Large-Scale Data Mining: Models and Algorithms
ECE 219 Winter 2023
Prof. Vwani Roychowdhury
UCLA, Department of ECE

Project 3: Recommender Systems

Due Feb 26, 2023 by 11:59 PM

Yaman Yucel - 605704529
Ozgur Bora Gevrek - 505846360
Eray Eren - 006075032

Introduction:

Recommendation systems are used for giving movie recommendations to users. In this project, we have investigated the performance of collaborative filtering models. The main idea behind the filtering methods is that since the matrix user-movie rating matrix is sparse, sparsity can be filled with correlation. Collaborative filtering models are divided into two categories which are neighborhood-based collaborative filtering and model-based collaborative filtering.

Therefore, we have firstly read the ratings.csv into a dataframe, then constructed the ratings matrix using all samples in the ratings.csv.

Question 1: Explore the Dataset

A) Compute the sparsity of the movie rating dataset:

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

We have found out that the sparsity of ratings matrix is 0.0169996, which means that 98.3% of the ratings matrix is empty, meaning the user did not rate or watch the movie

B) Plot a histogram showing the frequency of the rating values:

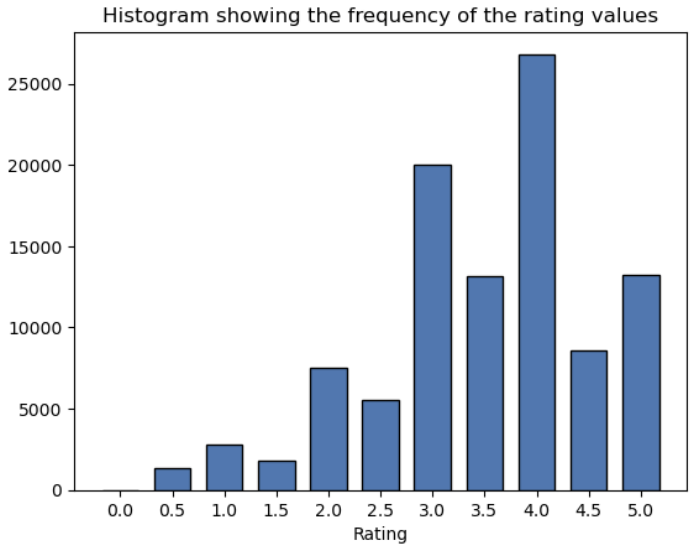

Figure Q1-B: Frequency of ratings in the ratings matrix

- There are no zero ratings.
- Most of the ratings are 4.0.
- Ratings below 3.0 are rare.

C) Plot the distribution of the number of ratings received among movies:

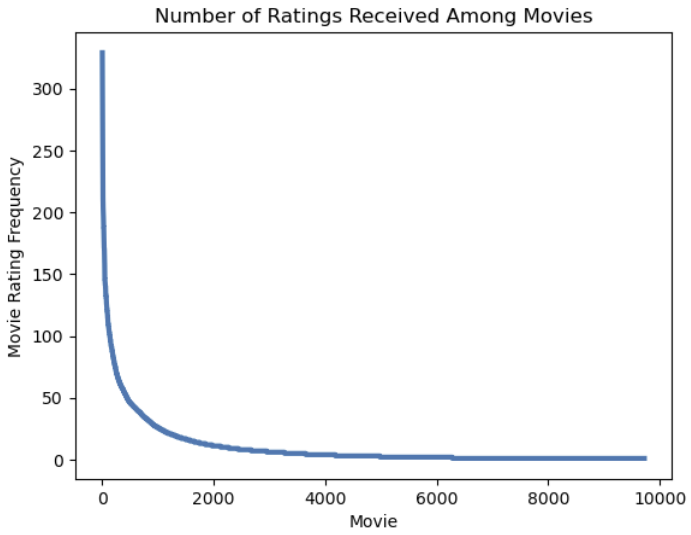

Figure Q1-C: Distribution of the number of ratings received among movies

- Most movies are rated between 0-50 times.
- Most rated movie is rated 329 times.
- Least rated movie is rated 1 time.

D) Plot the distribution of ratings among users:

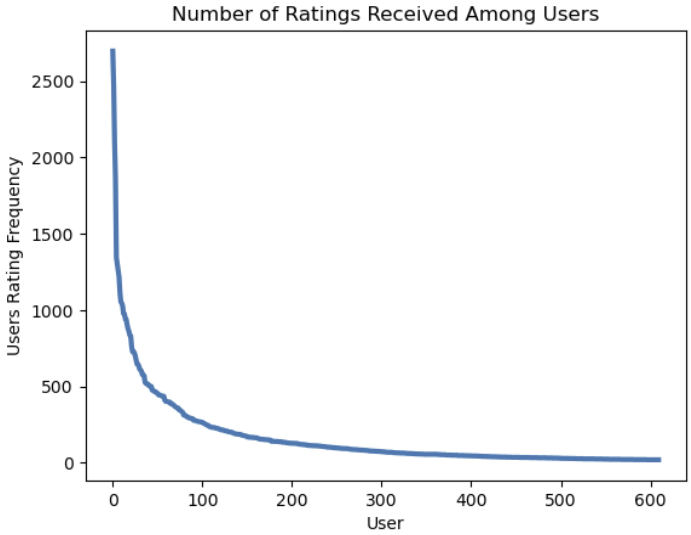

Figure Q1-D: Distribution of ratings among users

- Most users rated 0-100 movies.
- The user that has rated the most movies, rated 2698 movies.
- The user that has rated the least movies, rated 20 movies.

E) Discuss the salient features of the distributions.

By examining the plot at part C, we can easily deduce that a small number of movies have received the majority of the movies. Most of the films did not get ratings more than 50, whereas the most rated film got 329 ratings.

By examining the plot at part D, we can easily deduce that most of the users did not rate most of the movies. Only a small number of people rated the most movies. Therefore, our recommendation system will mostly rely on a small number of users.

These two deductions show the sparsity of the ratings matrix. There are 9724 movies and 610 users in our ratings matrix, however most users have rated a small number of movies. In addition to that, most users did not rate most movies.

If we had tried to create a recommender system from these ratings, we would have easily overfitted due to the sparsity of the ratings matrix. Some methods for generalization should be used to decrease the bias introduced by the sparsity.

F) Compute the variance of the rating values received by each movie:

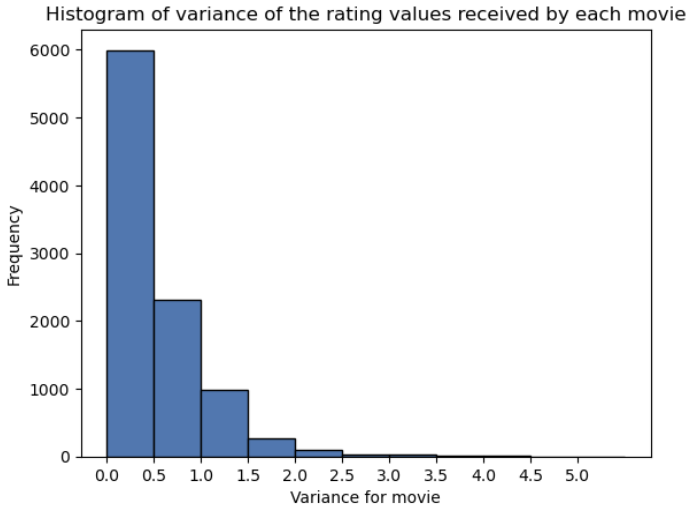

Figure Q1-F: Histogram of variance of the rating values received by each movie

- Most movies have a variance of ratings between 0 and 0.5.
- The MovieId that has the most variance is 2068 with variance 5.0625. This movie has been rated two times, therefore it is likely to achieve that variance.
- The MovieId that has the least variance is 40 with variance 0. Note that, there are movies that have rated only once, therefore the variance of that movie is 0.
- There are 3744 movies that have variance 0, meaning that they have only received one category of rating. 3744 movies is nearly one-third of total movies.
- **Shape of the resulting histogram** indicates that most of the movies get similar ratings from each user since the variance is low for most of the movies. There are a small number of movies that got different ratings from users which yield to high variance. This shows that users give the same rating or near ratings to most movies. However, also note that there are movies that have only one rating, therefore that movie has 0 variance.

Question 2: Understanding the Pearson Correlation Coefficient

A) Write down the formula for μ_u in terms of I_u and r_{uk}.

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

B) In plain words, explain the meaning of $I_u \cap I_v$. Can $I_u \cap I_v = \emptyset$?

$I_u \cap I_v$ is the list of movies where a movie is rated by both user u and user v. It is very likely that the intersection is empty since our ratings matrix is sparse. Most of the users did not rate most of the movies.

Question 3: Understanding the Prediction function: Can you explain the reason behind mean-centering the raw ratings($r_{vj} - \mu_v$) in the prediction function?

Rating pattern for each user is different. Some users will rate 5 to indicate they loved the movie, however some users will rate 4.5 or 4 to indicate their love. Mean-centering will decrease the bias introduced by the different rating patterns for each user. Ratings become balanced and biases that are introduced by the users will be discarded. In the end, the scale of ratings are the same after centering.

Question 4: Design a k-NN collaborative filter to predict the ratings of the movies in the original dataset and evaluate its performance using 10-fold cross validation. K = 2:2:100. Report RMSE, MAE and plots.

10-fold cross validation is used to optimize the k value for k-NN collaborative filters. For each value training dataset is splitted into 10 parts. Then, the validation score is computed 10 times using different combinations for the validation dataset. In each combination, a fold is assigned as a validation dataset to predict the model performance, remaining folds are used for training the model.

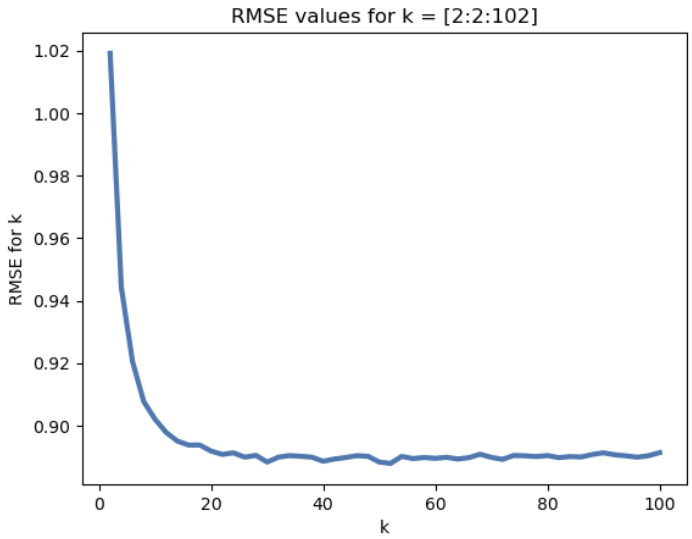

Figure Q4-1: K-NN Based Collaborative Filter RMSE Performance for Different k Values

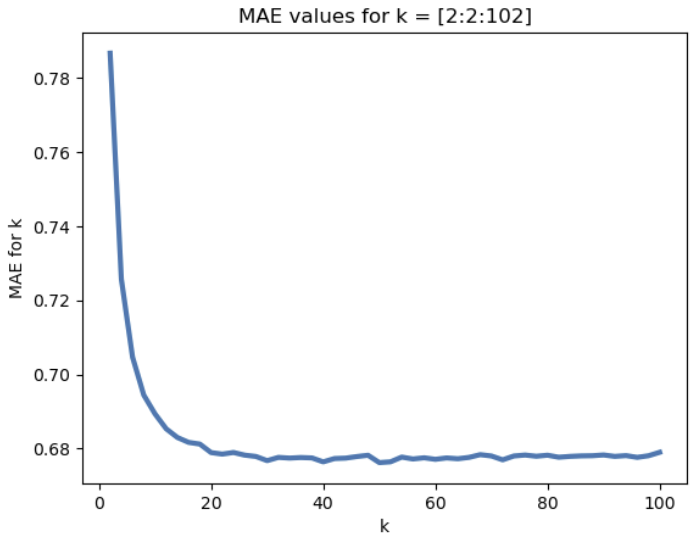

Figure Q4-2: K-NN Based Collaborative Filter MAE Performance for Different k Values

- Monotonic noisy decreasing trend can be seen at both plots. When k increases from 2 to 100, performance improves drastically at some point, then performance stays the same. The point is the steady state value in Question 5.

Question 5: Use the plot from question 4, 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 converge to a steady-state value. Please report the steady state values of average RMSE and average MAE.

- We have selected the steady state **k value to be 20**.
- We have obtained **RMSE = 0.8918** when k = 20.
- We have obtained **MAE = 0.6789** when k = 20.
- To compare the steady state k with minimum k, following is reported:
    - Minimum MAE's k = 50, Minimum RMSE's k = 52
    - Minimum MAE = 0.6761
    - Minimum RMSE = 0.8879
    - Note that, minimum MAE and RMSE is very close to the steady state MAE and RMSE.

Question 6: For each of the 3 subsets in the test set, design:
A k-NN collaborative filter to predict the ratings of movies in the test subset and evaluate each of the three model' performance using 10-fold cross validation. Report minimum average RMSE, plot average RMSE vs k. Plot the ROC curves for the k-NN collaborative filters for threshold values [2.5,3, 3.5, 4]. Report AUC.

- Although it is mentioned in the project specification that the data only in the test set should be trimmed, we trimmed all the dataset because of the announcement in BruinLearn.
- For ROC curves, we have used the best k found in the RMSE plots.
- Popular Subset:

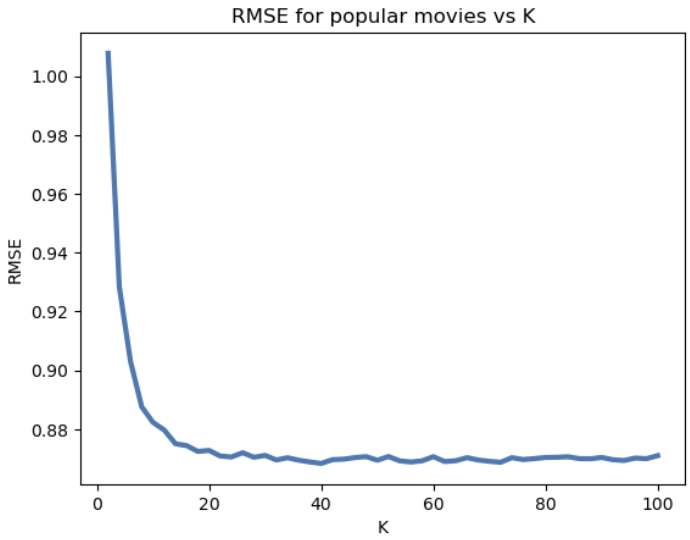

Figure Q6-1: kNN Collaborative Filter Performance on Popular Set

- **Best k for popular set: 40 with RMSE 0.8683394321151073**
- Note that, the k and RMSE is lower for popular dataset when compared with original dataset which shows that unpopular movies brings uncertainty and error to model training.

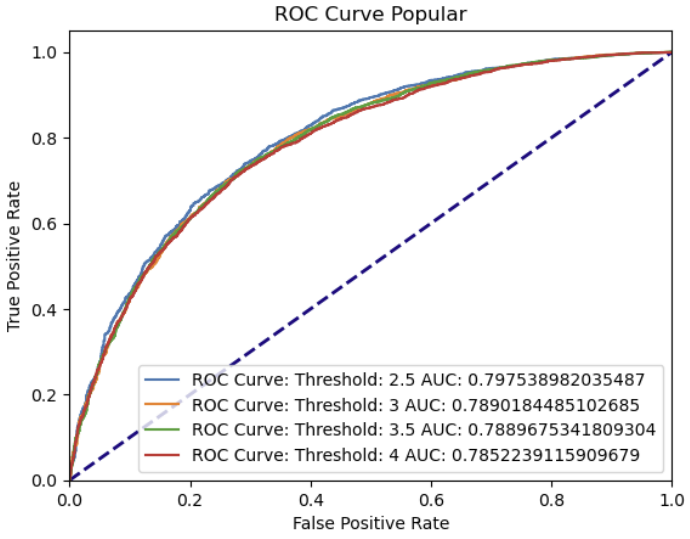

Figure Q6-2: kNN Collaborative Filter ROC on Popular Set, **AUC** values are reported on the figure.

- Unpopular Dataset:

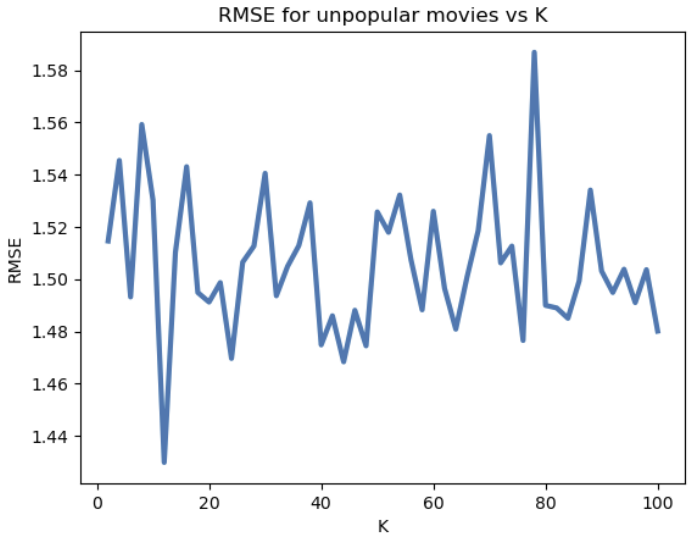

Figure Q6-3: kNN Collaborative Filter Performance on Unpopular Set

- **Best k for unpopular set: 12 with minimum RMSE 1.4298120283170819**
- Note that, RMSE is higher for unpopular dataset, since the subset is small and contains a very sparse ratings matrix. The RMSE curve is very noisy, since our data samples contain too many sparsity ratings and they are low in the number.

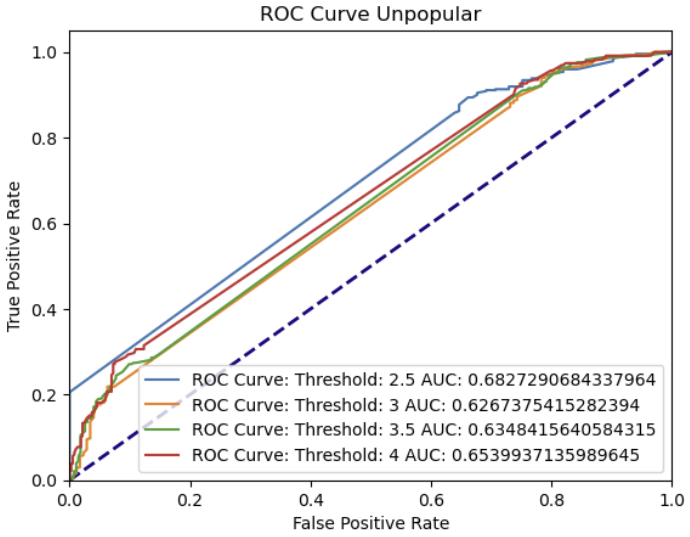

Figure Q6-4: kNN Collaborative Filter ROC on Unpopular Set, **AUC** values are reported on the figure.

- High Variance Dataset:

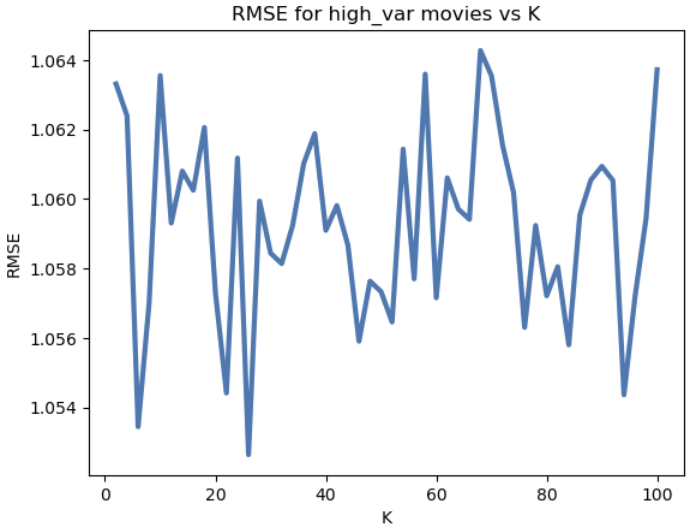

Figure Q6-5: kNN Collaborative Filter Performance on High Variance Set

- **Best k for high_var set: 26 with RMSE 1.052634586650521**
- Note that, RMSE is higher for high variance dataset, since the subset is small and contains a very sparse ratings matrix. The RMSE curve is very noisy, since our data samples contain too many sparsity ratings and they are low in the number. However, also note that RMSE is lower than the unpopular dataset, therefore containing high variance samples do not disrupt the model performance like the popularity of samples.

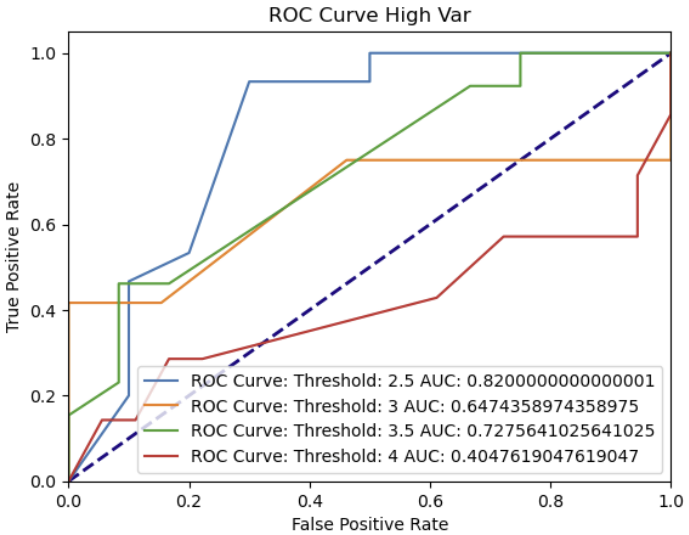

Figure Q6-6: kNN Collaborative Filter ROC on High Variance Set, **AUC** values are reported on the figure.

- Bonus ROC and AUC for original dataset:

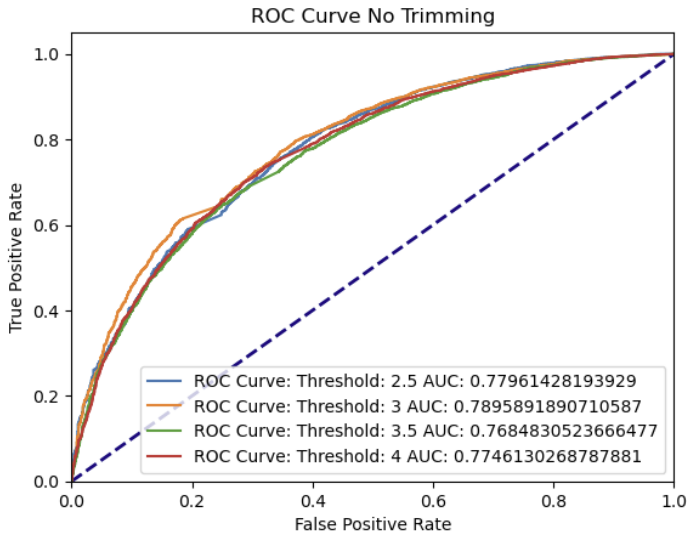

Figure Q6-7: kNN Collaborative Filter ROC on Original Set, **AUC** values are reported on the figure.

Question 7: Understanding the NMF cost function: 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.

We are given that weighting is for filtering only the ratings that are known in the dataset. This is represented by W. Because of this term the problem is no more convex. However, we can resolve the non-convexity by fixing the latent factor matrix U.

The convex weighted least-squares problem can be formulated as the following (U fixed):

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

Question 8A: NMF-based collaborative filtering to predict movie ratings, evaluate with 10-fold CV. Sweep the number of latent factors (k) from 2 to 50 by step size 2. For each k, report average CV RMSE and MAE. Plot RMSE vs k, MAE vs k.

We designed a NMF-based collaborative filtering, and tested our filter with 10-fold CV. We report the average values of RMSE and MAE in Figure Q8-1 for each k.

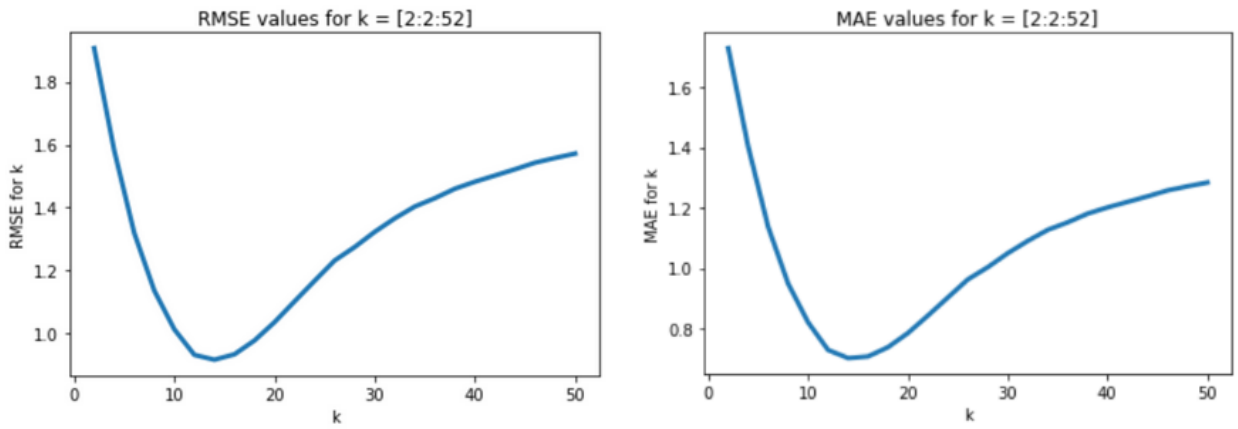

Figure Q8-1: The left figure is the average RMSE for different k, and the right figure is the average MAE for different k.

Question 8B: Use the plot from part A to find the optimal k in terms of min. average RMSE or MAE, report those as well. Is the optimal number of latent factors the same as the number of movie genres?

As can be seen from Figure Q8-1, best RMSE is **0.917**, and the best MAE is **0.702** with **k=14**.

There are 19 labeled genres in the dataset, and there are also movies with no listed genres. Our best filter's k is 14, and this value is close to the number of movie genres 19. We can see that k is close to the number of genres, but less than it. We can infer that data constitutes a space that can be represented with a smaller number of soft labels.

Question 8C: Trimmed Dataset. For each Popular, Unpopular and High Variance subsets design NMF-based collaborative filtering, evaluate with 10-fold CV. Sweep the number of latent factors (k) from 2 to 50 by step size 2. For each k, report average CV RMSE. Plot RMSE vs k. Report min. average RMSE. Plot ROC curves with AUC.

Figure Q8-2 shows the average RMSE evaluated with 10-fold cross validation for the popularity trimming. The best k is **14** with RMSE value of **0.898**.

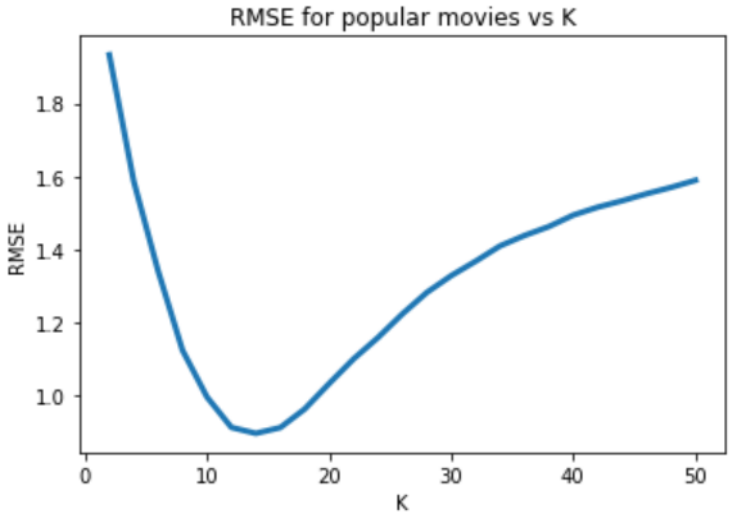

Figure Q8-2: The average RMSE for different k for popular subset movies.

Figure Q8-3 shows the average RMSE evaluated with 10-fold cross validation for the unpopularity trimming. The best k is **20** with RMSE value of **1.557**.

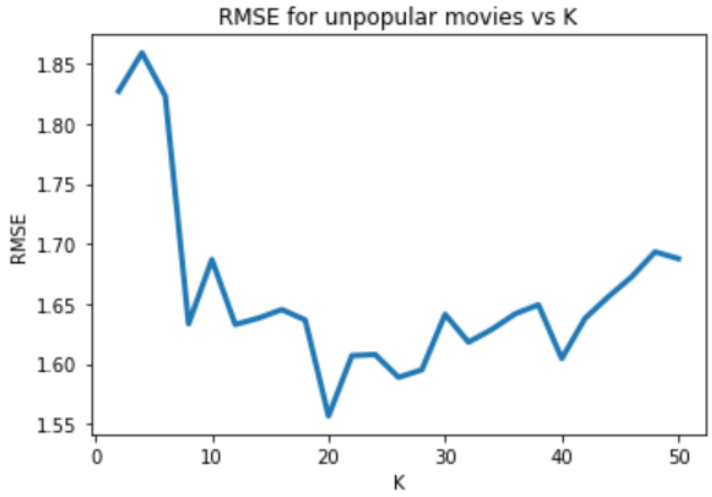

Figure Q8-3: The average RMSE for different k for unpopular subset movies.

Figure Q8-4 shows the average RMSE evaluated with 10-fold cross validation for the high variance trimming. The best k is **18** with RMSE value of **1.14**.

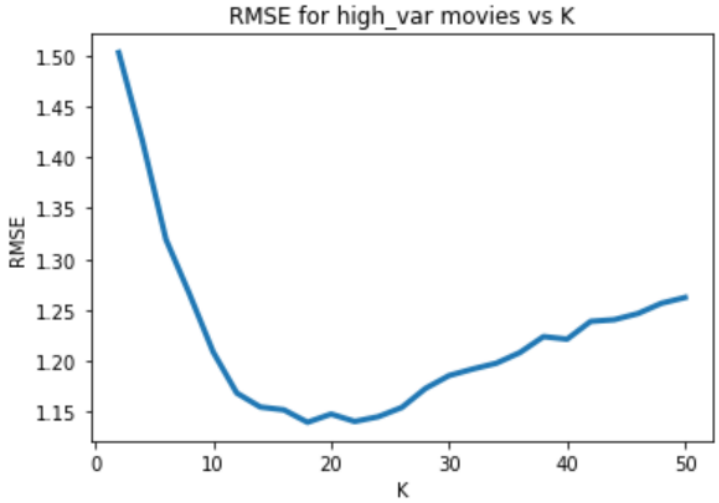

Figure Q8-4: The average RMSE for different k for high variance subset movies.

From the results with 4 different trimming options we can observe that best trimming subsets in terms of RMSE on validation set are as follows:

1. **No trimming - k=14, RMSE=0.702 (best)**
2. Popularity subset - k=14, RMSE=0.898
3. High variance subset - k=18, RMSE=1.14
4. Unpopularity subset - k=20, RMSE=1.557

- The best trimming is obtained with no trimming set.

All the ROC curves are evaluated by setting the number of latent variables of the NMF filter to 20.

Figure Q8-5 shows the ROC curves for different thresholds of no trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.775** with the threshold of **3.0**.

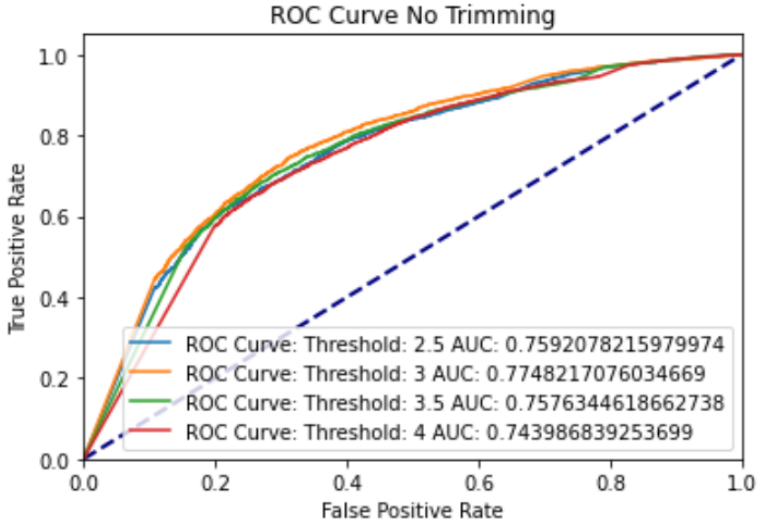

Figure Q8-5: ROC curves for different thresholds of no trimming subset. NMF filter's k is set to 20. The curve is evaluated on the evaluation set.

Figure Q8-6 shows the ROC curves for different thresholds of popular trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.784** with the threshold of **3.0**.

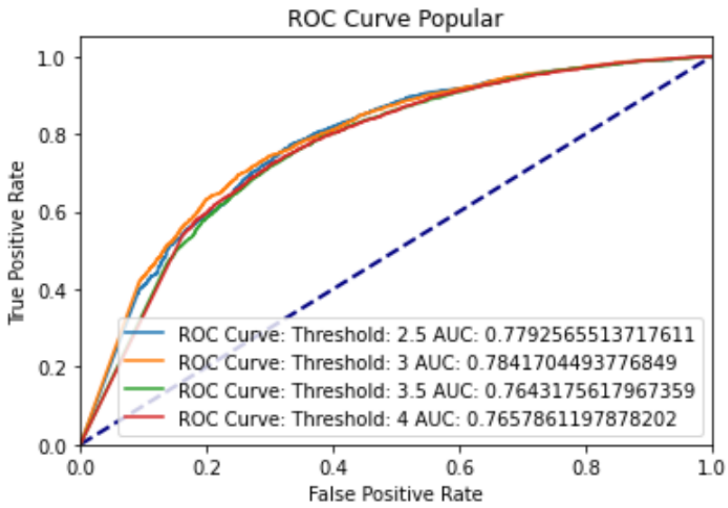

Figure Q8-6: ROC curves for different thresholds of popular subset. NMF filter's k is set to 20. The curve is evaluated on the evaluation set.

Figure Q8-7 shows the ROC curves for different thresholds of unpopular trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.610** with the threshold of **2.5**.

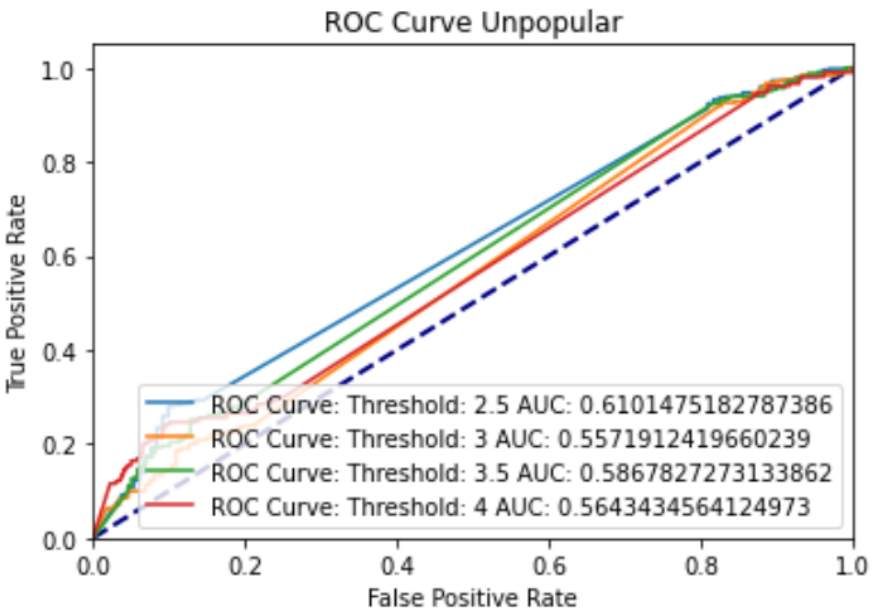

Figure Q8-7: ROC curves for different thresholds of unpopular subset. NMF filter's k is set to 20. The curve is evaluated on the evaluation set.

Figure Q8-8 shows the ROC curves for different thresholds of high variance trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.617** with the threshold of **3.5**.

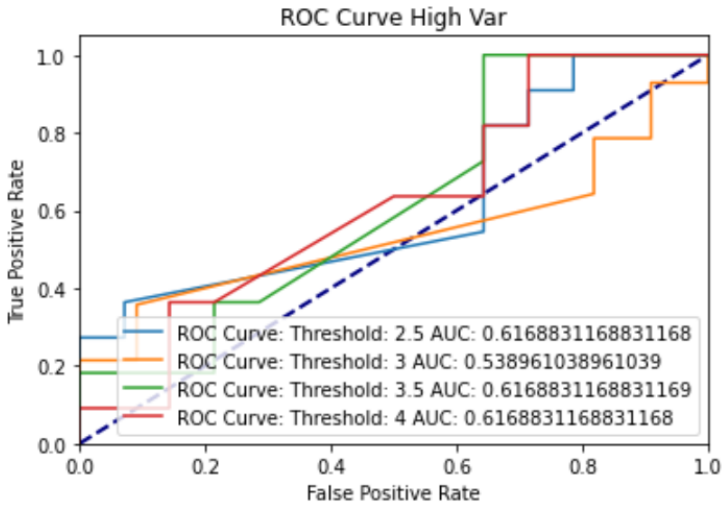

Figure Q8-8: ROC curves for different thresholds of high variance subset. NMF filter's k is set to 20. The curve is evaluated on the evaluation set.

From the AUC results with 4 different trimming options we can observe that best trimming subset in terms of AUC on the evaluation set is as follow:

- Popular trimming, threshold=3.0, AUC=0.784

Question 9: Interpreting the NMF model: Perform NMF 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 genres? Is there a connection between the latent factors and the movie genres?

The question examines the connection between the genres and latent factors. Figure Q9 shows the top 10 movie genres for different latent factor column indices of matrix V. The column indices are started from 1 instead of 0 for reporting. We can observe that there is not a clear distinction of genres in terms of the latent variables. However, we can observe similar genres in different latent variables. For example, in Figure Q9 latent factor 5 is composed of drama and comedy, while latent factor 17 is action, thriller, and horror dominant.

Latent factor index: 1	**Latent factor index: 5**	**Latent factor index: 9**
Animation|Children|Comedy	Adventure|Children|Sci-Fi	Drama
Action|Adventure|Drama|War	Comedy|Drama	Adventure|Drama|Romance
Drama|Sci-Fi|War	Comedy	Action|Crime|Drama|Sci-Fi|Thriller
Comedy|Drama	Drama	Comedy|Crime|Drama|Horror
Adventure|Drama|Romance	Drama	Drama|Horror
Action|Sci-Fi	Documentary	Animation|Children|Fantasy|Mystery
Comedy	Drama	Comedy|Drama|War
Crime|Horror|Thriller	Comedy|Drama|Romance	Drama|Film-Noir|Thriller
Comedy	Comedy|Drama|Musical	Action|Drama|Thriller
Action|Crime|Thriller	Drama|Horror|Romance|Thriller	Adventure|Children|Comedy

Latent factor index: 13	**Latent factor index: 17**
Drama|Mystery|Thriller	Thriller
Drama	Horror
Action|Crime|Drama|Thriller	Action|Drama
Action|Crime|Thriller	Action|Drama|Thriller
Comedy|Drama	Horror|Thriller
Drama|Sci-Fi	Action|Crime|Drama|Thriller
Action|Crime|Thriller	Crime|Drama
Action|Adventure|Fantasy|Sci-Fi	Animation|Children|Comedy
Drama|War	Comedy|Drama|Romance|Thriller
Action|Comedy|Crime|Drama	Crime|Drama

Figure Q9: Top 10 movie genres for different latent factor column indices of matrix V. The column indices are started from 1 instead of 0 for reporting.

Question 10A: MF-based collaborative filtering to predict movie ratings, evaluate with 10-fold CV. Sweep the number of latent factors (k) from 2 to 50 by step size 2. For each k, report average CV RMSE and MAE. Plot RMSE vs k, MAE vs k.

We designed a MF-based collaborative filtering, and tested our filter with 10-fold CV. We report the average values of RMSE and MAE in Figure Q10-1 for each k.

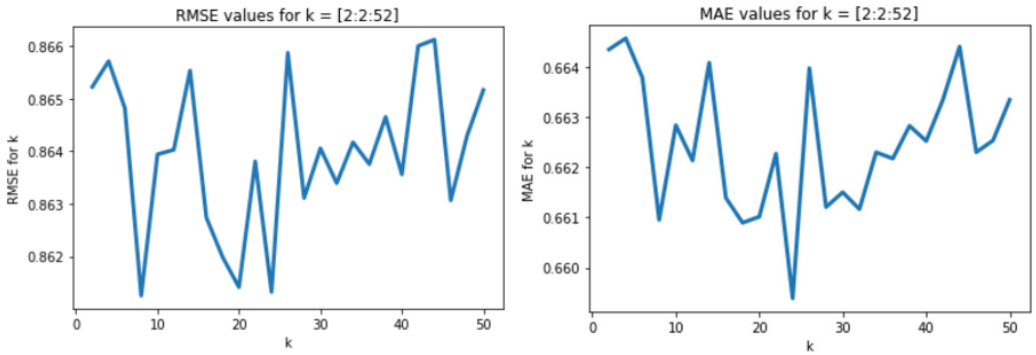

Figure Q10-1: The left figure is the average RMSE for different k, and the right figure is the average MAE for different k.

Question 10B: Use the plot from part A to find the optimal k in terms of min. average RMSE or MAE, report those as well. Is the optimal number of latent factors the same as the number of movie genres?

As can be seen from Figure Q10-1, the best RMSE is **0.861 with k=8**, and the best MAE is **0.66** with **k=24**.

There are 19 labeled genres in the dataset, and there are also movies with no listed genres. Although the best RMSE is with k=8, we can also observe similarly best RMSE for k=20 and k=24. For MAE, the best k is 24. We can infer that the best k is between 8 and 24 but close 20-24. We know that the number of genres is 19 (excluding not listed genres), and we observe that the best k range is close to this number.

Question 10C: Trimmed Dataset. For each Popular, Unpopular and High Variance subsets design MF-based collaborative filtering, evaluate with 10-fold CV. Sweep the number of latent factors (k) from 2 to 50 by step size 2. For each k, report average CV RMSE. Plot RMSE vs k. Report min. average RMSE. Plot ROC curves with AUC.

Figure Q10-2 shows the average RMSE evaluated with 10-fold cross validation for the popularity trimming. The best k is **28** with RMSE value of **0.8510**.

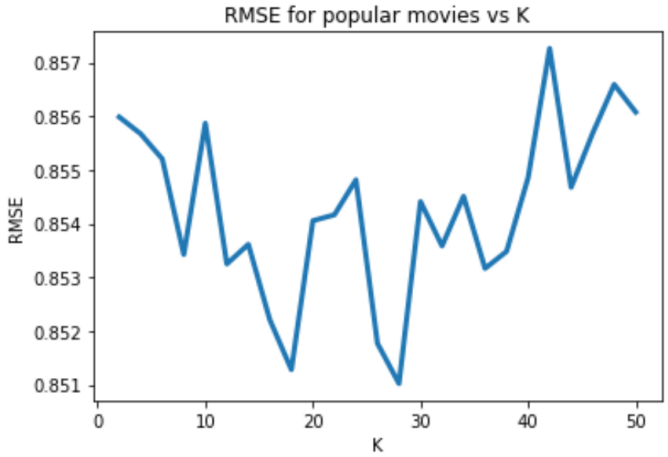

Figure Q10-2: The average RMSE for different k for popular subset movies.

Figure Q10-3 shows the average RMSE evaluated with 10-fold cross validation for the unpopularity trimming. The best k is **34** with RMSE value of **1.549**.

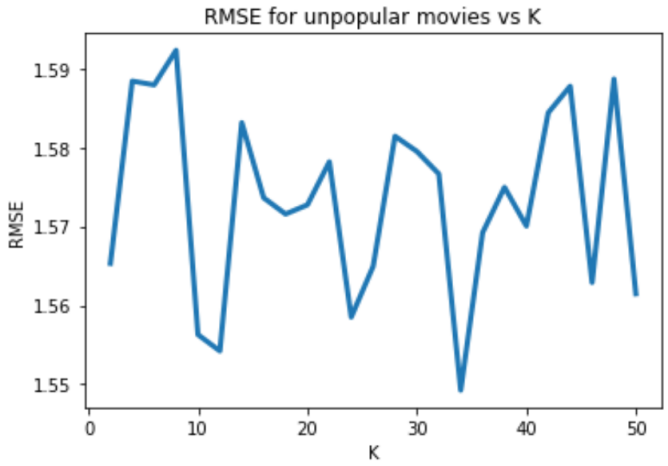

Figure Q10-3: The average RMSE for different k for unpopular subset movies.

Figure Q10-4 shows the average RMSE evaluated with 10-fold cross validation for the high variance trimming. The best k is **12** with RMSE value of **0.8913**.

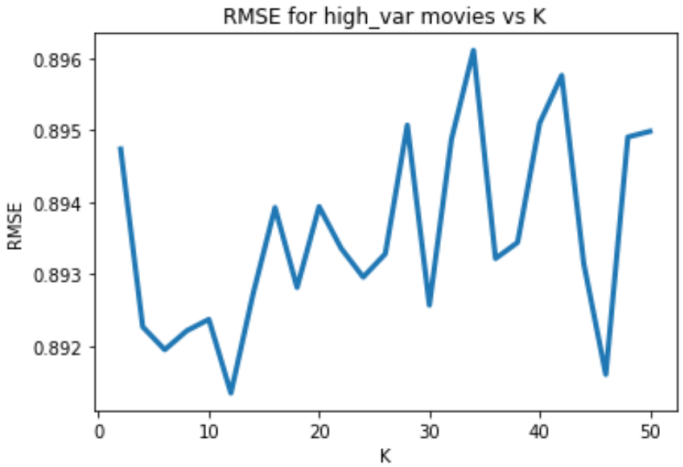

Figure Q10-4: The average RMSE for different k for high variance subset movies.

From the results with 4 different trimming options of MF-based filtering we can observe that best trimming subsets in terms of RMSE on validation set are as follows:

1. **Popularity subset - k=28, RMSE=0.8510 (best)**
2. No trimming - k=8, RMSE=0.861
3. High variance subset - k=12, RMSE=0.8913
4. Unpopularity subset - k=34, RMSE=1.549

The best trimming is obtained with a popularity set unlike NMF-based filtering. We can observe that MF-based filtering cannot outperform NMF-based filtering in terms of RMSE.

All the ROC curves are evaluated by setting the number of latent variables of the NMF filter to 20.

Figure Q10-5 shows the ROC curves for different thresholds of no trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.799** with the threshold of **3.0**.

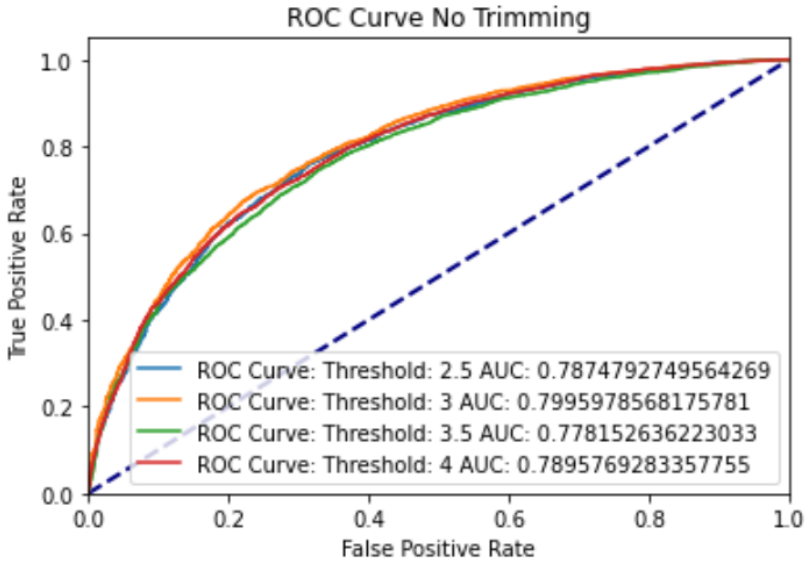

Figure Q10-5: ROC curves for different thresholds of no trimming subset. MF filter's k is set to 20. The curve is evaluated on the evaluation set.

Figure Q10-6 shows the ROC curves for different thresholds of popular trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.8** with the threshold of **3.0**.

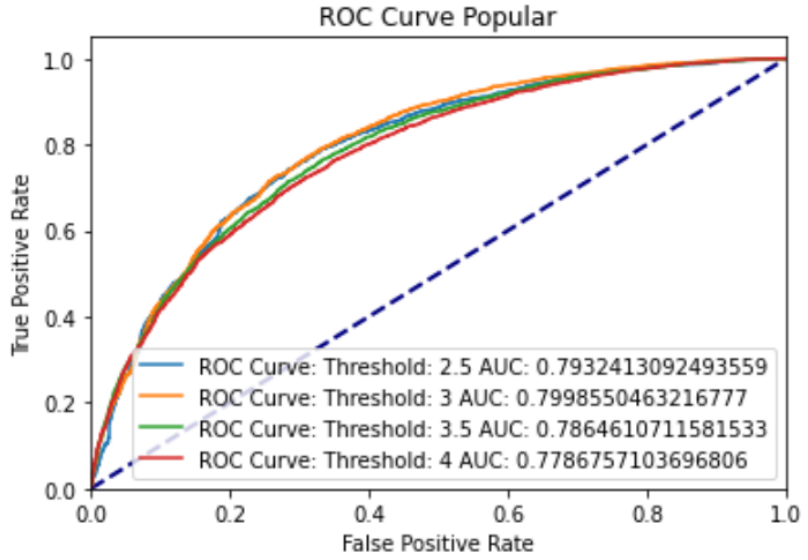

Figure Q10-6: ROC curves for different thresholds of popular subset. MF filter's k is set to 20. The curve is evaluated on the evaluation set.

Figure Q10-7 shows the ROC curves for different thresholds of unpopular trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.838** with the threshold of **3.0**.

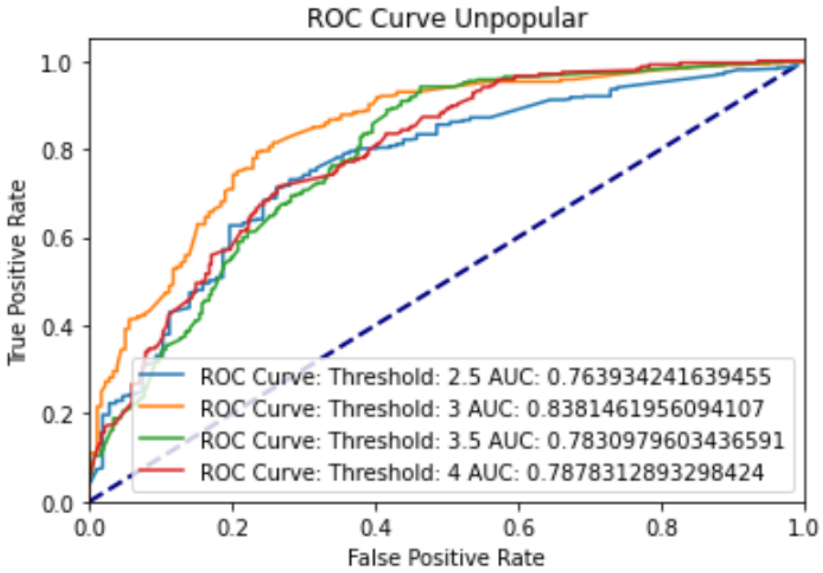

Figure Q10-7: ROC curves for different thresholds of unpopular subset. MF filter's k is set to 20. The curve is evaluated on the evaluation set.

Figure Q10-8 shows the ROC curves for different thresholds of high variance trimming subset evaluated on the evaluation set. The best AUC for this trimming is **0.676** with the threshold of **3.5**.

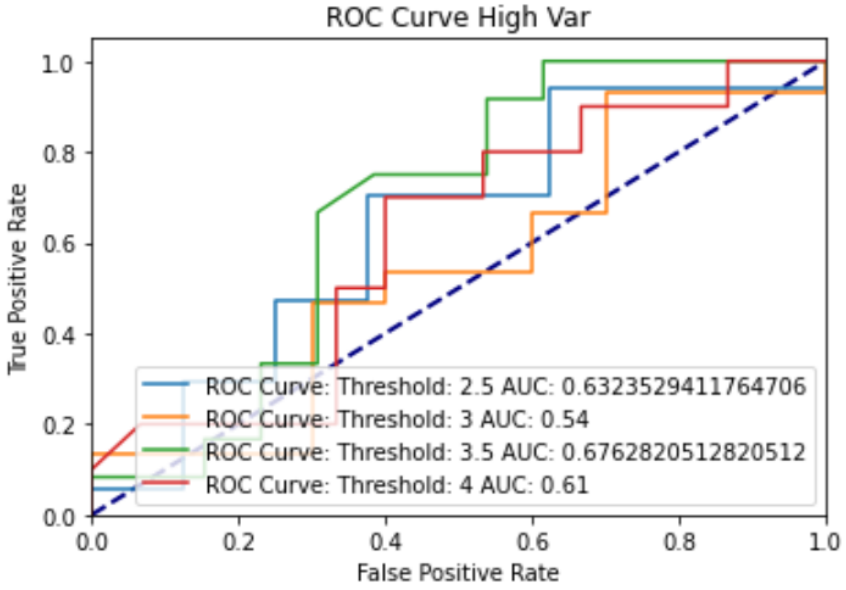

Figure Q10-8: ROC curves for different thresholds of high variance subset. MF filter's k is set to 20. The curve is evaluated on the evaluation set.

From the AUC results with 4 different trimming options we can observe that best trimming subset in terms of AUC on the evaluation set is as follow:

- Unpopular trimming, threshold=3.0, AUC=0.838

Question 11: Designing a Naive Collaborative Filter: • Design a naive collaborative filter to predict the ratings of the movies in the original dataset and evaluate it's performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE. • Performance on dataset subsets: For each of Popular, Unpopular and High-Variance test subsets - – Design a naive collaborative filter for each trimmed 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.

```
# Root Mean Squared Error for Movies:  0.93470117279113
# Root Mean Squared Error for Popular Movies:  0.9323293074687037
# Root Mean Squared Error for Unpopular Movies:  0.9708400340756242
# Root Mean Squared Error for High Variance Movies:  1.458850183004931
```

Question 12: Comparing the most performant models across architecture: Plot the best ROC curves (threshold = 3) for the k-NN, NMF, 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.

We constructed three different models with different k values according to the cross validation results we obtained in the previous parts. The threshold value was determined as 3 and the movies with rankings above the threshold value was set to 1. In the figure below, we drew the ROC curves for these models.

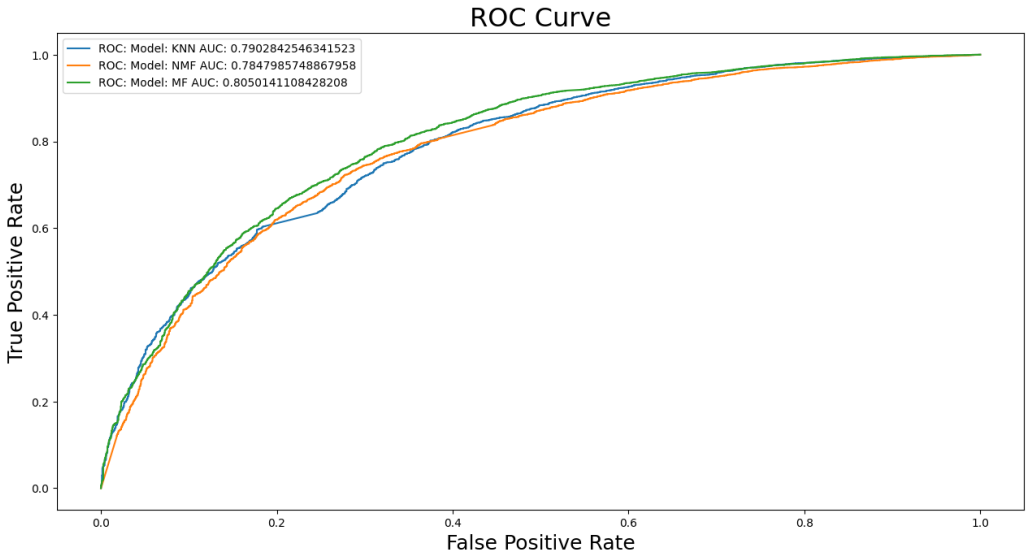

Figure Q12: FPR vs TPR ROC Curve for KNN, NMF and MF models.

From the ROC Curves combined with the AUC scores we can observe that MF model with the bias outperformed KNN and NMF models.

Question 13: Understanding Precision and Recall in the context of Recommender Systems: 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.

Precision refers to the proportion of true positive predictions (i.e., the number of correct positive predictions) out of all positive predictions made by the model. In other words, precision measures the accuracy of the positive predictions made by the model. Recall, on the other hand, refers to the proportion of true positive predictions out of all actual positive instances in the dataset. In other words, recall measures the completeness of the positive predictions made by the model. In simpler terms, precision is a measure of how many of the positive predictions made by the model are actually correct, while recall is a measure of how many of the actual positive instances the model was able to correctly identify.

Question 14: For each of the three architectures: – Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using the model's predictions. – Plot the average recall (Y-axis) against t (X-axis) and plot the average precision (Y-axis) against average recall (X-axis). – Use the best k found in the previous parts and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot. • Plot the best precision-recall curves obtained for the three models (k-NN, NMF, MF) in the same figure. Use this figure to compare the relevance of the recommendation list generated using k-NN, NMF, and MF with bias predictions.

In this part, we examined the precision and recall values of the previously explained three models. For KNN we pick the hyperparameter k as 20 and for NMF we pick k = 14 while k = 24 gives the best performance measure for MF with the bias.
In the next set of figures we display the precision and recall values as number of recommended items increase.

KNN Model

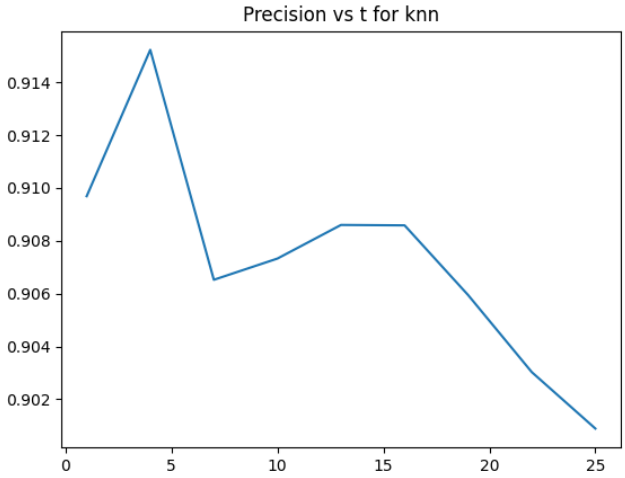

Figure Q14a: Precision versus # of movies recommended

As the # of movies recommended increases the precision is diminished.

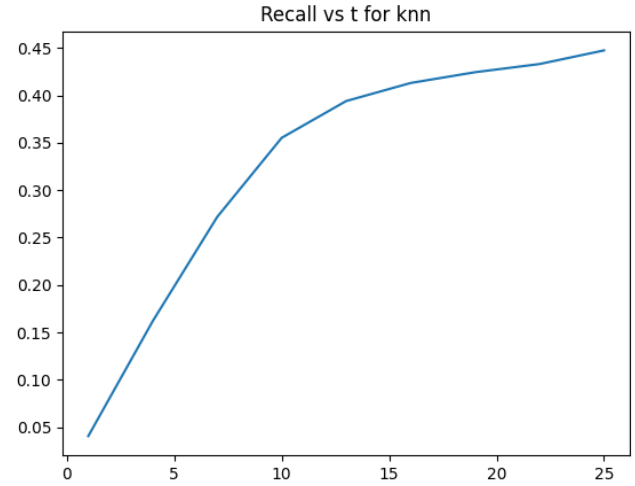

Figure Q14b: Recall versus # of movies recommended

As we increase the movies we recommend it is likely that the overlapping between the ground truth liked movies and the recommended movies will get bigger. Thus, it is expected to observe an increasing trend in recall rate when t increases.

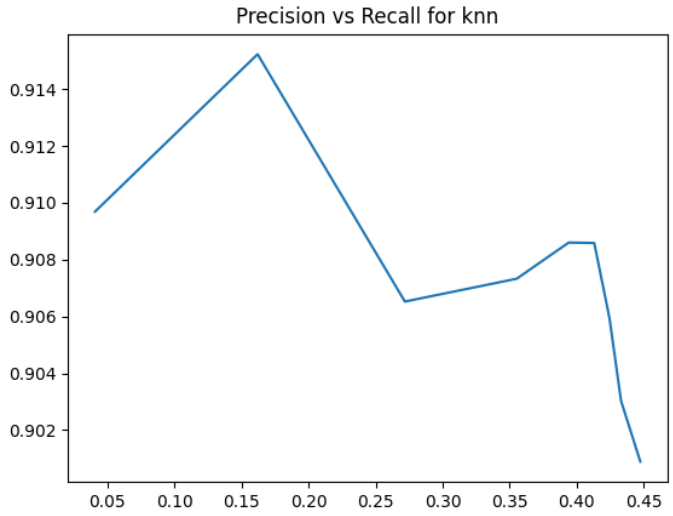

Figure Q14c: Precision versus Recall for KNN models

As seen from Fig Q14c, the recall and precision are inversely proportional, this is a natural outcome since for a fixed TP, FN decreases as FP increases. In other words, precision is less accurate when recall value is increased.

NMF Model

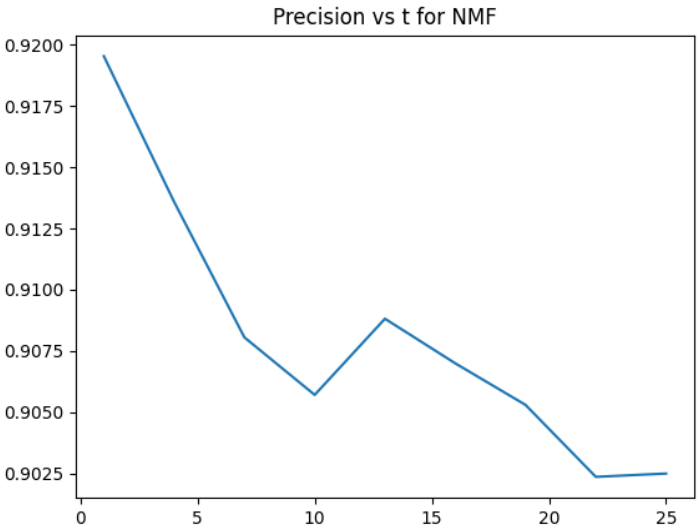

Figure Q14d: Precision versus # of movies for NMF models

The same comments can be made for both NMF and MF models as we did in KNN models. The inverse proportion between #of movies recommended and the precision is expected as well as the direct proportion between recall and the #of movies recommended.

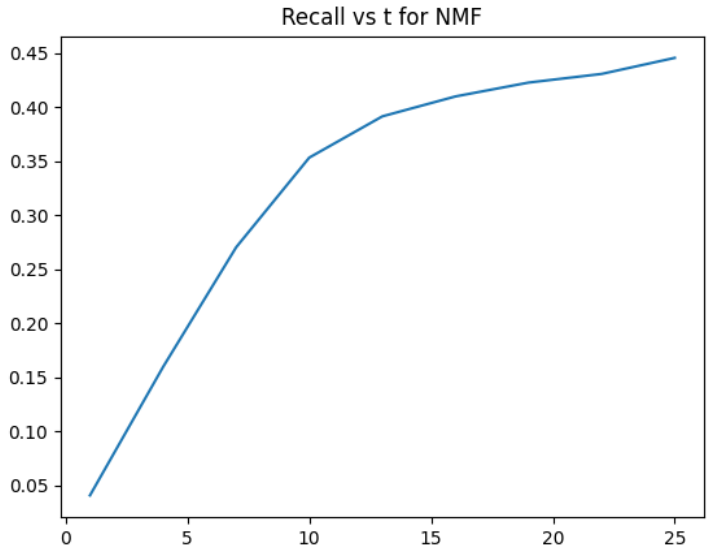

Figure Q14e: Precision versus # of movies for NMF models

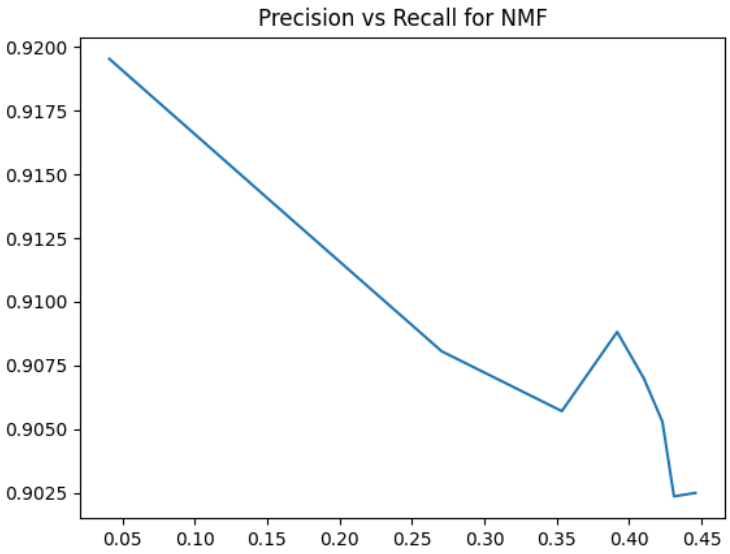

Figure Q14f: Precision versus Recall for NMF models

MF with Bias

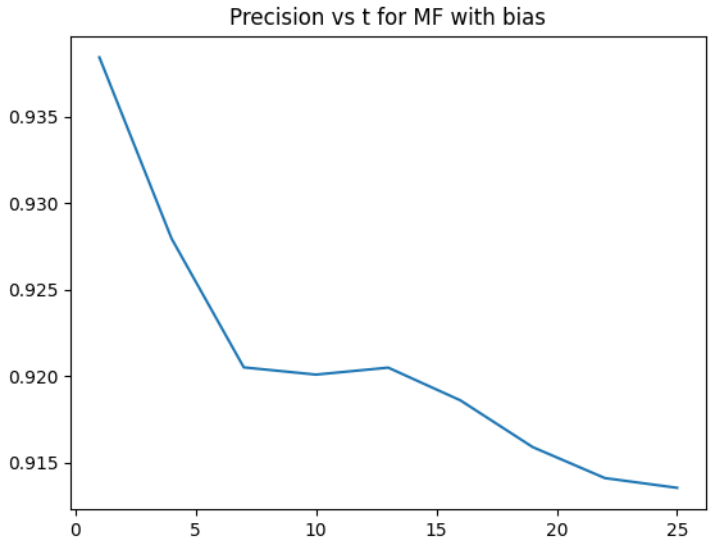

Figure Q14g: Precision versus # of movies for MFmodels

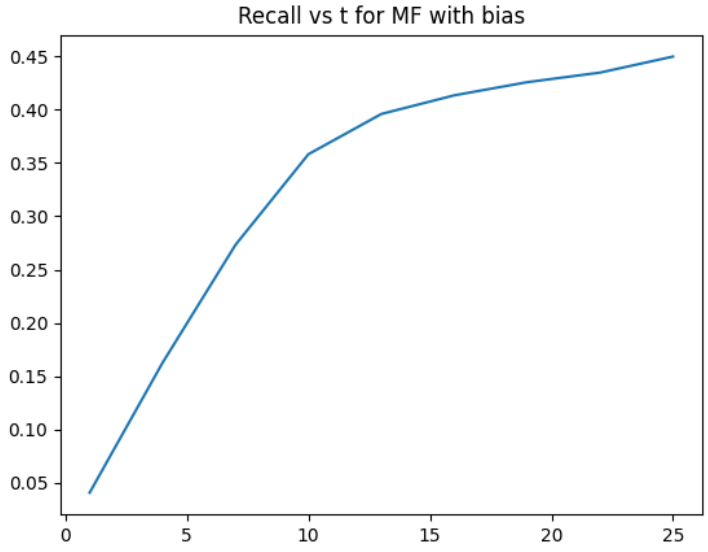

Figure Q14h: Precision versus # of movies for MFmodels

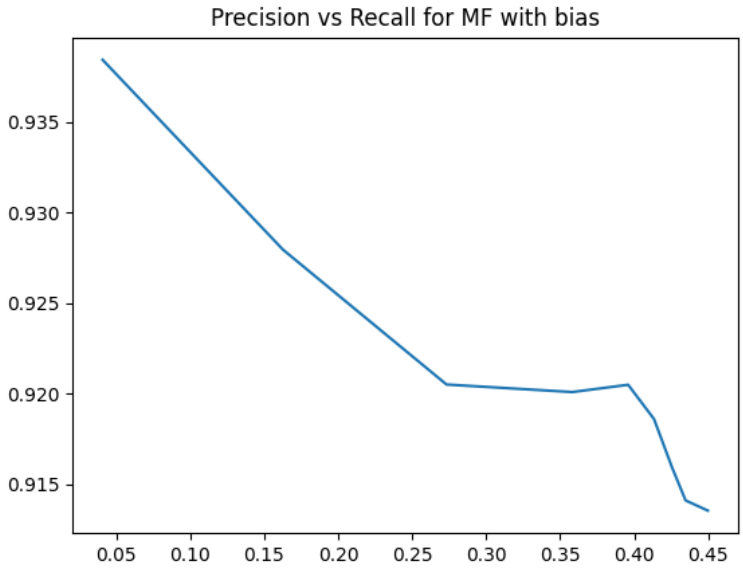

Figure Q14i: Precision versus Recall for MF models

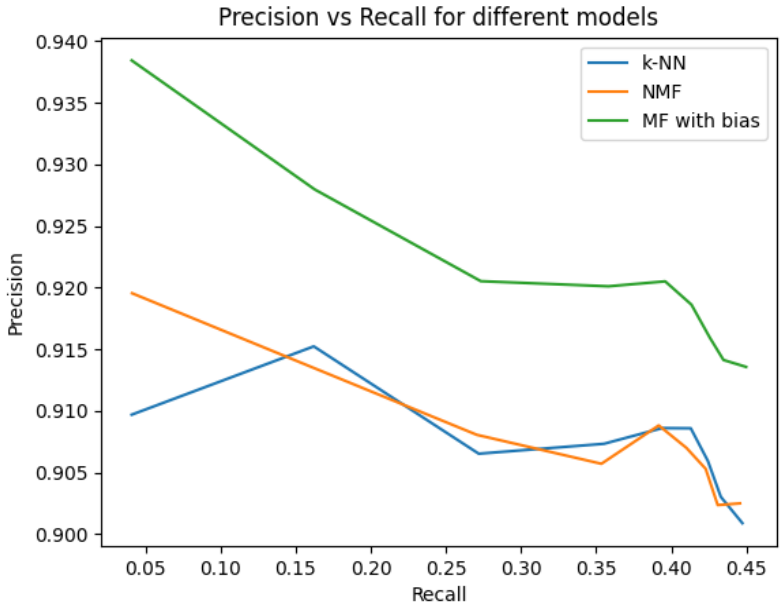

Figure Q14k: Comparison of all Three Models

The precision by itself does not carry much information about the accuracy of the model since it only concerns the TP and FP rates. On the other hand, it makes sense to scrutinize the precision values for a fixed recall rate for different models since this approach yields a fixed ratio between

FN and TP for each of the models. From Figure Q14k, we can observe that for the same recall, MF with bias outperforms the other two models.