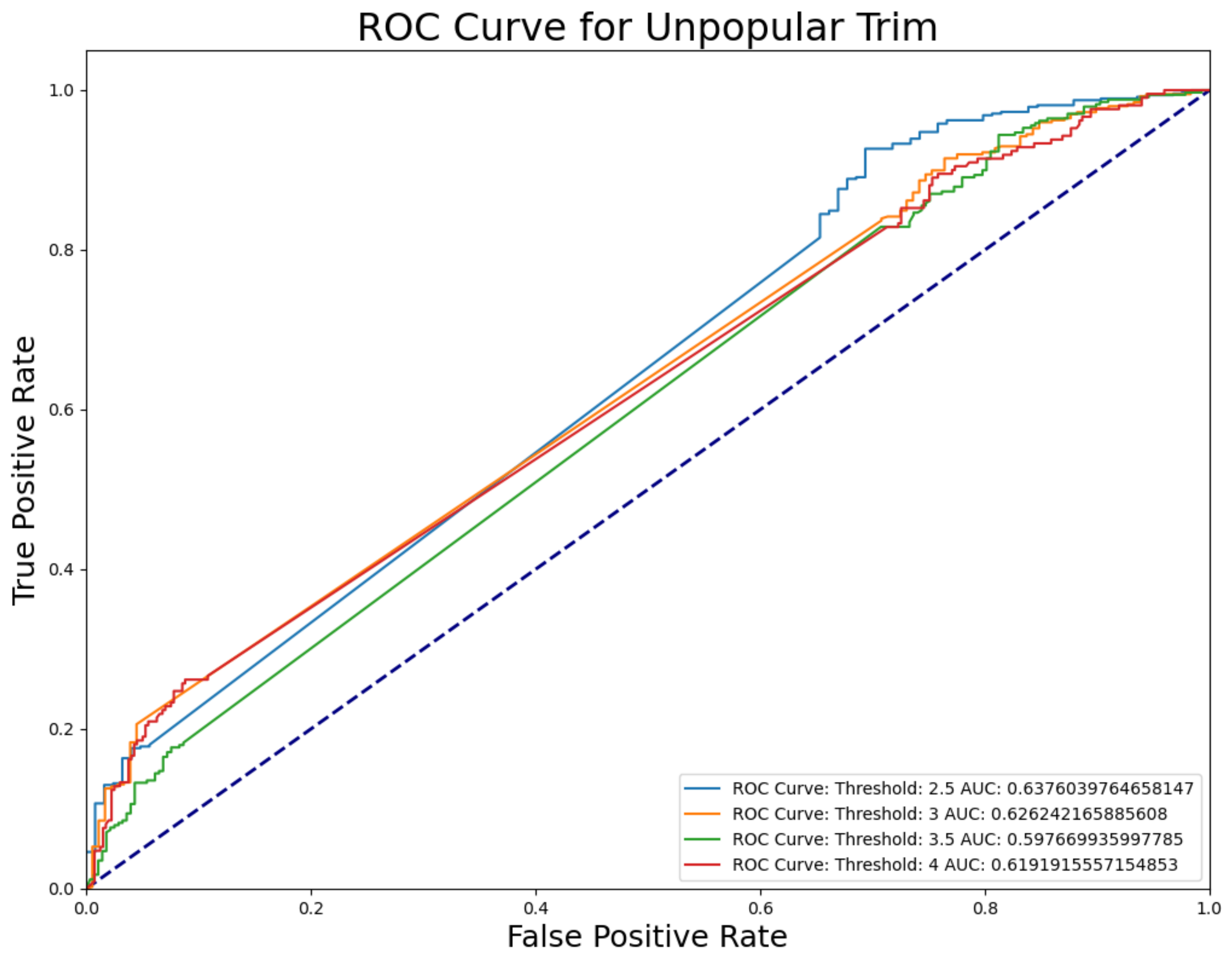
Unpopular Movie Trimming

For this part, we only consider the movies in the test dataset that have received less than or equal to 2 ratings. The progression of the average RMSE values with respect to k for this trimming is as follows:



As can be seen, the monotonically decreasing trend observed in the popular movie trimming is not present here and the effect of k on rmse appears to be more random. The minimum RMSE for this setup has been observed to be **1.11024** when the k value is 16.

We can plot the ROC cure for when k=16 with different thresholds as follows:

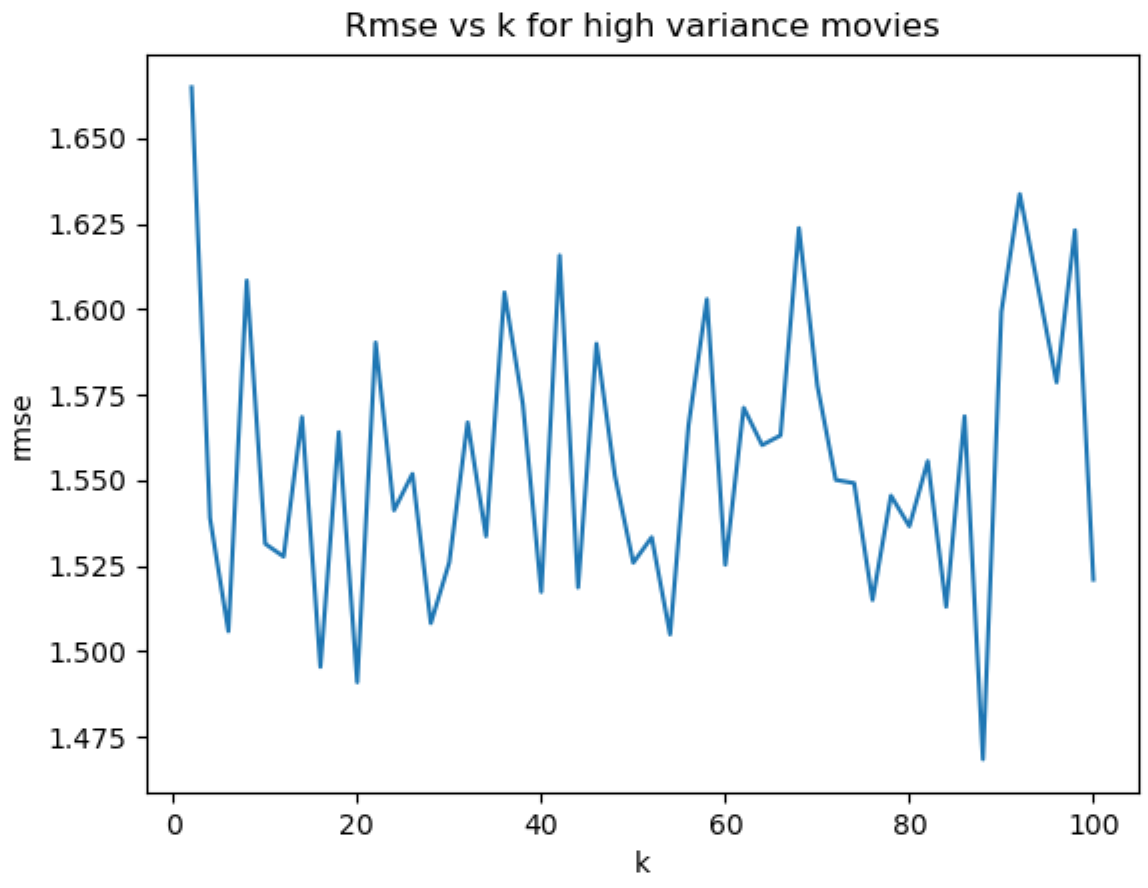


As can be seen the best AUC is obtained when the threshold is set to 2.5.

High Variance Trim

In the final trimming configuration, we only consider the movies in the test set that have a variance of at least 2 and that have received at least 5 ratings. The progression of the average RMSE values with respect to k for this trimming is as follows:

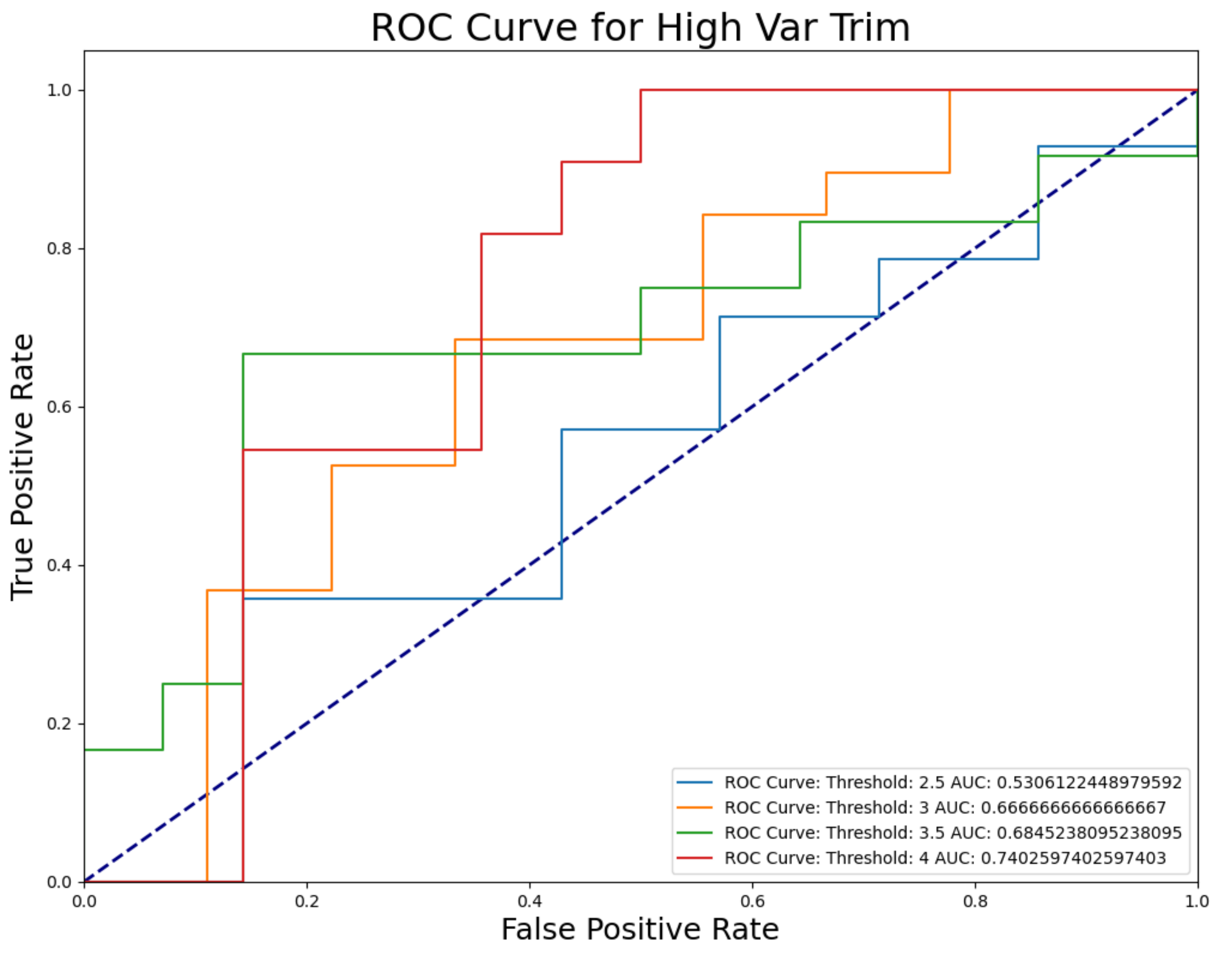
Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



As with the unpopular movie trimming, the effect of changing k on the average RMSE appears to be much more random. For this configuratioın, the minimum average RMSE is found to be **1.46846** when k is equal to 88.

The ROC curve for this trim when k is 88 for different thresholds can be plotted as follows:

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



As can be seen, the highest AUC is obtained when the threshold is set to 4.

PART 5 Model-based Collaborative Filtering

In this configuration of collaborative filtering, we make use of machine learning models to predict users' ratings.

Question 7

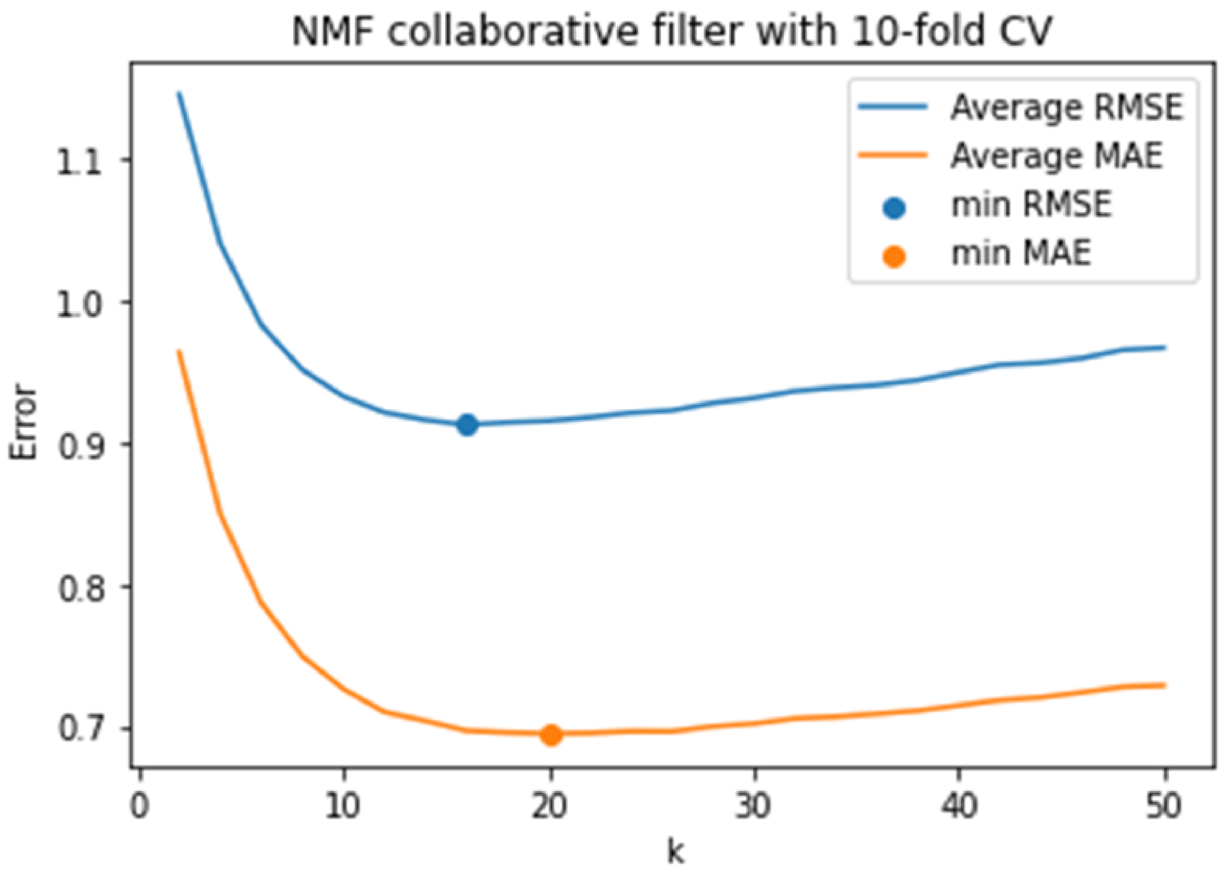
Due to the presence of the W_{ij} that is used to take only the ratings that are known into account, the given optimization problem is non-convex. The least-squares formulation of the given problem for a fixed U can be given as:

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

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839

Question 8

A) After designing the NMF-based collaborative filter, we can determine the best number of latent factors by plotting the change in RMSE and MAE with respect to the amount of latent factors and picking the number that gives the best results. The plots for these two error metrics are found to be as follows:



B)

For RMSE, we can examine the plot to determine that the lowest RMSE is obtained when the number of latent factors (k) is **16**. For this k value, the RMSE value is found to be **0.9125122115921815**.

For MAE, the best results are obtained when k is equal to **20.** For this k value, the MAE is observed to be **0.6948483666520986**.

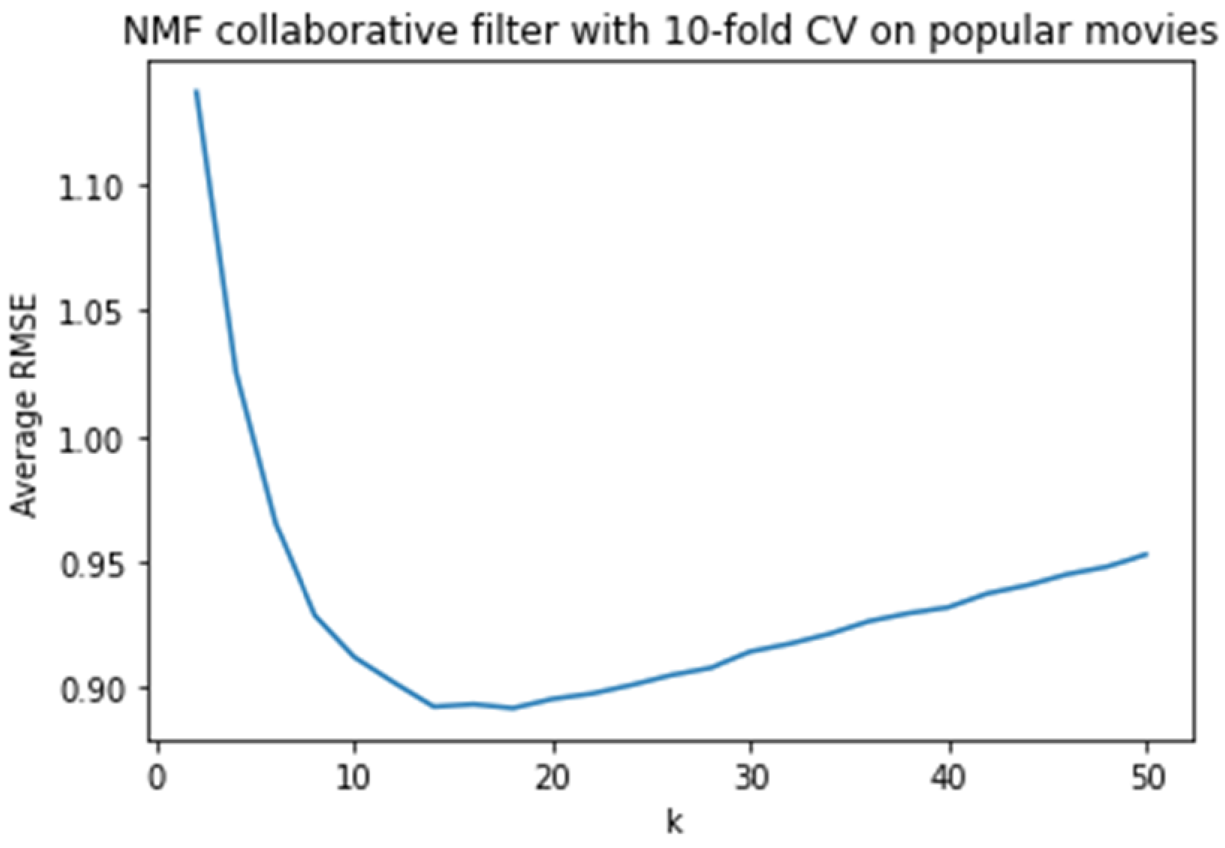
By examining the results for RMSE and MAE, we can determine that the best results are obtained when the amount of latent factors is somewhere between 16 and 20. In our dataset: there are a total of 20 genres. These are: no genres listed, Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, IMAX, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western. As there are a total of 20 genres, then we can state that the optimum number of latent factors is close to the number of distinct genres.

C)

As we have done for KNN filtering, let us examine the performance of the NMF collaborative filtering when the test set is trimmed based on different conditions.

Popular Trimming

The change in average RMSE for NMF collaborative filtering as the amount of latent factors change when tested on popular movies in the test set is as follows:

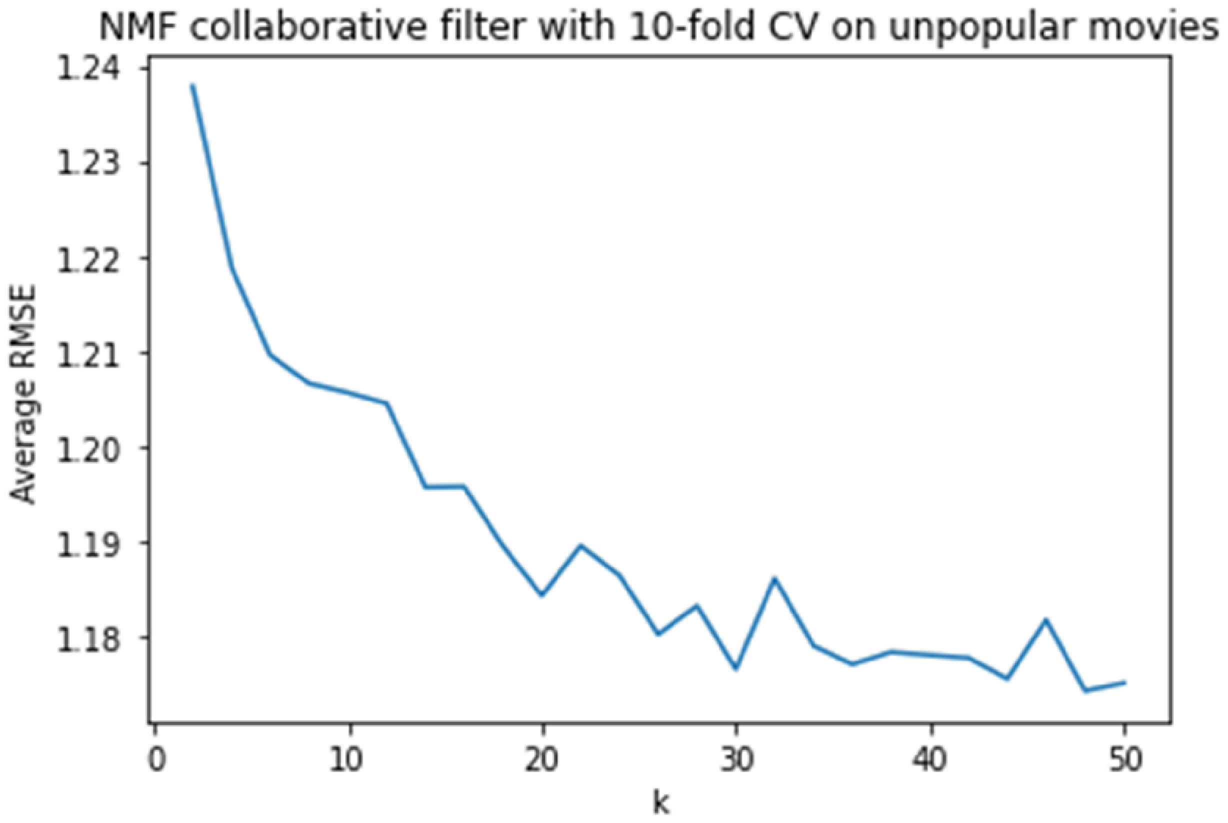


For this configuration, the minimum RMSE is found to be **0.8917780281064542**.

Unpopular Trimming

The change in average RMSE for NMF collaborative filtering as the amount of latent factors change when tested on unpopular movies in the test set is as follows:

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839

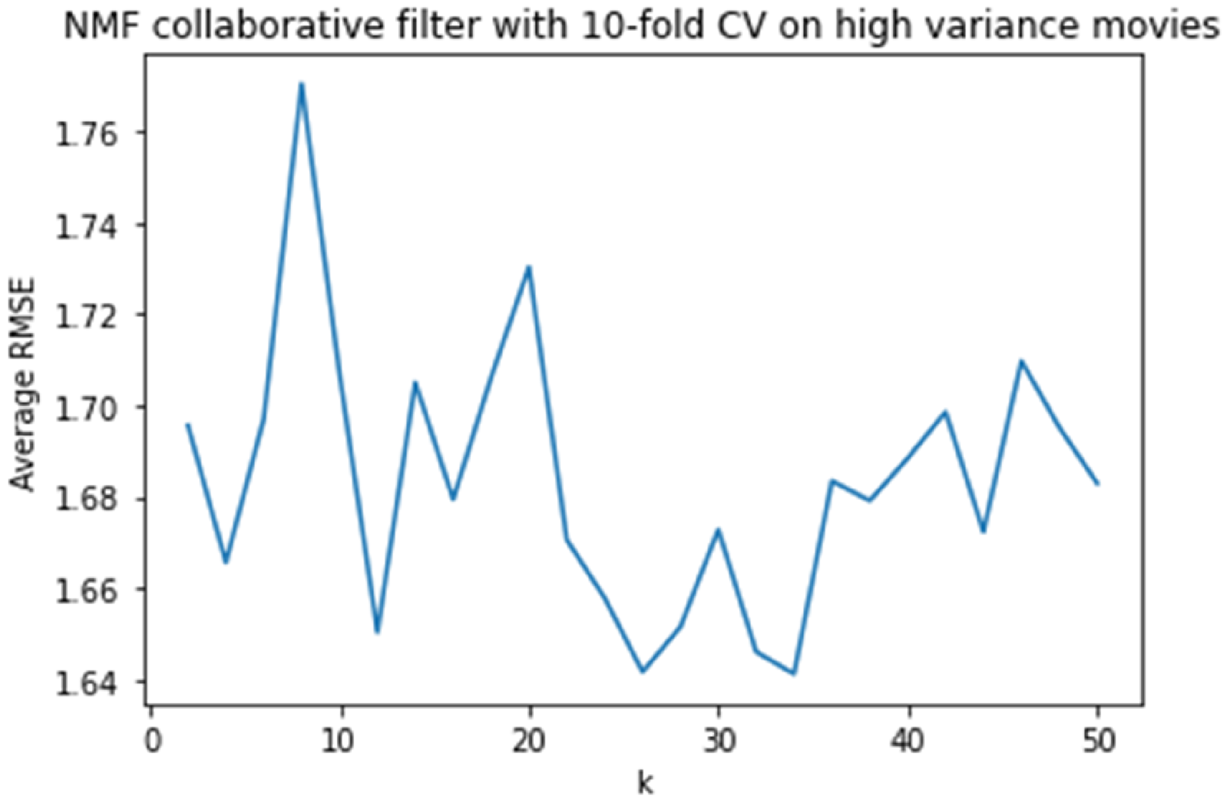


For this configuration, the minimum RMSE is found to be **1.1742788574599023.**

High Variance Trimming

The change in average RMSE for NMF collaborative filtering as the amount of latent factors change when tested on movies that have been determined to have high variance in the test set is as follows:

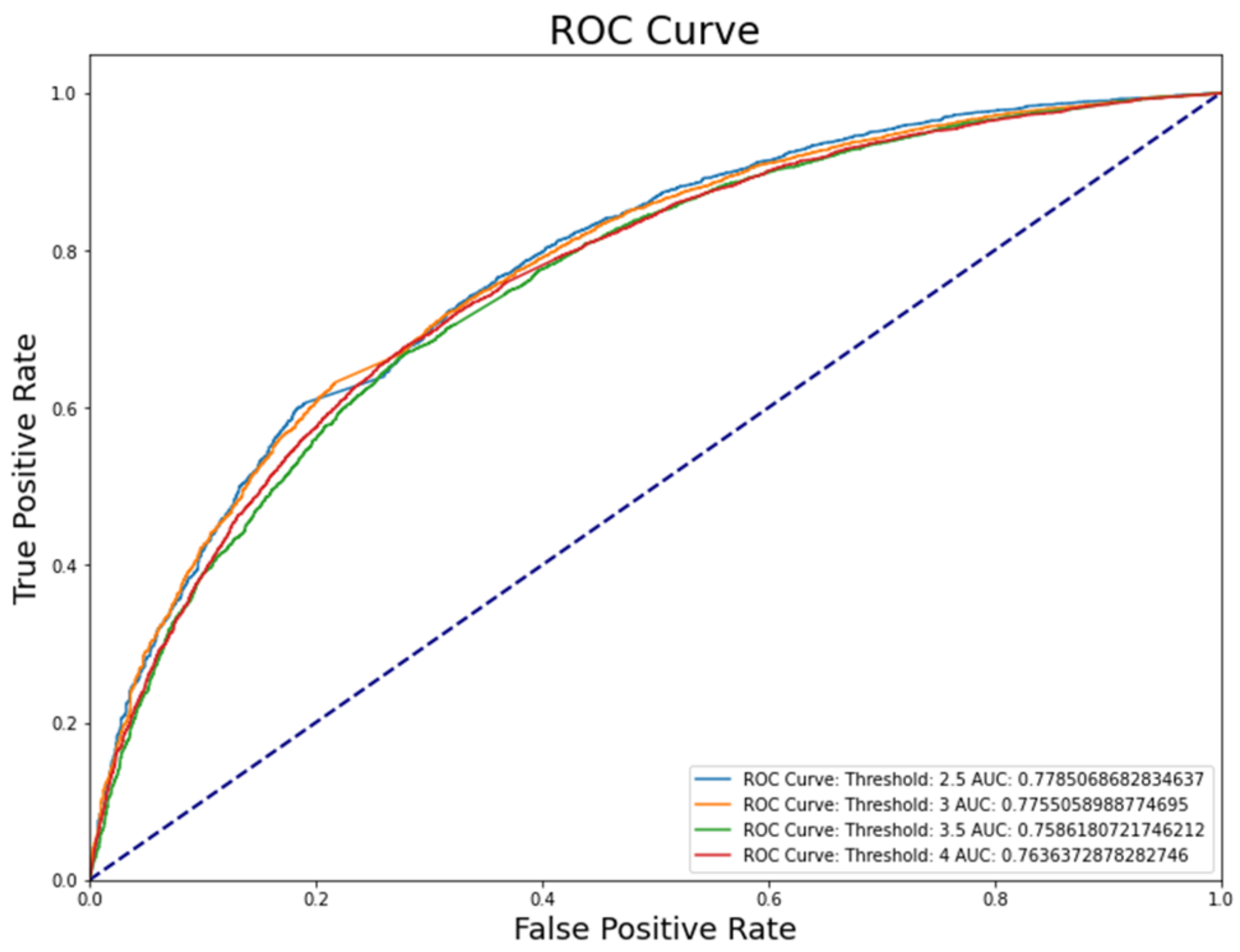
Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



For this configuration, the minimum RMSE is found to be **1.6412497999450313.**

We can evaluate the performance of the NMF filter when no trimming is performed on the test set by utilizing ROC curves. Here as we did for the previous section of the homework, we plotted distinct ROC curves based on the threshold applied to the results and noted the AUC values. For this experiment, we set the amount of latent variables to be 20.

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



As can be seen, the best results are obtained when the threshold is set to 2.5.

Question 9

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839

col = 0	col = 5	col = 6	col = 15	col = 20
Crime|Drama|Thriller	Comedy|Drama|Fantasy	Documentary	Action|Drama	Comedy|Western
Comedy|Drama|Romance	Comedy	Drama	Comedy|Drama	Horror|Mystery|Sci-Fi|Thriller
Drama	Drama	Comedy	Action|Drama|Romance|War	Comedy|Drama|Romance
Drama	Action|Comedy	Action|Horror|Thriller	Drama	Comedy
Comedy|Drama|Romance	Musical	Drama	Comedy	Action|Sci-Fi|Thriller
Comedy|Fantasy|Horror|Musical|Thriller	Drama|Musical	Drama|Film-Noir|Mystery|Romance	Animation|Comedy	Comedy|Romance
Adventure|Children|Comedy|Drama	Drama	Crime|Horror|Sci-Fi	Animation|Children|Fantasy|Musical	Action|Drama
Comedy	Romance	Comedy|Romance	Comedy|Romance	Action|Adventure|Thriller
Drama	Documentary	Drama	Drama	Comedy|War
Drama|Romance	Action|Comedy|Crime	Comedy	Animation|Comedy|Drama	Comedy|Fantasy

The table above gives us information about the genres of the top 10 movies for the different columns of the matrix V. From the table, we can determine that the top 10 movies belong to a small collection of genres. Furthermore, it seems like each latent factor is responsible for a different grouping of genres. Because each movie belongs to multiple genres it is difficult to reduce each latent factor to a single genre, thus we see a clustering effect in the top movies for each column of V. It does seem that each column has a unique combination of these genres however, meaning that they are each responsible for a different direction or clustering. Moreover, it appears that as the number of latent factors increases, the number of distinct movie genres decreases.

Matrix Factorization with Bias

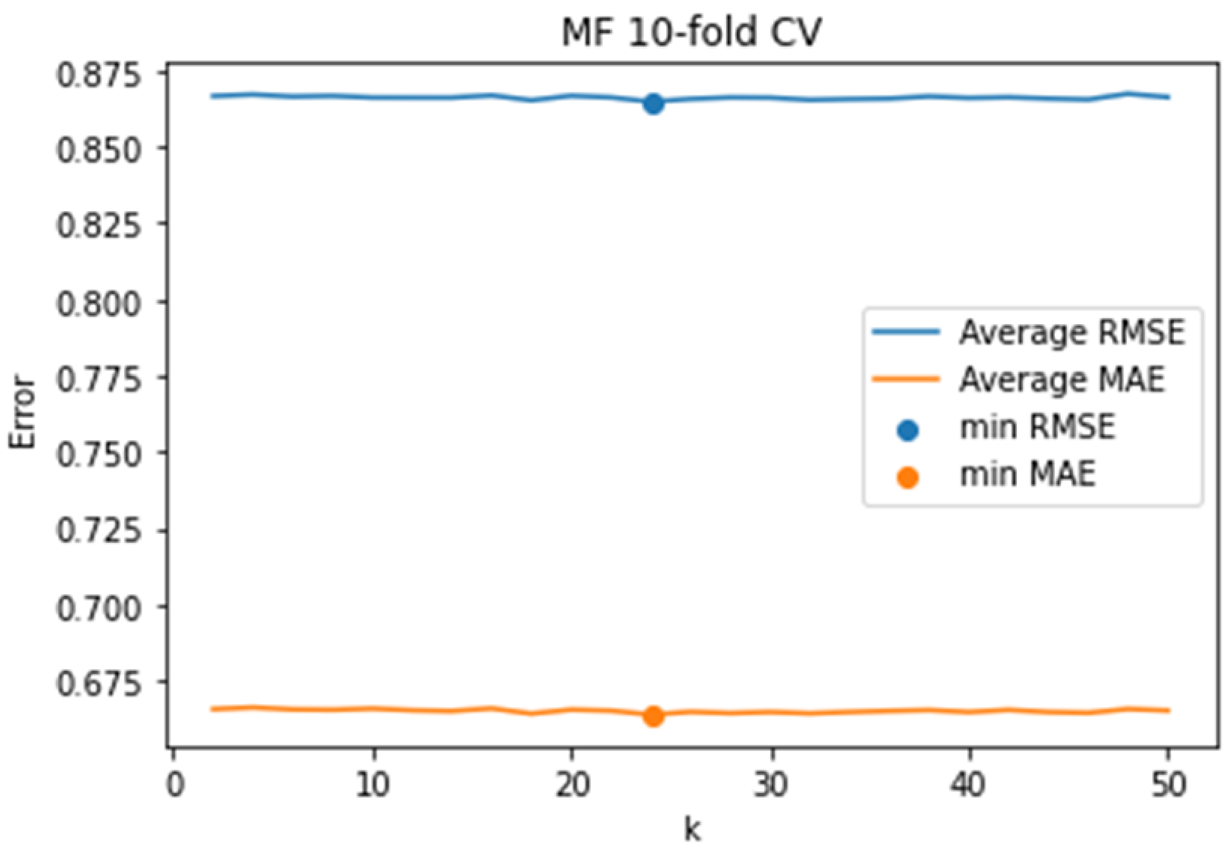
Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839

In this part of the project, we modify the cost function used for learning the U and V matrices for NMF and add a bias term.

Question 10

As we have done with NMF collaborative filtering, we can design a MF collaborative filter and see the performance on the original dataset using cross validation. We can also change the number of latent factors and see change in RMSE and MAE to determine the optimum number of latent factors. The plot for the change in the aforementioned loss functions with respect to the number of latent vectors is as follows:

A)



B)

For both the RMSE and MAE, we can observe that the best results are obtained when the number of latent factors (k) is equal to **24**. For this value, the RMSE is found to be **0.8649781438299895** and the MAE is observed to be **0.6642428382904689.**

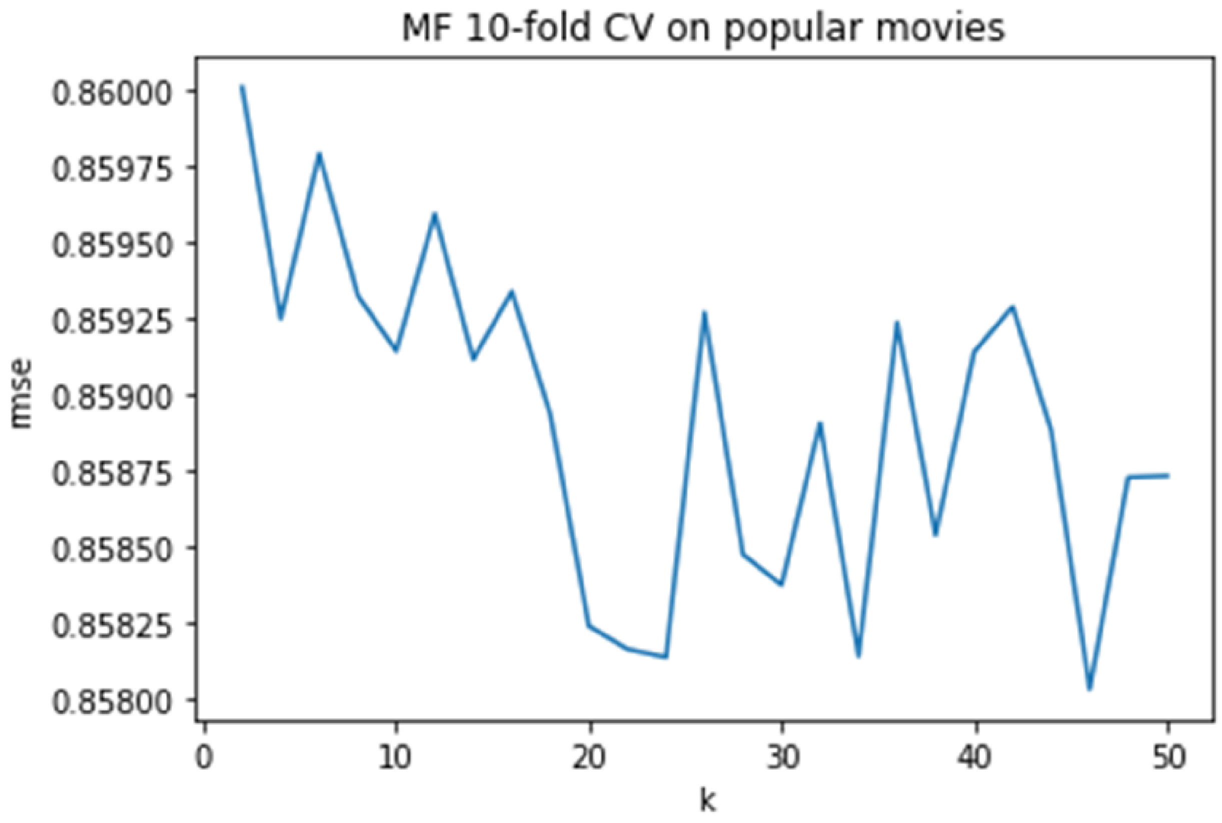
As the amount of distinct genres has been previously found to be 20, we can state that for MF-based collaborative filtering, the best value for the number of latent factors is slightly higher than the number of genres.

C)

Now let us examine the performance of the MF collaborative filtering when the test set is trimmed with different configurations.

Popular Trimming

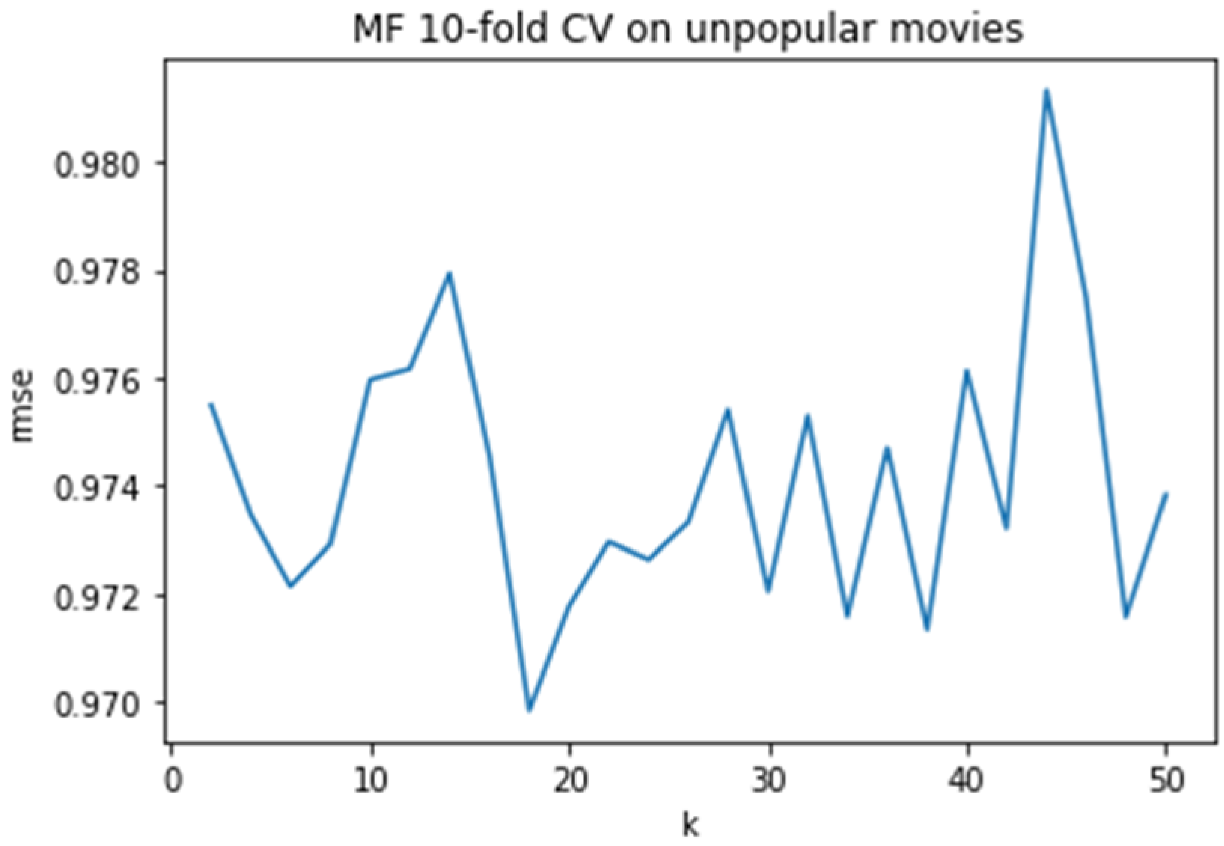
The change in average RMSE for MF collaborative filtering as the amount of latent factors change when tested on popular movies in the test set is as follows:



The minimum average RMSE here is found to be **0.8580332570722489**.

Unpopular Trimming

The change in average RMSE for MF collaborative filtering as the amount of latent factors change when tested on unpopular movies in the test set is as follows:

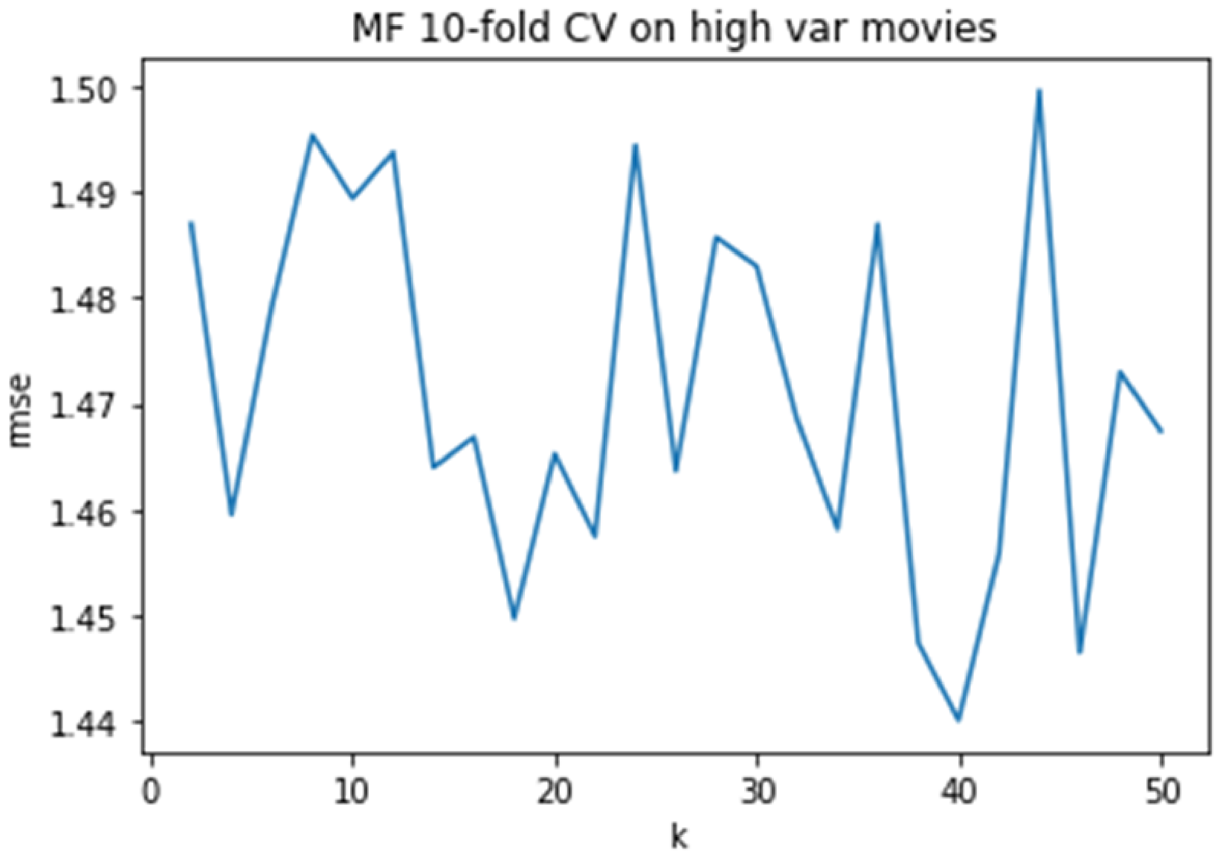


The minimum average RMSE for this setup is found to be **0.9698352410400902.**

High Variance Trimming

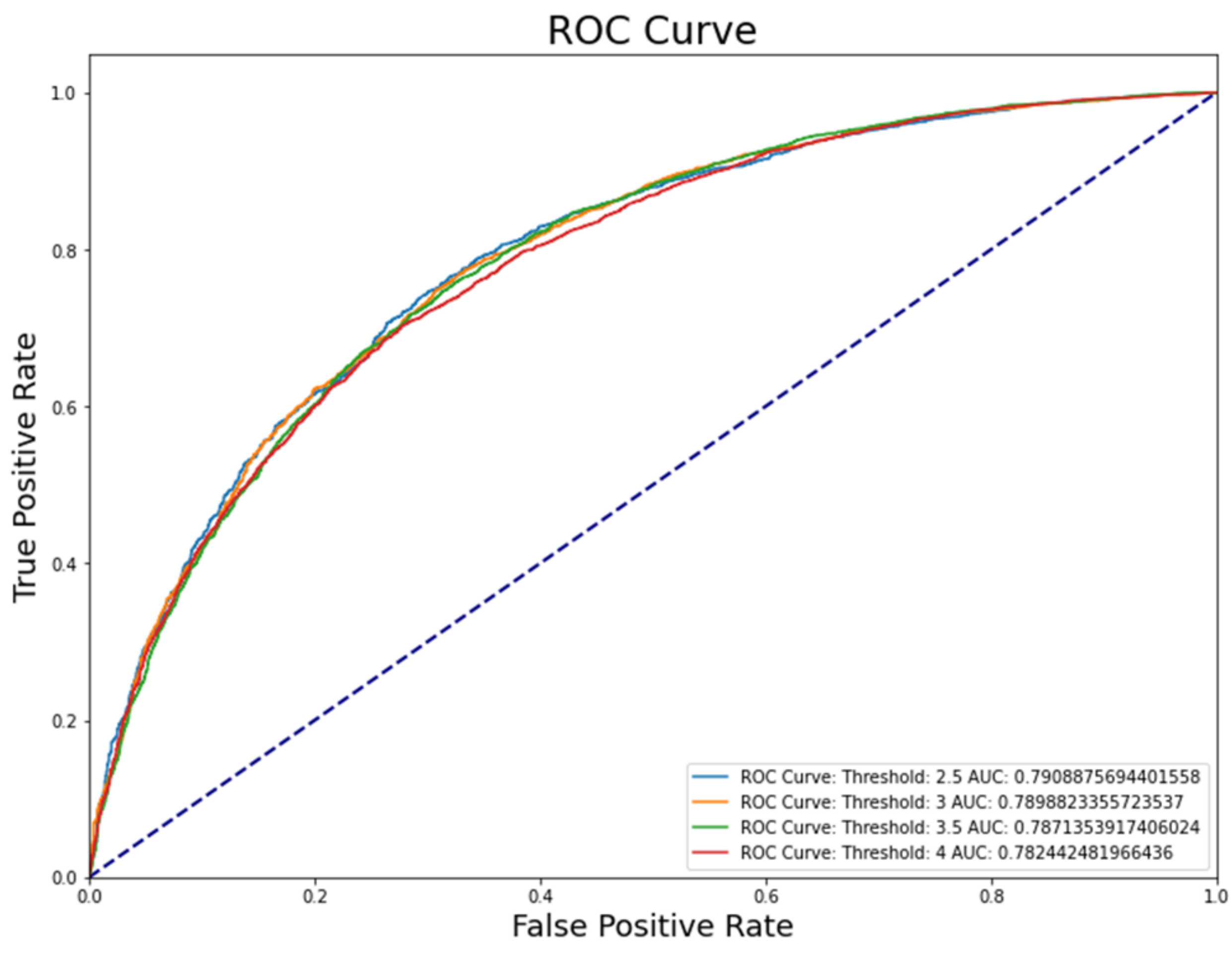
The change in average RMSE for MF collaborative filtering as the amount of latent factors change when tested on movies that have been determined to have high variance in the test set is as follows:

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



The minimum average RMSE for this setup is found to be **1.4400669483486412.**

By plotting the ROC curves with different thresholds, we can examine the performance of MF-based collaborative filtering when no trimming is applied to the test set. As before, to generate the ROC curves we used the thresholds specified in the project description and noted the AUC values. For this part of the project, the amount of latent variables was taken to be 24. The resulting ROC curves are as follows:



As can be seen, the best results are obtained when the threshold is set to 2.5.

PART 6 Naive Collaborative Filtering

For this part of the project, instead of relying on a type of model, we will be predicting the rating given to a movie by a user using the mean rating of that user. As can be understood, this is really a basic approach to the problem at hand.

To create our model, we just take the average rating of all the users in the dataset using the ratings they have provided. Then, when testing, we assign the calculator mean of user u as the rating user u would have given to a movie m.

Question 11

We can evaluate the performance of our naive littering by performing k-fold cross validation and calculating the average RMSE across all the folds. If we do so, the average RMSE value is determined to be: **0.934696.**

Now as we have done before, let us trim our testset based on some criteria and see how the performance changes.

Popular Trimming

The average RMSE across all folds if we just include all the movies that have received more than 2 ratings when we utilize naive filtering is **0.9323211928420088.**

Unpopular Trimming

The average RMSE across all folds if we included the movies that have received less than or equal to 2 ratings when we use naive filtering is **0.9711977610065399.**

High Variance Trimming

If we trim the test set so that only movies that have received more than 5 ratings and have a variance of at least 2 for each fold in our k-fold cross validation, we get an average RMSE of **1.4743610761361792** with naive filtering.

PART 7 Performance Comparison

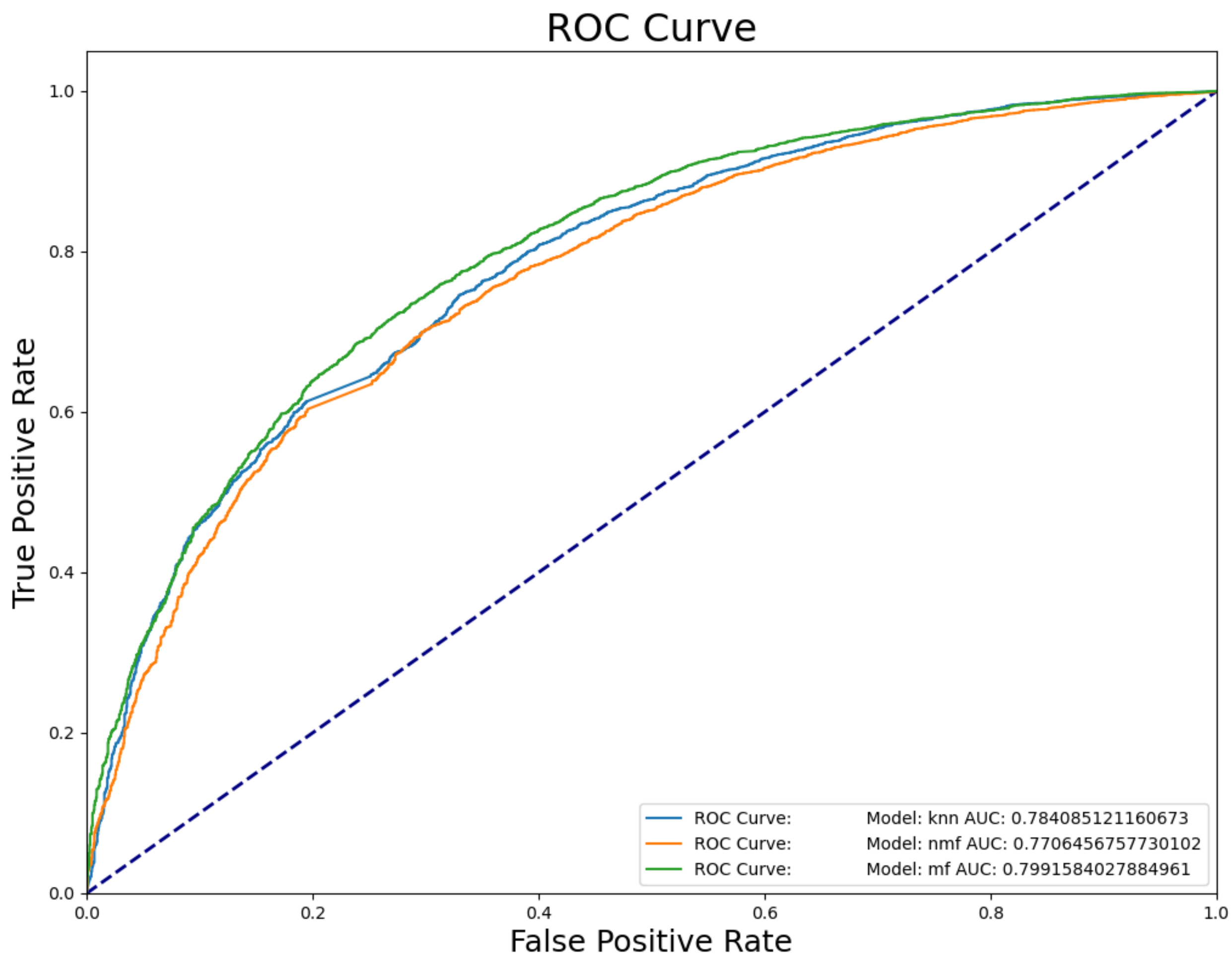
Question 12

For this part of the project, we compare the performance of the different models that we have utilized so far, namely the KNN, NMF and MF with bias by plotting the ROC curves for each model.

To perform the comparison, we train a KNN model with k equal to 20, an NMF model with the number of latent factors equal to 18 and an MF with bias model with the number of latent factors equal to 24. These hyperparameters have been determined by the previous results obtained during the implementation of the project.

To plot the ROC curve, we set the threshold to 3 as instructed, so that any rating that has a value more than or equal to 3 is assigned as 1. The rest is assigned to class 0.

The ROC curves for the three models are as follows:



From the figure, we can see that the MF with bias has the best performance when the threshold is 3, as it has the highest AUC value. It is followed by KNN and finally the NMF models in terms of performance.

PART 8 Ranking

Question 13

Precision gives us the fraction of the items an user liked out of all the items that were recommended to him/her. It can be thought as out of all the items we recommended, how many of them the user actually liked.

On the other hand, recall is the fraction of the recommended items out of all the items that the user liked. It can be thought as out of all the items the user liked, how many of them were recommended to the user.

Question 14

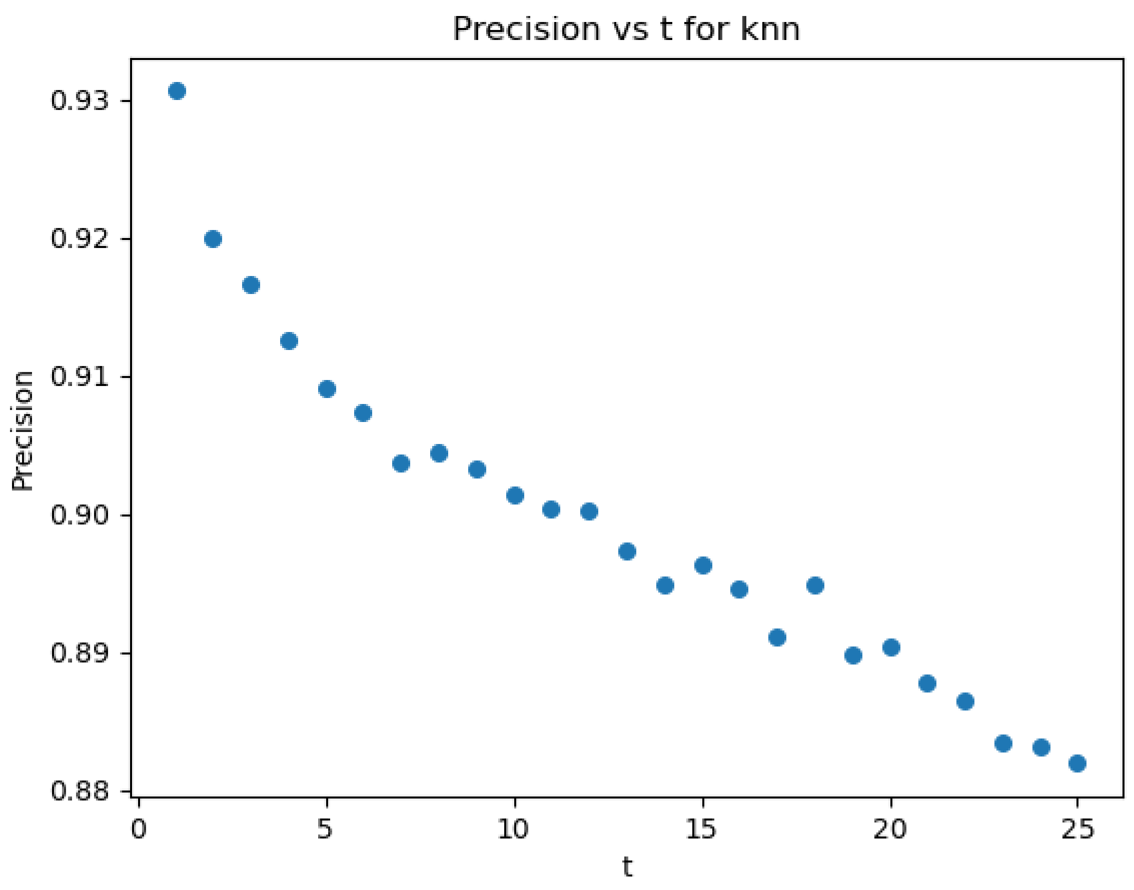
Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839

For this part of the project, we were tasked with examining how the precision and recall of each of our previous approaches (KNN, NMF, MF with bias) change as the amount of recommended items changes. Here, we performed cross validation for each model and for each t value. The t values ranged from 1 to 25.

To create our models, we used the set of hyperparameters that were previously found to be the best. For KNN, the k value was set to 20; for NMF, the number of latent factors was set to 18 and for MF with bias, the number of latent factors was set to 24.

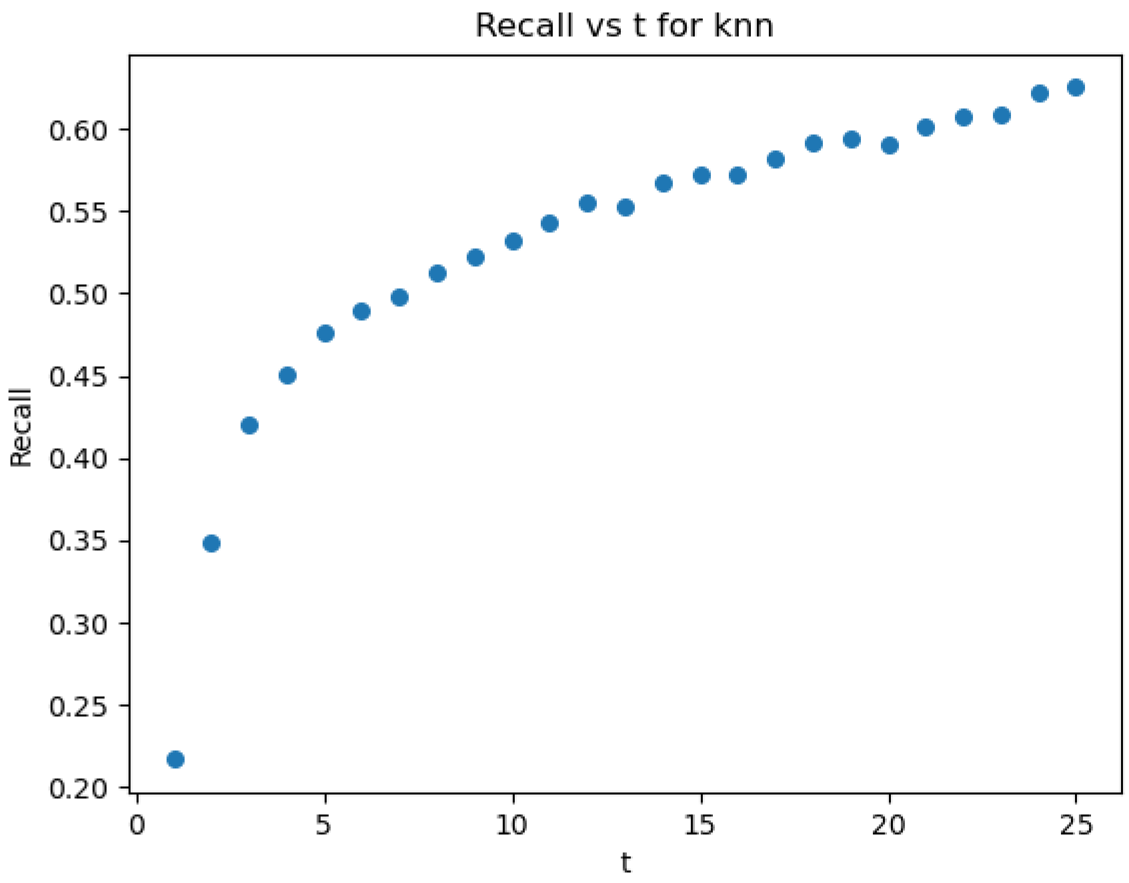
The results are as follows:

KNN

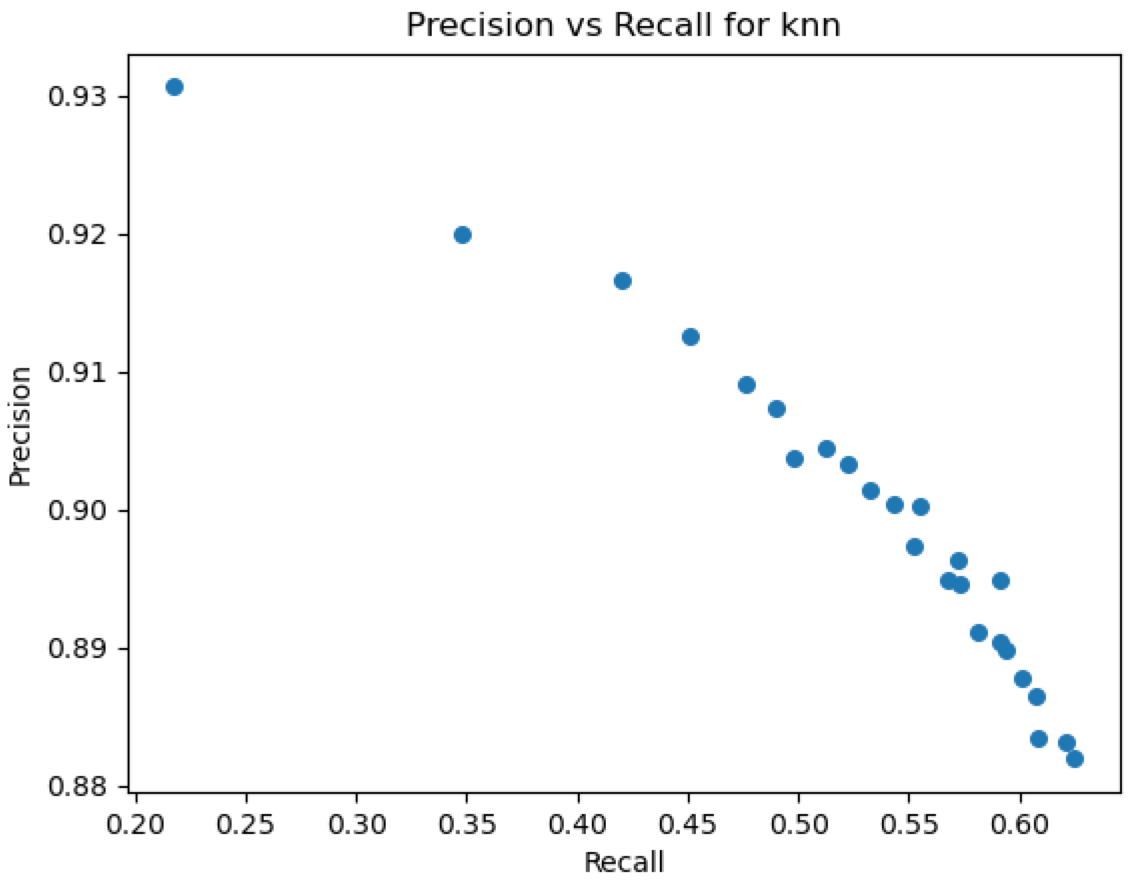


As can be seen, as the t value increases, the precision value decreases. This is expected since the more movies we recommend from the ones we predicted to have high ratings, the more likely it is that the user was not interested in those movies. For example, if we recommend all the possible movies, then our precision would be low as the user would not be interested in most of the movies we recommended.

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



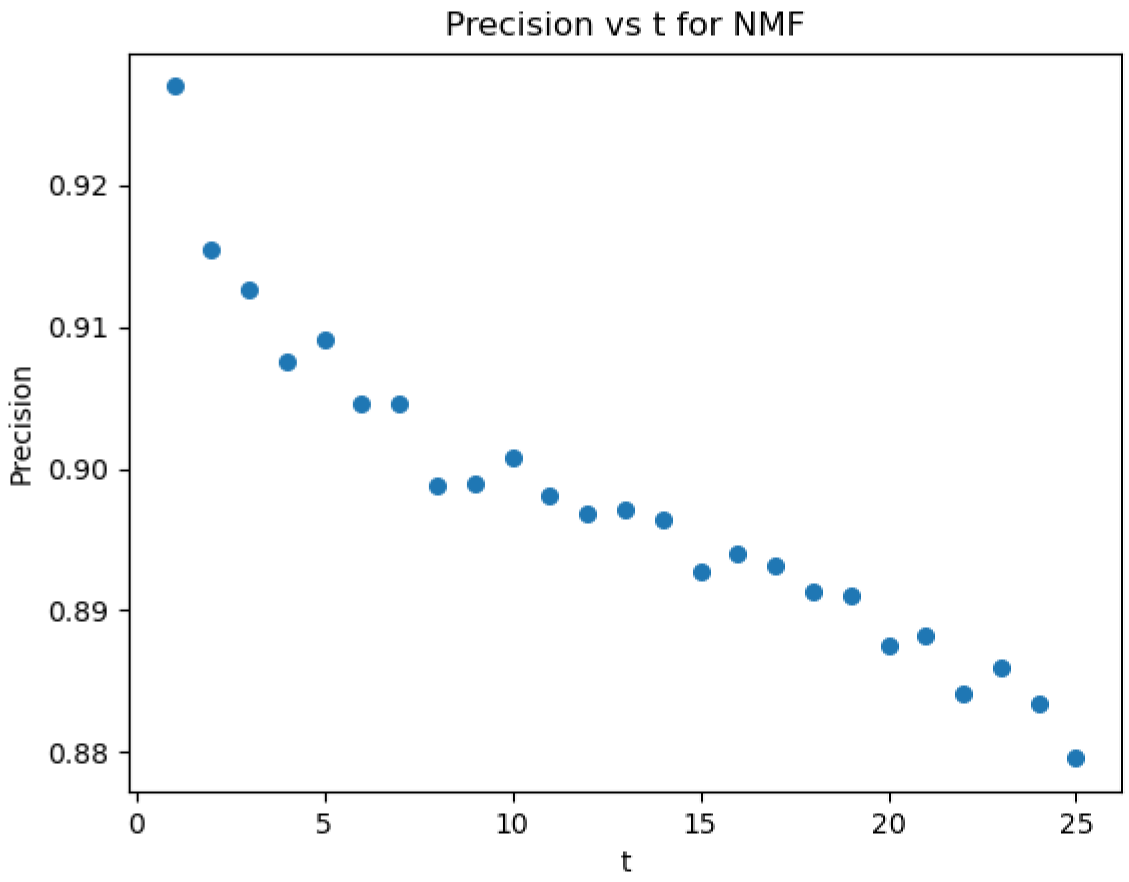
Inverse to the relationship between precision and t, we see that as the t value increases, the recall value increases. This is also expected, because as we recommend more movies, it is more likely that we recommend the movies that the user liked. For example, if we recommended all the possible movies, then our recall would be 1 since all the movies that the user liked would be recommended.



As can be seen, there is an inverse relationship between precision and recall. They can not be both increased at the same time, indicating that we need to choose what we want to optimize for our problem.

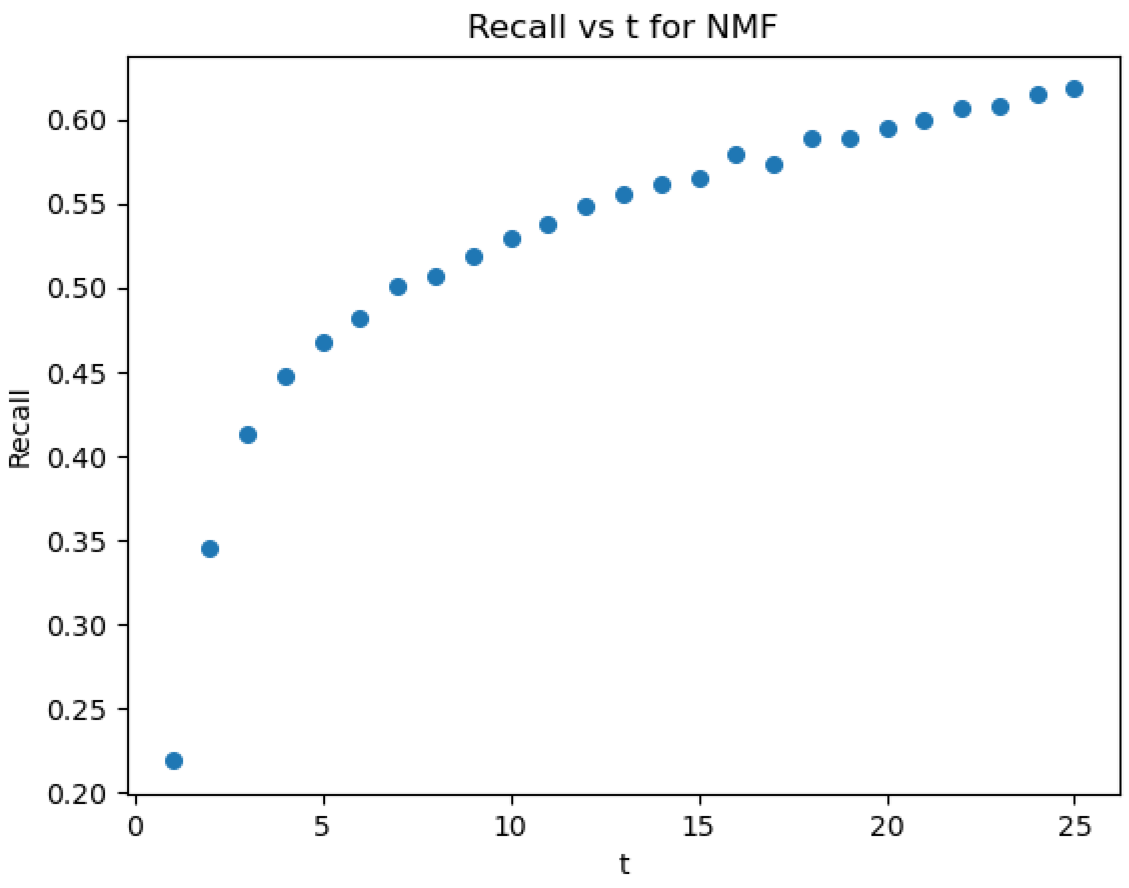
NMF

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839

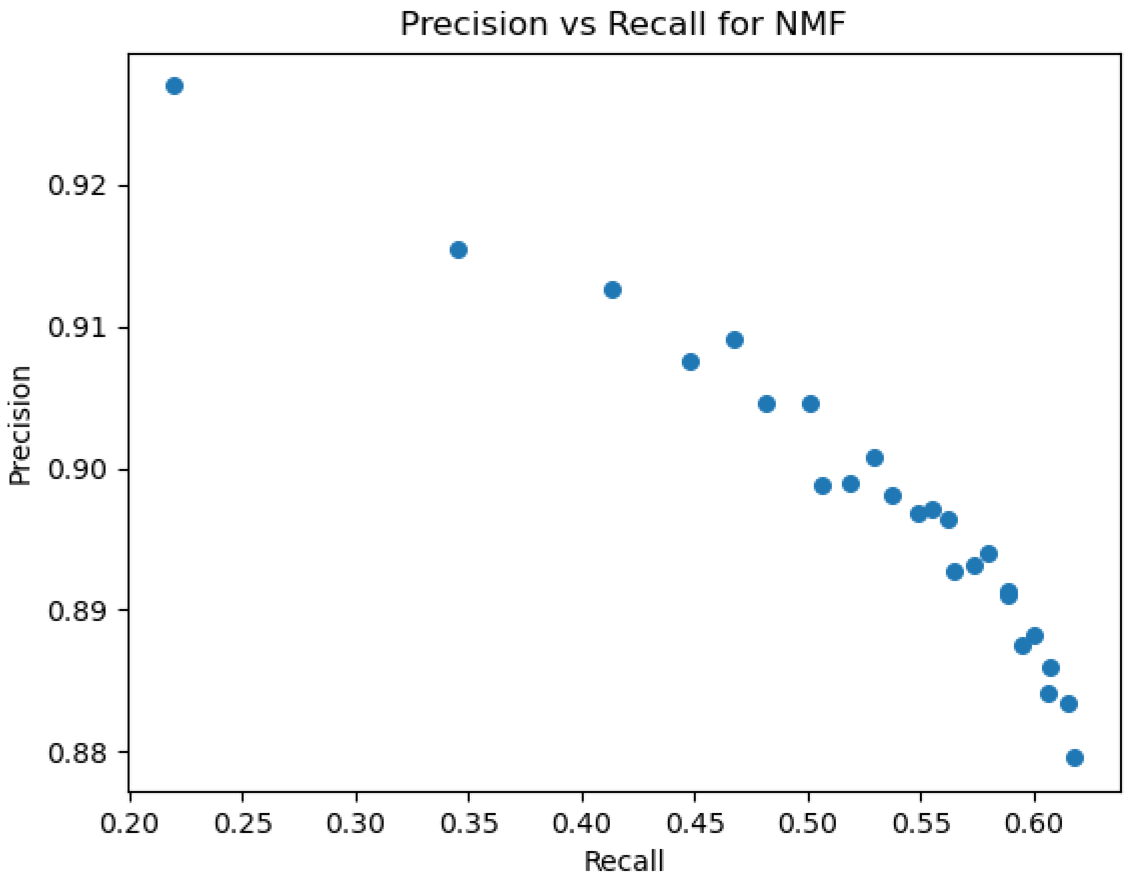


As with the previous precision vs t graph, as the t value increases, the precision value decreases.

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



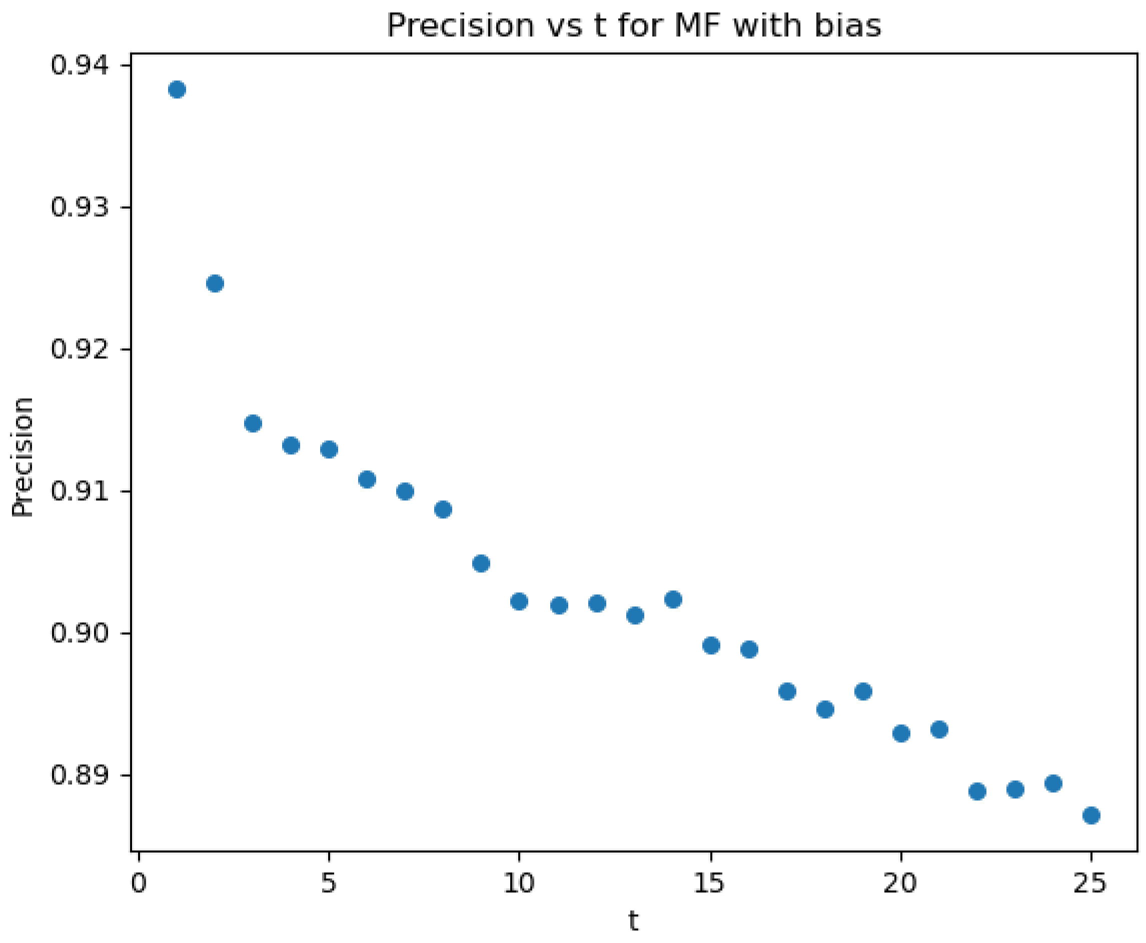
As with the previous recall vs t graph, as the t value increases, the recall value increases.



As can be seen, there is an inverse relationship between precision and recall. The tradeoff between precision and recall that was present for KNN is also present for NMF.

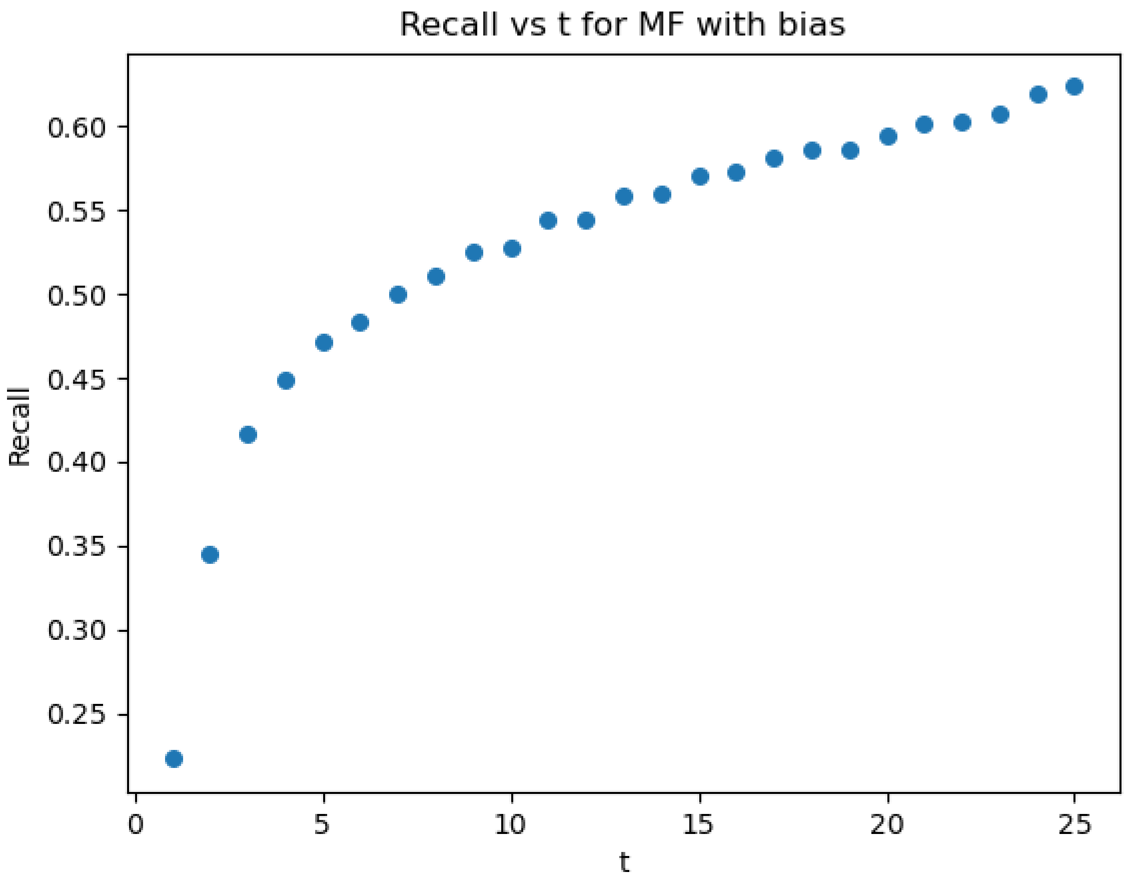
MF with bias

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



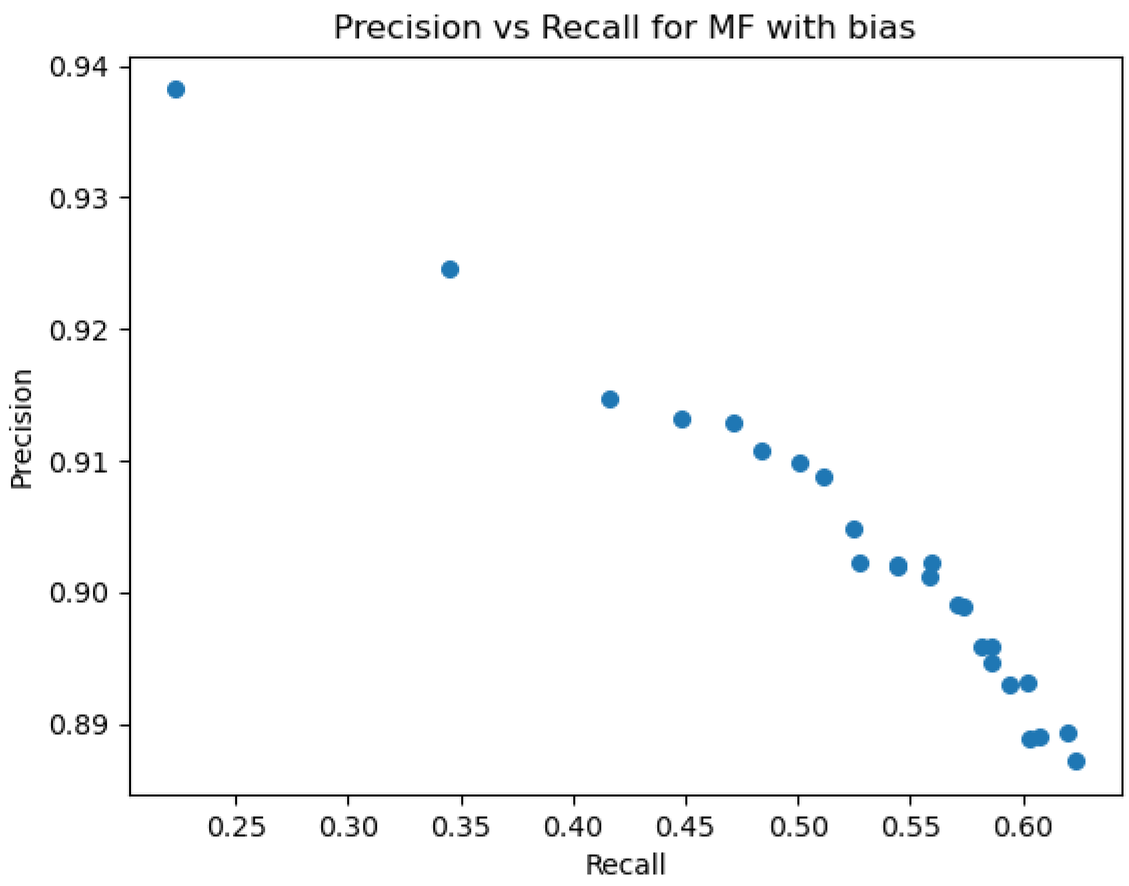
As before we can see that as t increases, the precision decreases as expected.

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



As before, there is a positive relationship between t and recall. As t increases, the value of recall also increases.

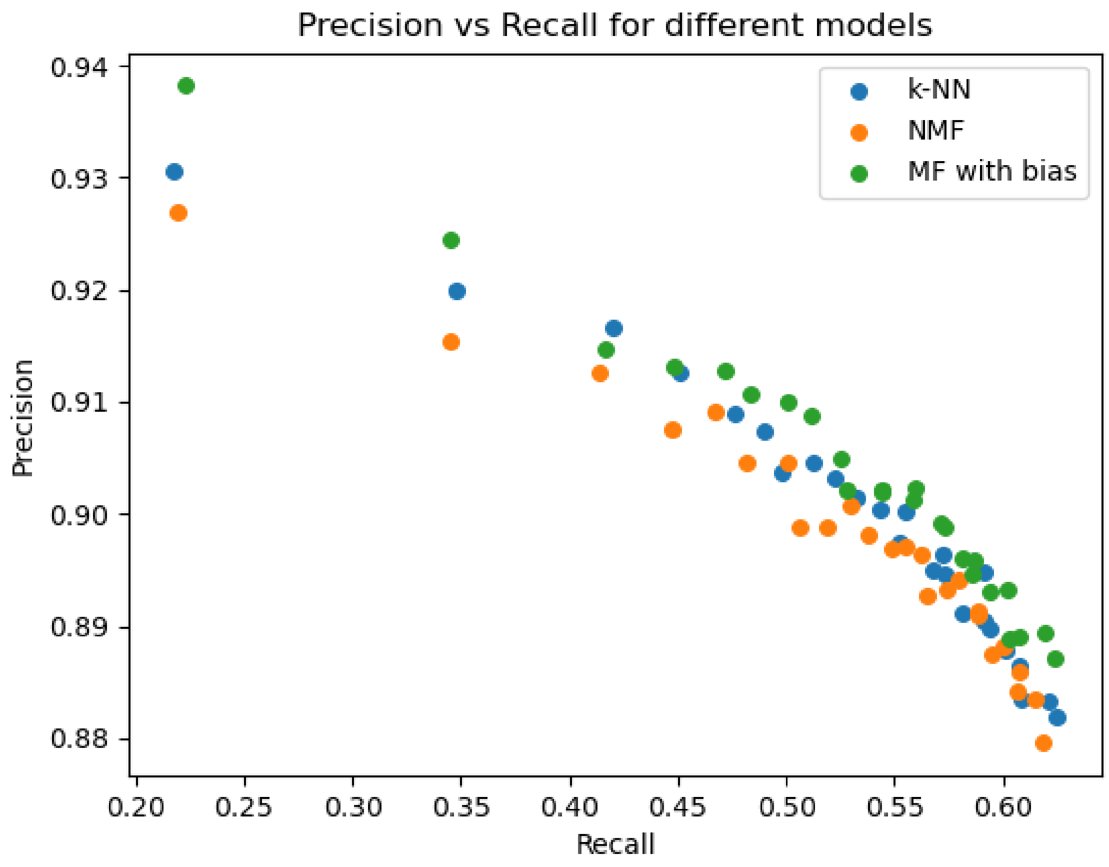
Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



As we observed for KNN and NMF, there is an inverse relation between precision and recall for MF with bias. We can not maximize one without decreasing the value of the other.

Now, let us draw the precision vs recall graphs for all the created models to see which model performs the best and gives the most relevant recommendations.

Deniz Orkun Eren – UID: 905624625
Matthew Waliman – UID: 605848839



By examining the figure, we can see that the inverse relationship between precision and recall is apparent in all three models. However, we can see that the best performance is obtained when we use the MF with bias approach. This is because the tradeoff between precision and recall is the least for this model. For a specific precision, it achieves the best recall and for a specific recall, it achieves the best precision. Therefore, we can conclude the most relevant suggestions are given when we use MF with bias.

In terms of performance, MF with bias is followed by k-NN and then NMF.