219 - Large-Scale Data Mining: Models and Algorithms

Report on

Project 3: Recommender Systems

Authors

Jayanth Shreekumar (805486993)

Narayan Gopinathan (605624014)

Yuheng He (505686149)

Winter 2022

Introduction

Recommender systems are algorithms that suggest similar items that the user has had experience with in the past. They do this by predicting the rating that a user would give to an item. The term "user" here refers to the entity to which the recommendation is provided, and the term "item" is the product being recommended to the user. Recommender systems have become pivotal in our daily lives with the advent of companies such as Amazon, Netflix, TikTok etc. where the of data is so vast that prioritizing what a specific user would enjoy is an important part of the industry. Recommender systems can be broadly classified into:

- Collaborative filtering
- Content-based filtering

In this project, we focus on collaborative filtering methods.

Collaborative filtering models utilize the past interactions between users and items to recommend items to users. The intuition is that these user-item interactions can be used to recommend items to a current user that were bought by a different user that had similar preferences. Therefore, it is more likely that the current user would buy the product too. Collaborative filtering is sub-divided into two categories:

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

Question 1

Question 1 A

The dataset contains a total of 100836 ratings accounting for 9724 movies voted by 610 users. The sparsity of the dataset is given by the formula:

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

where Total number of possible ratings = (Number of users) $\times$ (Number of movies)

The sparsity of our dataset is:
Sparsity = .017

Clearly, the dataset is sparse, meaning that most users have rated only a few movies, and only a few movies have been rated by multiple users.

Question 1 B

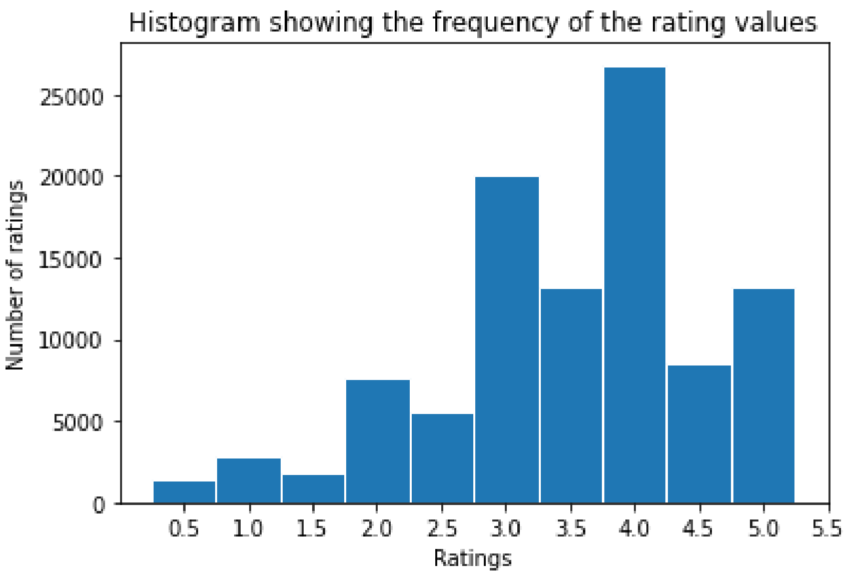

Figure 1: The histogram for the number of movies with each rating

We see that most users rate most movies highly. There is a selection bias because they look at ratings before selecting movies, and so highly rated movies are more likely to be chosen, and receive more ratings. Also, most users seem to prefer integer ratings as seen in the graph.

Question 1 C

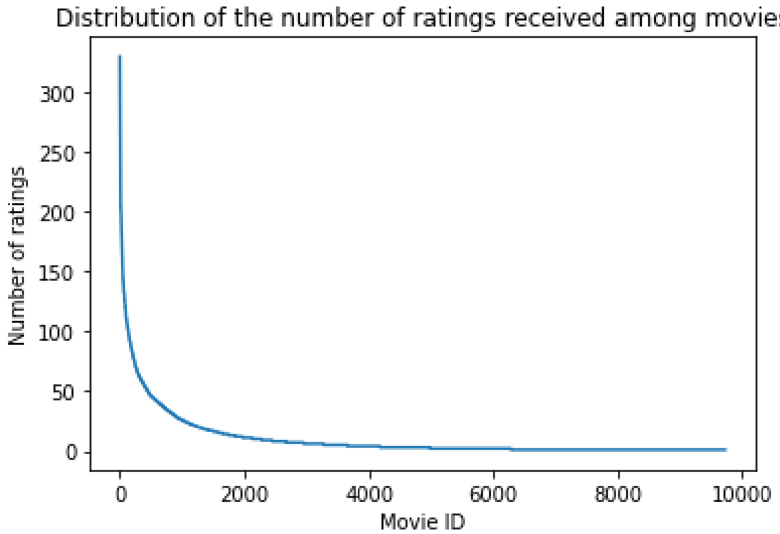

Figure 2: The distribution of the number of ratings that each movie received

We see that the curve is exponentially decreasing, implying that very few movies are actually rated by more than 50 people.

Question 1 D

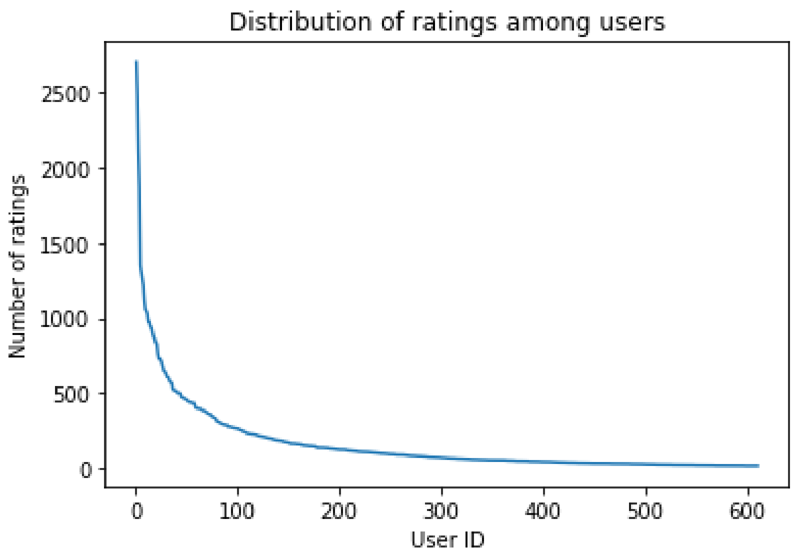

Figure 3: The distribution of the ratings among users

Similar to 1 C, we see that the curve is exponentially decreasing, implying that very few users actually rate multiple movies.

From questions 1C and 1D, we observe that these are the reasons for sparsity.

Question 1 E

The salient features are that there are very few movies that are highly rated and there are very few movies that are rated consistently. The majority of movies receive few ratings because they are not widely viewed. This means that the matrix is very sparse. The distribution appears hyperbolic in nature. The number of ratings drops off quickly as the number of users increases, as a very small portion of the movies have a large number of ratings.

Sparse matrices are a problem as the volume of the data increases rapidly in each dimension. Due to this, the idea of distance becomes vague, with each entry being roughly equidistant to each other. This is the curse of dimensionality. Further, as a lot of the entries are defaulted to 0 as they are not present, we can also face the problem of overfitting: we need a higher order model to fit the data itself as it is in very high dimension, but due to not enough data, it is easier to fit all the data when in that higher dimensional space.

Question 1 F

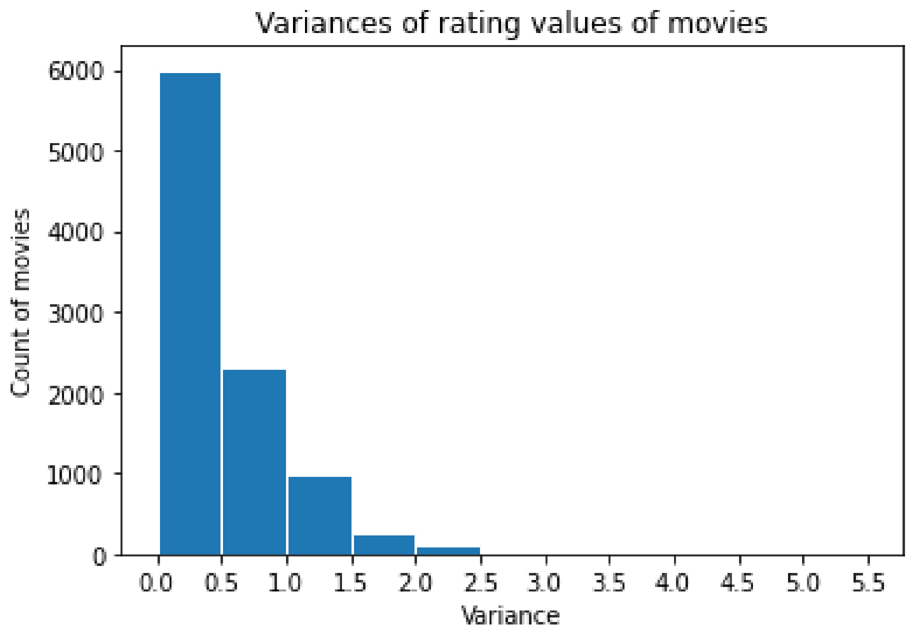

Figure 4: Variance of rating values received by each movie

We see that most movies have very low variance in the range of 0 to 1. Therefore, we conclude that the ratings are consistent and movies that are highly rated a few times are expected to continue to be highly rated because people will watch movies by taking recommendations using the ratings published online and only watch movies that are deemed "good" and in turn rate them highly. Similarly, movies that do not receive a high rating are expected to continue in the same trend.

Question 2

Question 2 A

The mean will be equal to the sum of all user ratings given by the user divided by the number of ratings and is given by:

$$\mu_u = \frac{\sum_{k \in I_u} r_{uk}}{len(I_u)}$$

Question 2 B

As I_u denotes the set of movies rated by user u and I_v denotes the set of movies rated by user v, $I_u \cap I_v$ denotes all the movies that have been rated by both users u and v. As there is a possibility that two users might not have any common movies in the ones that they have rated, $I_u \cap I_v$ can be $\emptyset$.

Question 3

If an individual rates all movies highly, then they will shift the baseline ratings to be high. Conversely, if a different user rates all movies poorly, then that can shift the average a lot. This matters a lot if the values are outliers. For example, one user who consistently gives movies one star ratings can decrease the average for everything. We need to do a normalization to ensure that each movie is measured in relation to each users' mean. This enables us to consistently compare ratings from one user to ratings from another.

Question 4

In statistics, the k-nearest neighbors algorithm (k-NN) is a non-parametric supervised learning method. The k-NN of a given user in our scenario can be represented by P_u, which is the top k user with highest Pearson-correlation to the user u.

The Person-correlation coefficient can be expressed as follow

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

where I_u indices item rated by a user u, I_v indices item rated by a user v, μ_u indicates the mean rating for a user u of r ratings, and U_k indicates the ratings of user u for a specific item k.

The prediction function is shown as

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

In this part, we utilized a k-NN collaborative filter in prediction of the ratings featuring the original dataset using 10-fold cross validation. We constructed a series of k values ranging from 2 to 100 with a step size of 2. We obtained the RMSE and MAE score for each individual k value to evaluate the performance of the filter. The results are presented in Fig. 5 and Fig. 6, respectively.

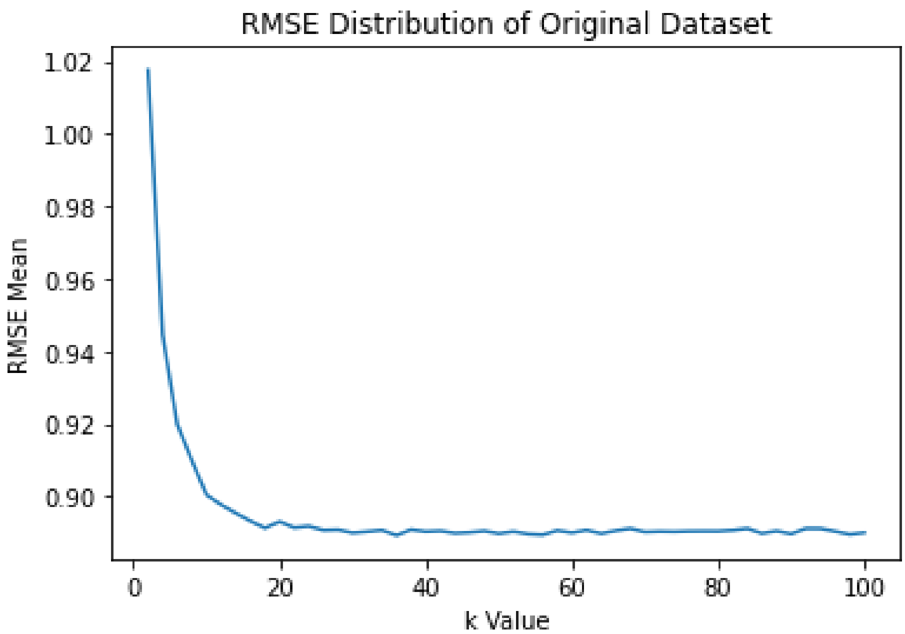

Figure 5: RMSE Distribution of Original Dataset

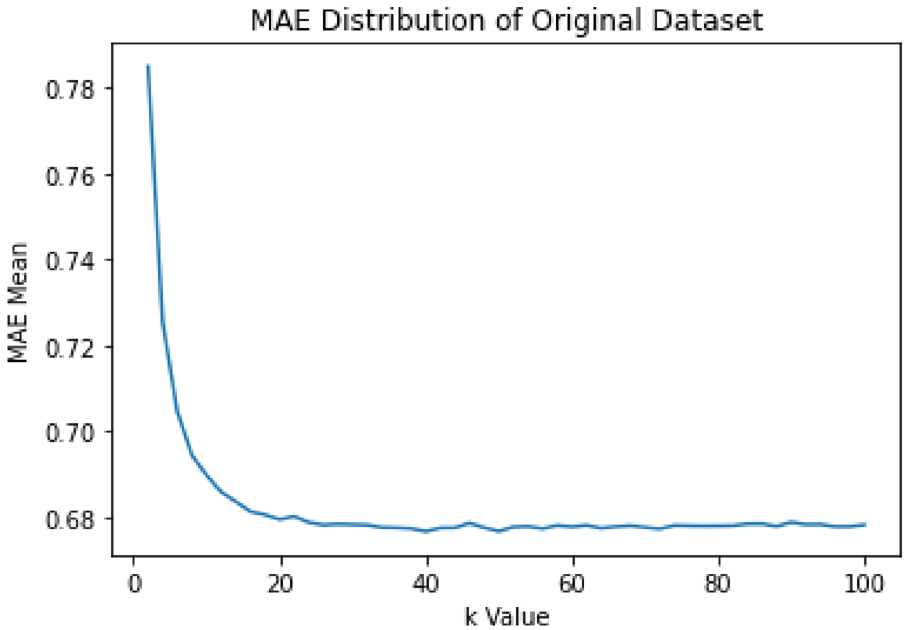

Figure 6: MAE Distribution of Original Dataset

From the resulting plots we can see that the average of RMSE as well as MAE drop significantly as k value increases initially. With this outcome, we conclude that the filter performs better as k increases. Nevertheless, the average scores reach a plateau where they no longer decrease further. The steady-state value of **RMSE is about 0.8886** and the steady state value of **MAE is about 0.6766**. We calculate the minimum k in the next part.

Question 5

The minimum k is about 20 where we see a clear elbow, and the corresponding RMSE value is 0.89170, and the corresponding MAE value is 0.67880.

Question 6

In this part, we first trimmed the data into three different categories, namely, popular dataset, unpopular dataset, and high variance dataset. To each trimmed dataset, we apply k-NN collaborative filter in prediction of the ratings.

We first trimmed the dataset into popular movie dataset. If a movie has received more than 2 ratings, we termed it a popular movie. The popular movie dataset contains movies that are considered popular. Similarly, we applied k-NN collaborative filter with an array of k values spanning from 2 to 100 with a step size of 2. We obtained the RMSE scores to assess the performance of the filter with different k values. The results are plotted in Fig. 7. We also obtained ROC curves of different threshold values: 2.5, 3, 3.5, and 4, which is shown in Fig. 8.

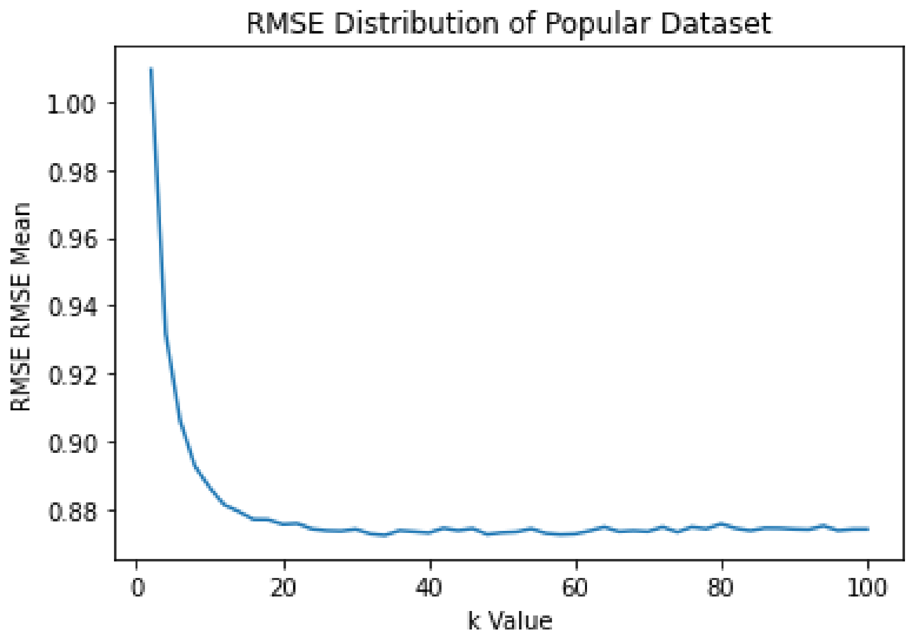

Figure 7: RMSE Distribution of Popular Dataset

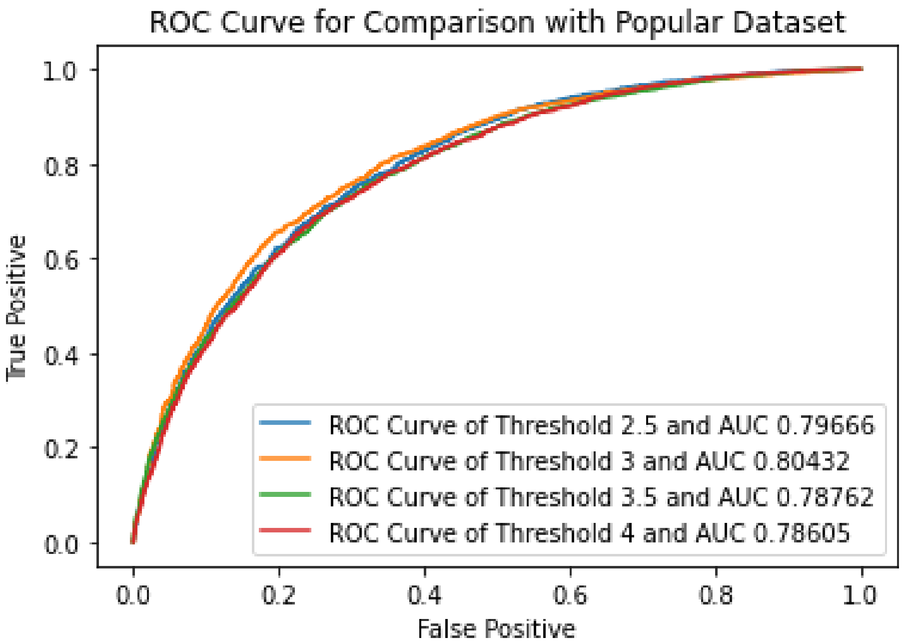

Figure 8: ROC Curve of Popular Dataset

As observed from the figure, the RMSE mean value plummets drastically as k increases, and it also reaches a plateau at a point with $k = 20$. On the other hand, the minimum **RMSE value is 0.87262** at $k = 38$.

The ROC curve shows that the AUC of threshold 2.5 is 0.79666, the AUC of threshold 3 is 0.80432, the AUC of threshold 3.5 is 0.78762, the AUC of threshold 4 is 0.78605, and the largest AUC is 0.80432 with the corresponding threshold 3.

We then trimmed the dataset into unpopular movie dataset. If a movie has received less than or equal to 2 ratings, we termed it an unpopular movie. The unpopular movie dataset contains movies that are considered unpopular. As we performed earlier, a k-NN collaborative filter with an array of k values spanning from 2 to 100 with a step size of 2 is applied. We obtained the RMSE scores to assess the performance of the filter with different k values. The results are plotted in Fig. 9. We also obtained ROC curves of different threshold values: 2.5, 3, 3.5, and 4, which is shown in Fig. 10.

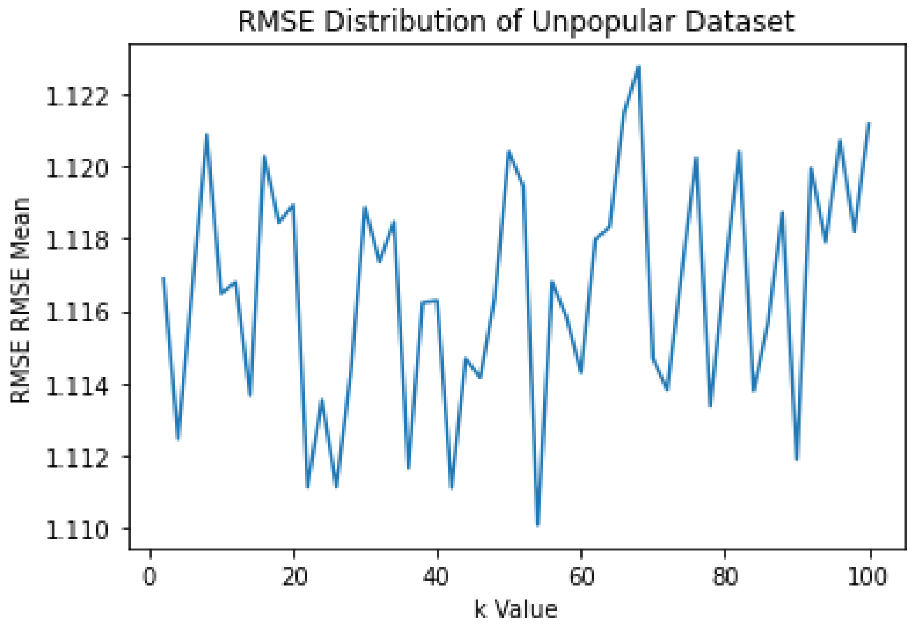

Figure 9: RMSE Distribution of Unpopular Dataset

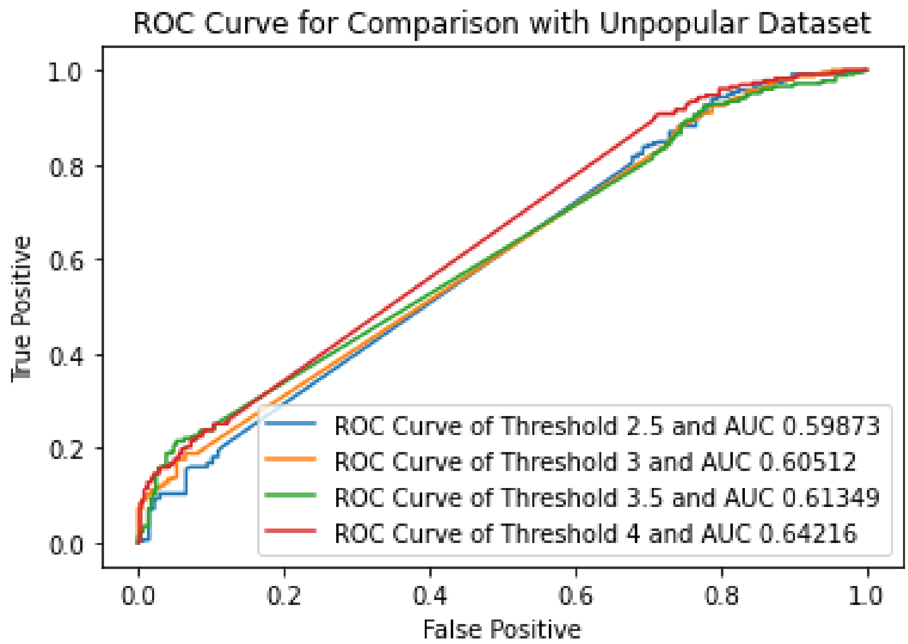

Figure 10: ROC Curve of of Unpopular Dataset

As observed from the figure, the RMSE mean values do not follow the previous results. The RMSE score is oscillating and dose not reach a steady point. This outcome is expected in that there are only a small amount of unpopular movies in the training dataset and with the popular movies removed from the testing dataset we may reach a prediction with a large errors. Hence the performance is not guaranteed to improve as k value increases. The minimum **RMSE value is 1.11007** at $k = 54$.

The ROC curve shows that the AUC of threshold 2.5 is 0.59873, the AUC of threshold 3 is 0.60512, the AUC of threshold 3.5 is 0.61349, the AUC of threshold 4 is 0.64216, and the largest AUC is 0.64216 with the corresponding threshold 4.

Lastly, the dataset is trimmed into high variance movie dataset. The high variance movie dataset contains movies that have variance of at least 2 and has received at least 5 ratings in the entire dataset. The high variance movie dataset contains movies that are considered high variance. Similarly, a k-NN collaborative filter with an array of k values spanning from 2 to 100 with a step size of 2 is applied. We obtained the RMSE scores to assess the performance of the filter with different k values. The results are plotted in Fig. 11. We also obtained ROC curves of different threshold values: 2.5, 3, 3.5, and 4, which is shown in Fig. 12.

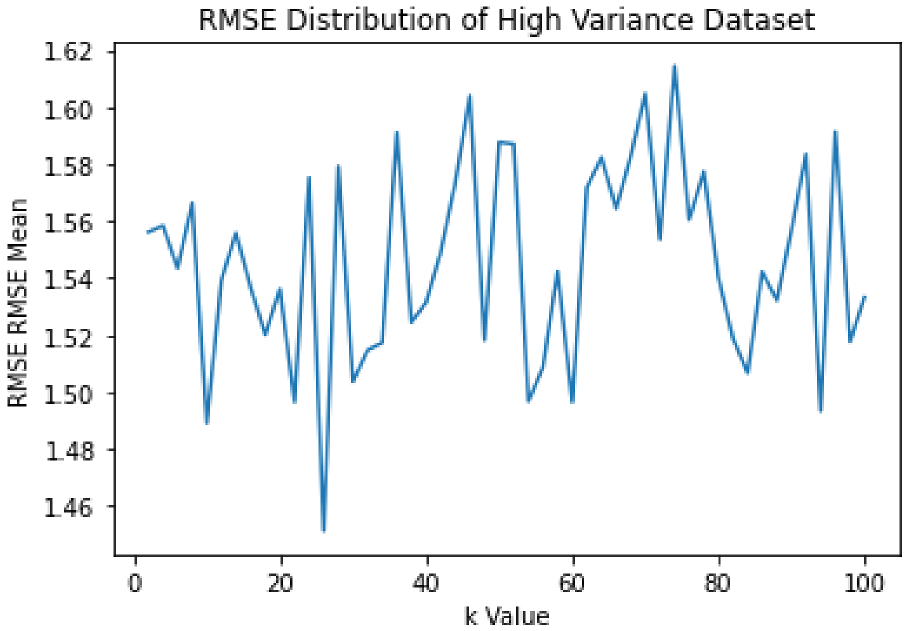

Figure 11: RMSE Distribution of High Variance Dataset

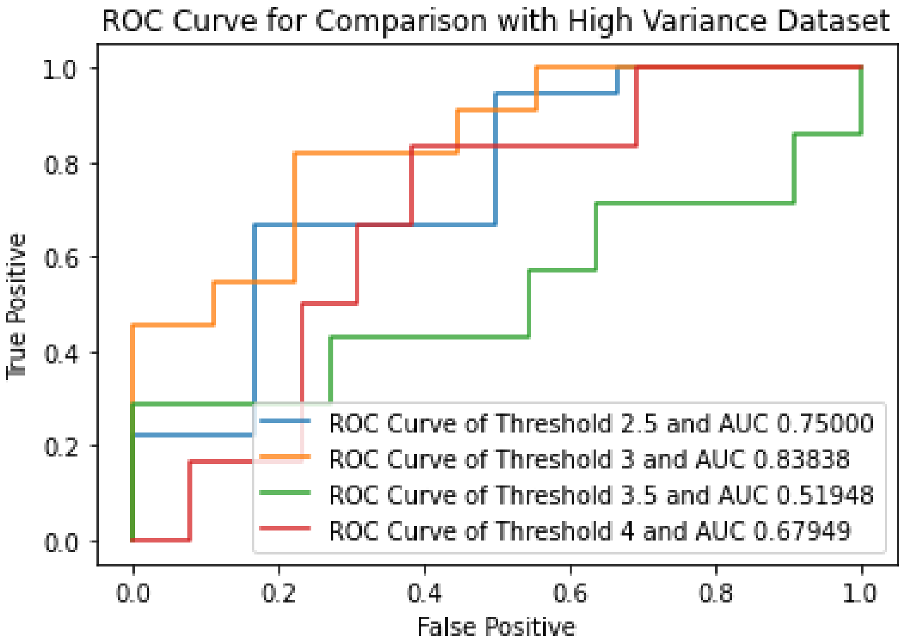

Figure 12: ROC Curve of of High Variance Dataset

We believe that the ROC curve is granular because there are not enough test samples. The figure conveys a similar result as the one in the unpopular dataset, where the RMSE score changes drastically as k value rises. The fact that the testing dataset contains high variance data indicates that it will always end up with some errors. Increasing the k value with this trimmed dataset will not help as well as it does in the original and popular dataset. The minimum **RMSE value is 1.45131** at $k = 26$.

The ROC curve shows that the AUC of threshold 2.5 is 0.75000, the AUC of threshold 3 is 0.83838, the AUC of threshold 3.5 is 0.51948, the AUC of threshold 4 is 0.67949, and the largest AUC is 0.83838 with the corresponding threshold 3.

We also report the ROC curves of different threshold values: 2.5, 3, 3.5, and 4, of the original dataset, which is shown in Fig. 13.

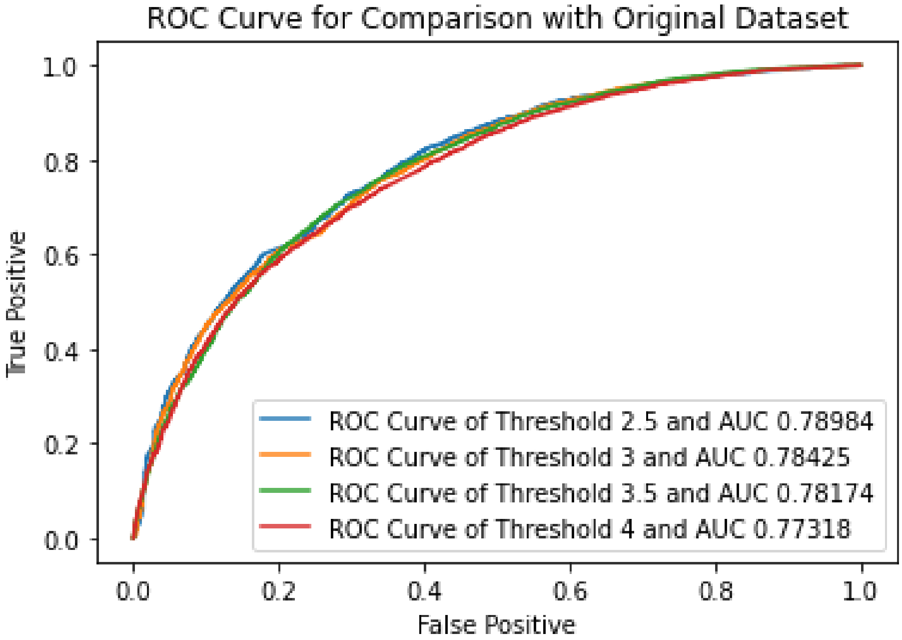

Figure 13: ROC Curve of of Original Dataset

The ROC curve shows that the AUC of threshold 2.5 is 0.78984, the AUC of threshold 3 is 0.78425, the AUC of threshold 3.5 is 0.78174, the AUC of threshold 4 is 0.77318, and the largest AUC is 0.78984 with the corresponding threshold 2.5. We see that as we go away from the mean threashold value of 2.5, AUC decreases monotonically.

Question 7

No, it is not a convex optimization problem as we expect the loss function to have local minima. For a fixed U, we can solve the problem using alternating least squares. The least-squares problem is given by:

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

The solution to this is given by:

$$V = U^T(UU^T)^{-1}R$$

where R is the rating matrix.

Question 8

Non-negative matrix factorization is a popular latent factor-based collaborative filtering algorithm that is used to estimate the missing entries of the rating matrix R. The main constraint here is that the latent factors U and V are non-negative. However, the main advantage is that the obtained factors U and V are highly interpretable.

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

subject to $U \geq 0, V \geq 0$.

Once we obtain the matrices U and V, we can estimate the rating of user i for an item j as:

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

Question 8 A

In this question we are asked to design an NMF based collaborative filter to predict the ratings of movies in the original dataset and report the average RMSE and MAE.

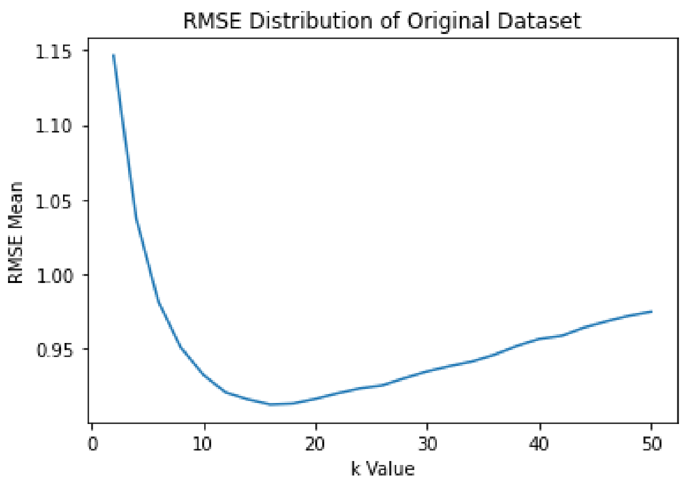

Figure 14: RMSE Distribution of Original Dataset

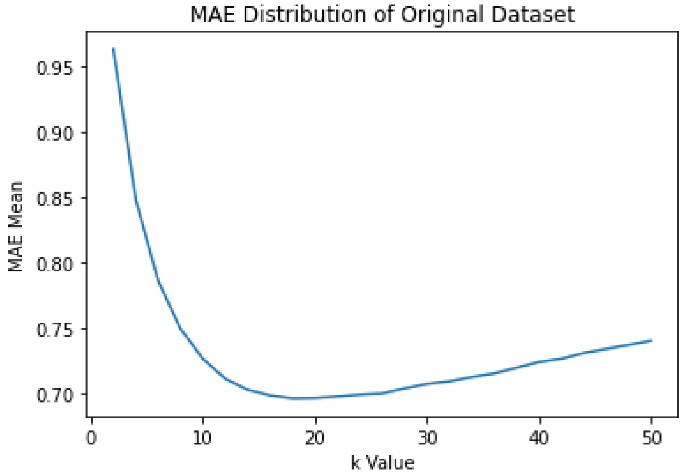

Figure 15: MAE Distribution of Original Dataset

From both figures, we can see a clear elbow. The errors decrease exponentially upto a certain value of k after which we see a linear increase. This is as expected as we have seen in project 1. This happens because initially, increase in k leads to more information being captured by the model leading to less error. However, at a certain k value, the curse of dimensionality comes into play, and the reduced matrix becomes sparse. Due to this, error increases.

Question 8 B

From the figures above, we see that the optimal k value occurs at $\mathbf{k = 16}$ for an RMSE value of $\mathbf{RMSE = 0.9127}$ and the optimal k value occurs at $\mathbf{k = 18}$ for an MAE value of $\mathbf{MAE = 0.6962}$.

There are 20 genres in the dataset(including "no genres listed") which is very close to either k value.

Question 8 C

Here, we first trimmed the data into three different categories, namely, popular, unpopular, and high variance datasets. On each trimmed dataset, we apply NMF CF to predict the ratings.

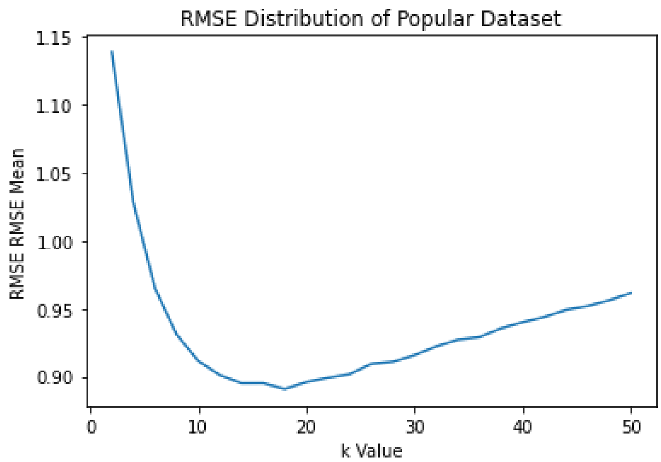

Figure 16: RMSE Distribution of Popular Dataset

The characteristics of the RMSE plot for the popular dataset is very similar to the original one. The minimum average **RMSE = 0.8909** for the optimal k value **k = 18**. Again, k is very close to the number of genres.

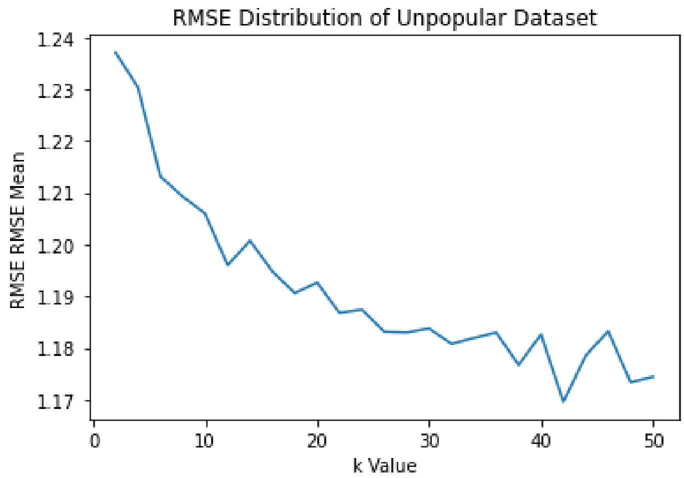

Figure 17: RMSE Distribution of Unpopular Dataset

The minimum average **RMSE = 1.1696** for the optimal k value **k = 42** on the unpopular dataset. Clearly, the error has increased compared to the error obtained on the popular dataset. This is as expected as now, the matrix is very sparse and so the model has a hard time predicting. Further, this is reflected in the optimal value of k being far off from the number of genres. Therefore, it is very clear that it is possible to predict ratings accurately if the matrix is dense rather than when the matrix is sparse.

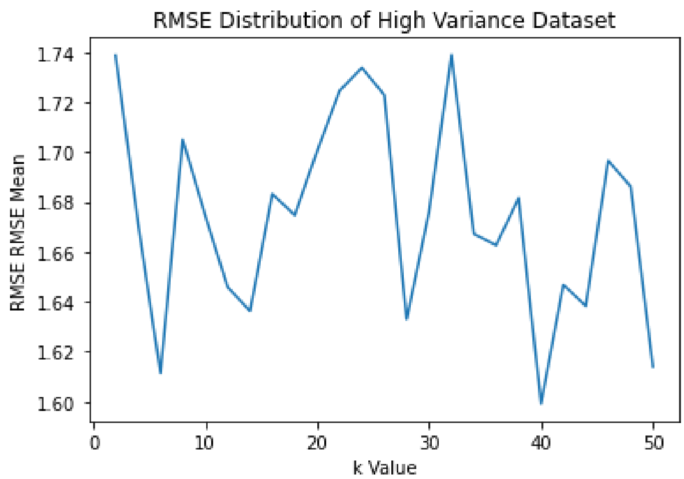

Figure 18: RMSE Distribution of High Variance Dataset

The minimum average **RMSE = 1.5992** for the value **k = 40** on the high variance dataset. This has the highest error of all three subsets. This can be explained by the fact that the ratings here vary wildly, and so the model has a tough time predicting any rating accurately. For example, if there are just 2 ratings for a movie, and they are 0.5 and 4, it would not be possible for even a human to predict the actual rating accurately.

ROC CURVES

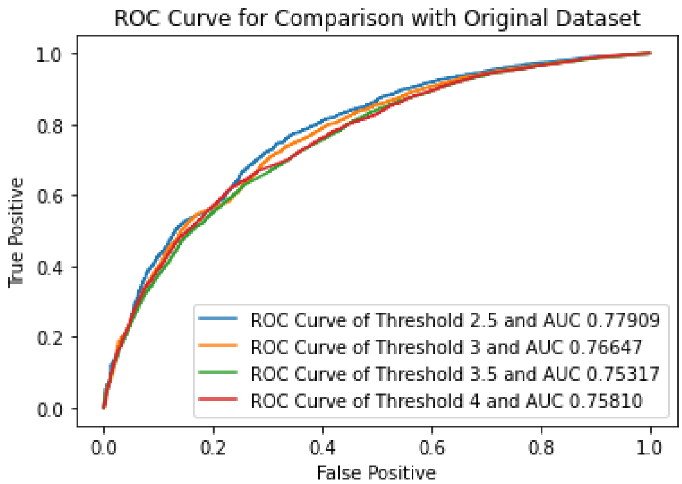

Figure 19: ROC Curves for the specified thresholds with k = 16

The ROC curves for the threshold values of [2.5, 3.0, 3.5, 4.0] with the optimal latent factor value k = 16 is shown above. We observe that as k moves away from the mean value of 2.5, AUC generally decreases.

Threshold	**AUC**
2.5	0.77909
3.0	0.76647
3.5	0.75317
4.0	0.75810

Table 1: AUC values for given thresholds

Question 9

In this question, we are asked to find any patterns found in the movie genres for the latent vectors. It does seem like each vector encodes data from a small set out of the 19 available genres(20 including "no genres listed").
For example, the genres of the top 10 movies for column vector 1 is as follows:

Column: 1

Children—Comedy
Comedy
Comedy—Drama—Romance
Comedy
Adventure—Drama—Mystery
Action—Documentary—Drama—Thriller
Action—Comedy
Western
Comedy—Romance
Drama

We see that column 1 mostly encodes movies that belong to the comedy genre and a bit of action. As another example, consider column 8:

Column: 8

Drama
Drama—Romance
Action—Drama
Action—Crime—Drama—Thriller
Drama—Romance
Action—Thriller
Comedy—Crime—Drama
Drama
Drama
Adventure—Drama

This column seems to purely encode the Drama genre.Therefore, there is clearly a "preference" shown by the latent vectors.

Question 10

Matrix factorization (MF) is a method for predicting the ratings that a user will give to an item. It can include a bias term to account for the biases that individuals have (for example, if somebody rates all movies highly), and for biases that items have (for example, if a movie is rated very highly by everybody.

The optimization formula for the MF is given by the following equation:

$$\min_{U,V,B_u,B_i} \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(r_{ij} - \hat{r}_{ij})^2 + \lambda||U||_F^2 + \lambda||V||_F^2 + \lambda\sum_{u=1}^{m} b_u^2 + \lambda\sum_{i=1}^{n} b_i^2$$

After this minimization problem is solved, we can use the values for predicting the ratings. The predicted rating given by user i for item j, denoted as $\hat{r}_{ij}$ is given by :

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

In the above equation, μ is the mean of all ratings, b_i is the bias of user i, and b_j is the bias of item j.

Question 10 A

In this question we are asked to design an MF based collaborative filter with bias to predict the ratings of movies in the original dataset and report the average RMSE and MAE.

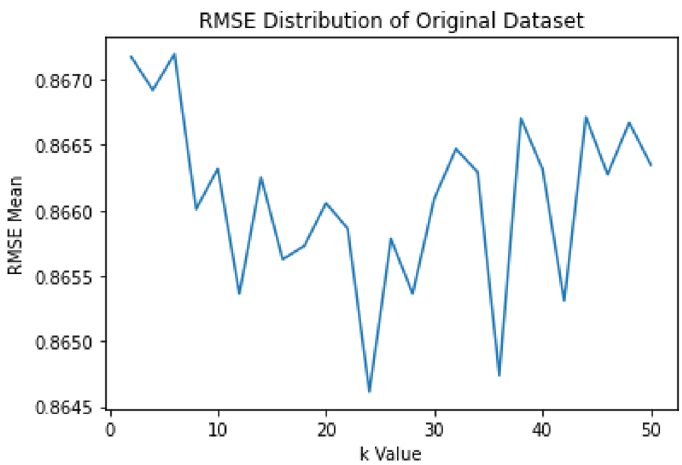

Figure 20: RMSE Distribution of Original Dataset

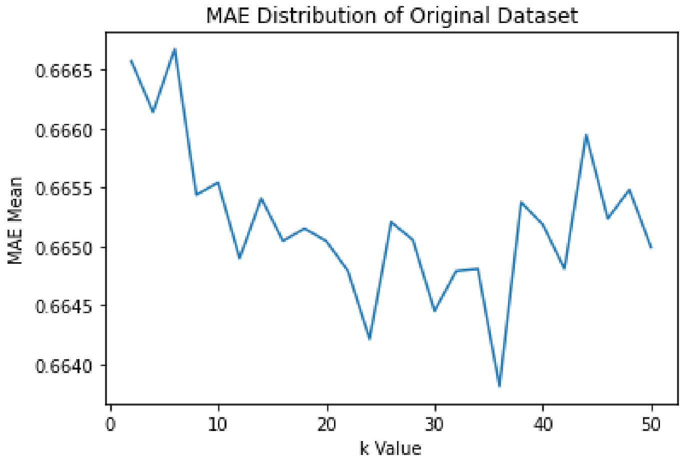

Figure 21: MAE Distribution of Original Dataset

The main difference between the graphs for MF and NMF is that the range of values are different. The NMF values lie in the broad range of 1.15 and 0.9 but the MF with bias values lie in the smaller range of 0.864 to 0.868. Also, we observe that the minimum average RMSE value of **0.8646** for **k = 24** and the minimum average MAE value of **0.6638** for **k = 36** are smaller than the values obtained for both NMF and kNN. This is because SVD does not have the positive values constraint that is applied for NMF and is thus more flexible. However, because the values are not positive, interpretation is harder than that of NMF.

Question 10 B

From the figures above, we see that the optimal k value occurs at **k = 24** for an RMSE value of **RMSE = 0.8646** and the optimal k value occurs at **k = 36** for an MAE value of **MAE = 0.6638**.

There are 20 genres in the dataset which is very close to the k value for RMSE.

Question 10 C

Here, we first trimmed the data into three different categories, namely, popular, unpopular, and high variance datasets. On each trimmed dataset, we apply MF with bias CF to predict the ratings.

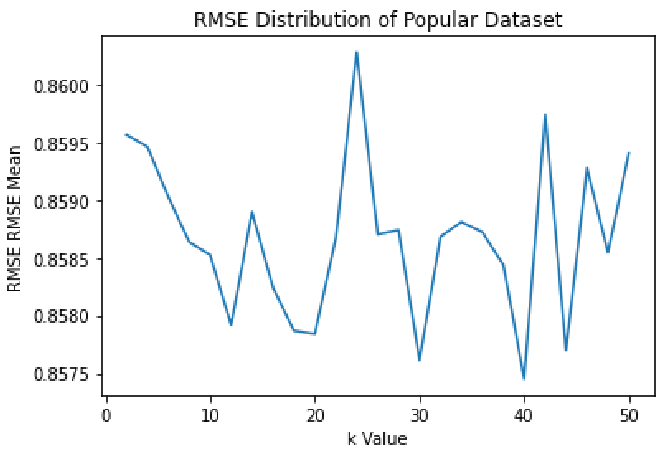

Figure 22: RMSE Distribution of Popular Dataset

The characteristics of the RMSE plot for the popular dataset is very similar to the original one. The minimum average **RMSE = 0.8574** for the optimal k value **k = 40**. Note that the difference in RMSE for k = 20 (equal to number of genres) and the optimal k = 40 here is in the 3rd decimal and is very small.

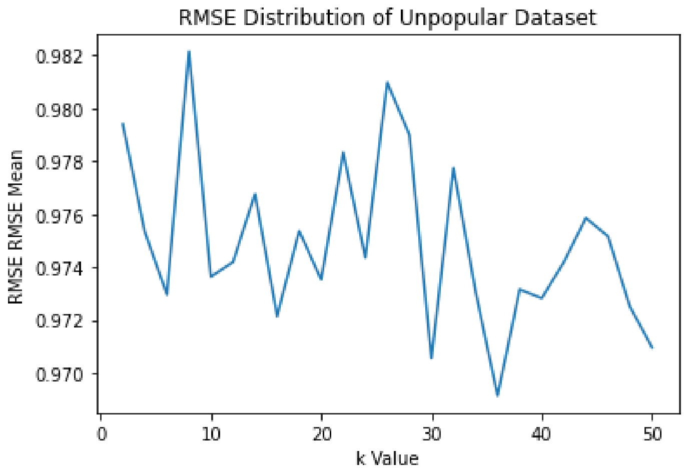

Figure 23: RMSE Distribution of Unpopular Dataset

The minimum average **RMSE = 0.9692** for the optimal k value **k = 36** on the unpopular dataset. Clearly, the error has increased compared to the error obtained on the popular dataset. This is as expected as now, the matrix is very sparse and so the model has a hard time predicting. Further, this is reflected in the optimal value of k being far off from the number of genres. Therefore, it is very clear that it is possible to predict ratings accurately if the matrix is dense rather than when the matrix is sparse.

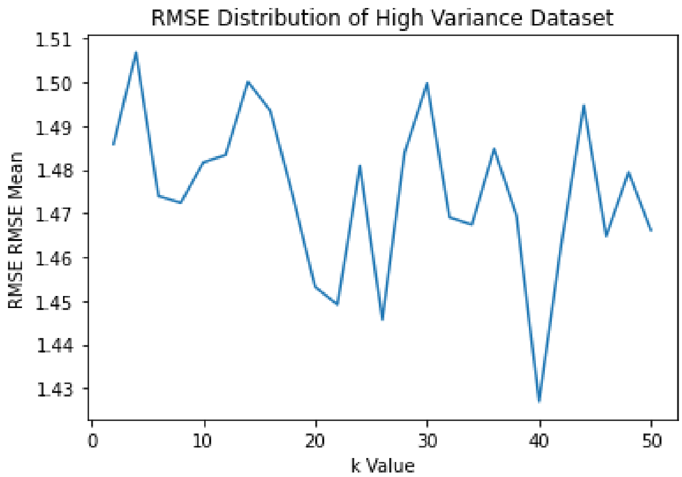

Figure 24: RMSE Distribution of High Variance Dataset

The minimum average **RMSE = 1.4269** for the value **k = 40** on the high variance dataset. This has the highest error of all three subsets. This can be explained by the fact that the ratings here vary wildly, and so the model has a tough time predicting any rating accurately. For example, if there are just 2 ratings for a movie, and they are 0.5 and 4, it would not be possible for even a human to predict the actual rating accurately.

ROC CURVES

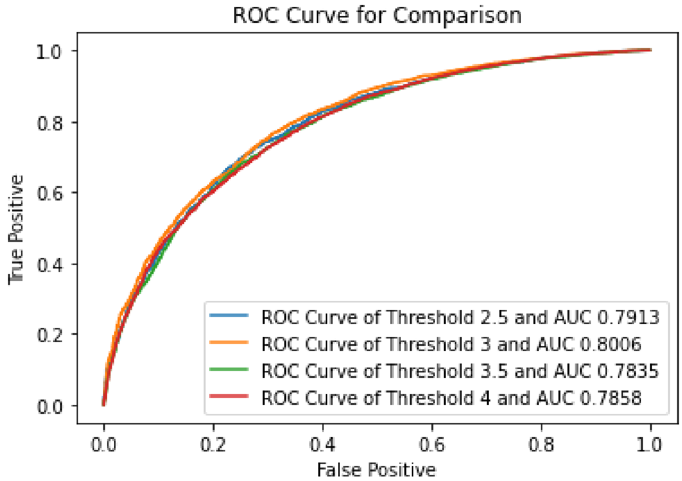

Figure 25: ROC Curves for the specified thresholds with k = 24

The ROC curves for the threshold values of [2.5, 3.0, 3.5, 4.0] with the optimal latent factor value k = 24 is shown above.

Threshold	**AUC**
2.5	0.7913
3.0	0.8006
3.5	0.7835
4.0	0.7858

Table 2: AUC values for given thresholds

Question 11

The naive collaborative filter is a very simplistic model that simply predicts the rating of an unrated movie as the mean rating given by that user. Although this is a flawed idea, it is beneficial to a certain degree as there is no training required.

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

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

We obtained the average **RMSE = 0.9347** across all 10 folds.

On the trimmed datasets, the average RMSE we obtained are as follows:

Dataset	**RMSE**
Popular	0.9323
Unpopular	0.9712
High Variance	1.4572

Table 3: Average RMSE values obtained for naive CF on the trimmed datasets

Again, we see the general trend that the RMSE value on the popular dataset is lower than the RMSE value on the full test data. This is because the movies are rated consistently and the model has enough information to make informed decisions. Also, as expected, we see that the RMSE values for both the unpopular and high variance datasets are much larger for the same reasons as described before.

Question 12

The ROC curves that we obtained for the optimal k values for each model is shown below:

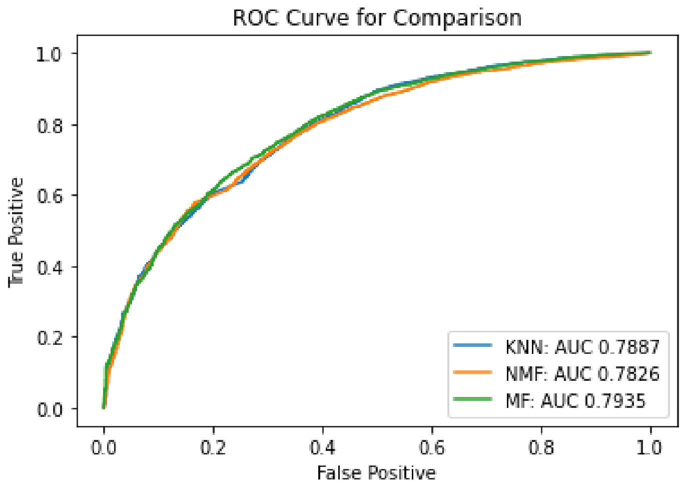

Figure 26: ROC Curves for comparison

The k value and AUC for each model is shown in the following table:

Model	**Optimal k**	**AUC**
KNN	20	0.7887
NMF	16	0.7826
MF	24	0.7935

Table 4: AUC values obtained for optimal k value for each of our models

We see that the optimal MF with bias collaborative filter outperforms its counterparts. We explain this by the following reasons:

1. MF outperforms NMF because the NMF filter has the constraint that all values must be positive. Due to this, it is less flexible. However, the NMF model is much more interpretable as seen earlier.

2. MF outperforms kNN because the kNN simply performs the closest distance algorithm and does not make use of data-specific information. Also, as k increases in the kNN, the sense of distance is lost, leading to the curse of dimensionality.

3. MF filter captures the idea of a user bias and so is well equipped to deal with anomalous users.

Question 13

$$Precision(t) = \frac{|S(t) \cap G|}{|S(t)|} \tag{1}$$

$$Recall(t) = \frac{|S(t) \cup G|}{|G|} \tag{2}$$

Precision is the rate of accuracy of the recommendation system. Equation 1 states that the precision is defined to be the ratio of the correct recommendations over total of all recommendations. This means that it is the proportion of recommendations that are correct and were liked by the user.

Recall, according to Equation 2, is the ratio of recommended items which were liked by the user, over the total of all items which were liked by the user. This means that it is the proportion of the liked items of a user which were actually recommended by the system.

Question 14

If a model performs well in prediction, it should obtain a high precision value while maintaining a low recall value, hence we need to take a look into the value of precision, recall and also the correlation between precision and recall.

From Fig. 27 we can obtain precision scores, recall scores and the correlative score of precision and recall of the **k-NN** model. It is observed from the precision plot that as t increments, the precision scores decreases. This outcome is expected because as the size t increases, it is more difficult to maintain the relatively high precision as obtained in a smaller size batch, so the precision score will drop slightly but the overall score still stays above 85%. the recall score, on the other hand, increases as the size t increases. The recall score increases because intuitively, it is easier to guess a batch of favorite movies of a user than to guess one single one. Therefore, the incremental trend makes sense. The correlative score of precision and recall has a decreasing trend, in that as we stated earlier, if a model performs well in prediction, it should obtain a high precision value while maintaining a low recall value, so when precision drops, recall score rises.

From Fig. 28 we can obtain precision scores, recall scores and the correlative score of precision and recall of the **NMF** model. It is observed from the precision plot that as t increments, the precision scores decreases. This outcome is expected because as the size t increases, it is more difficult to maintain the relatively high precision as obtained in a smaller size batch, so the precision score will drop slightly but the overall score still stays above 85%. the recall score, on the other hand, increases as the size t increases. The recall score increases because intuitively, it is easier to guess a batch of favorite movies of a user than to guess one single one. Therefore, the incremental trend makes sense. The correlative score of precision and recall has a decreasing trend, in that as we stated earlier, if a model performs well in prediction, it should obtain a high precision value while maintaining a low recall value, so when precision drops, recall score rises.

From Fig. 29 we can obtain precision scores, recall scores and the correlative score of precision and recall of the **MF** model. It is observed from the precision plot that as t increments, the precision scores decreases. This outcome is expected because as the size t increases, it is more difficult to maintain the relatively high precision as obtained in a smaller size batch, so the precision score will drop slightly but the overall score still stays above 85%. the recall score, on the other hand, increases as the size t increases. The recall score increases because intuitively, it is easier to guess a batch of favorite movies of a user than to guess one single one. Therefore, the incremental trend makes sense. The correlative score of precision and recall has a decreasing trend, in that as we stated earlier, if a model performs well in prediction, it should obtain a high precision value while maintaining a low recall value, so when precision drops, recall score rises.

The conclusion of the comparison plot is placed after Fig. 30.

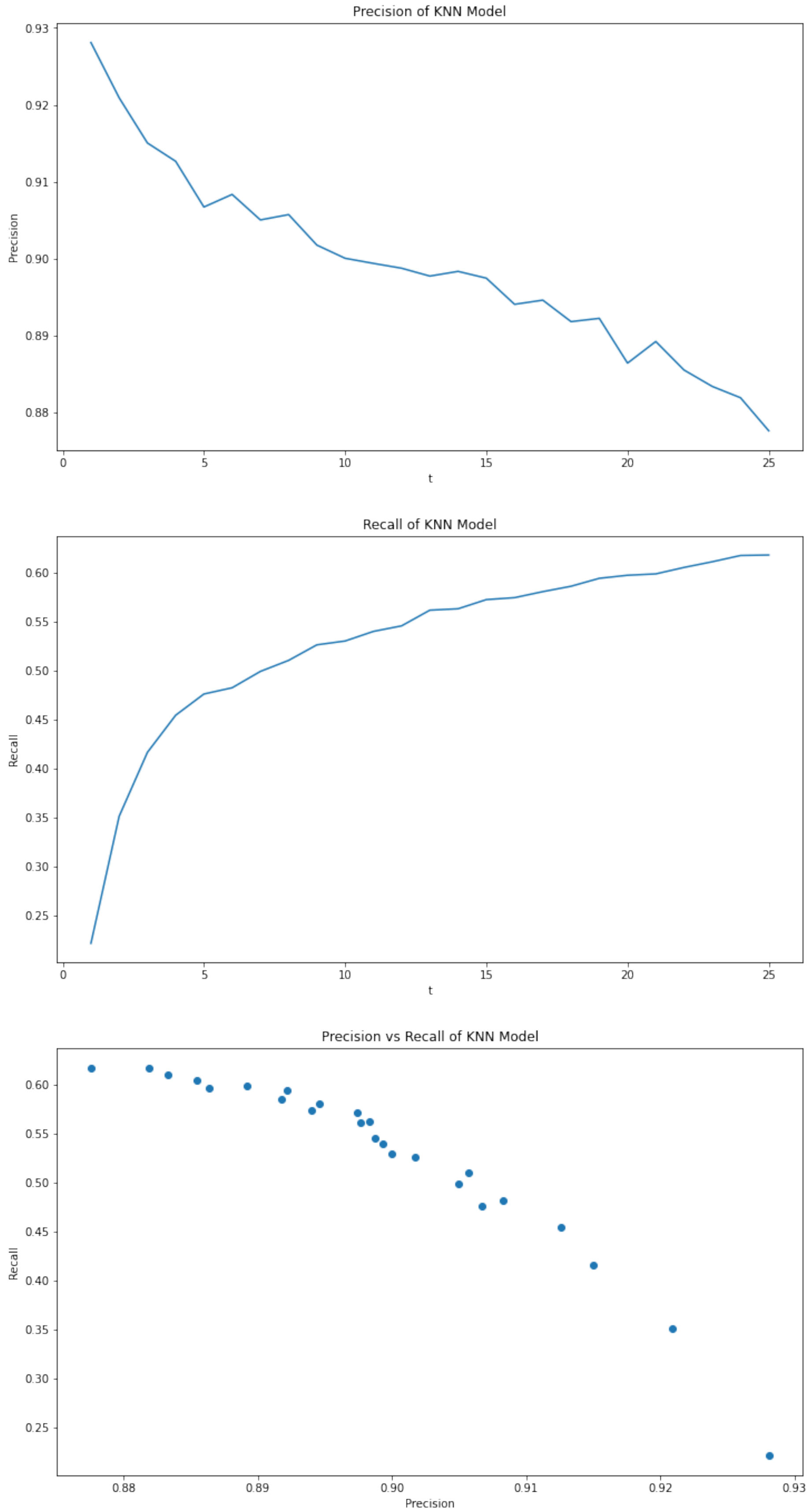

Figure 27: Precision, Recall, and Their Correlation of the k-NN Model

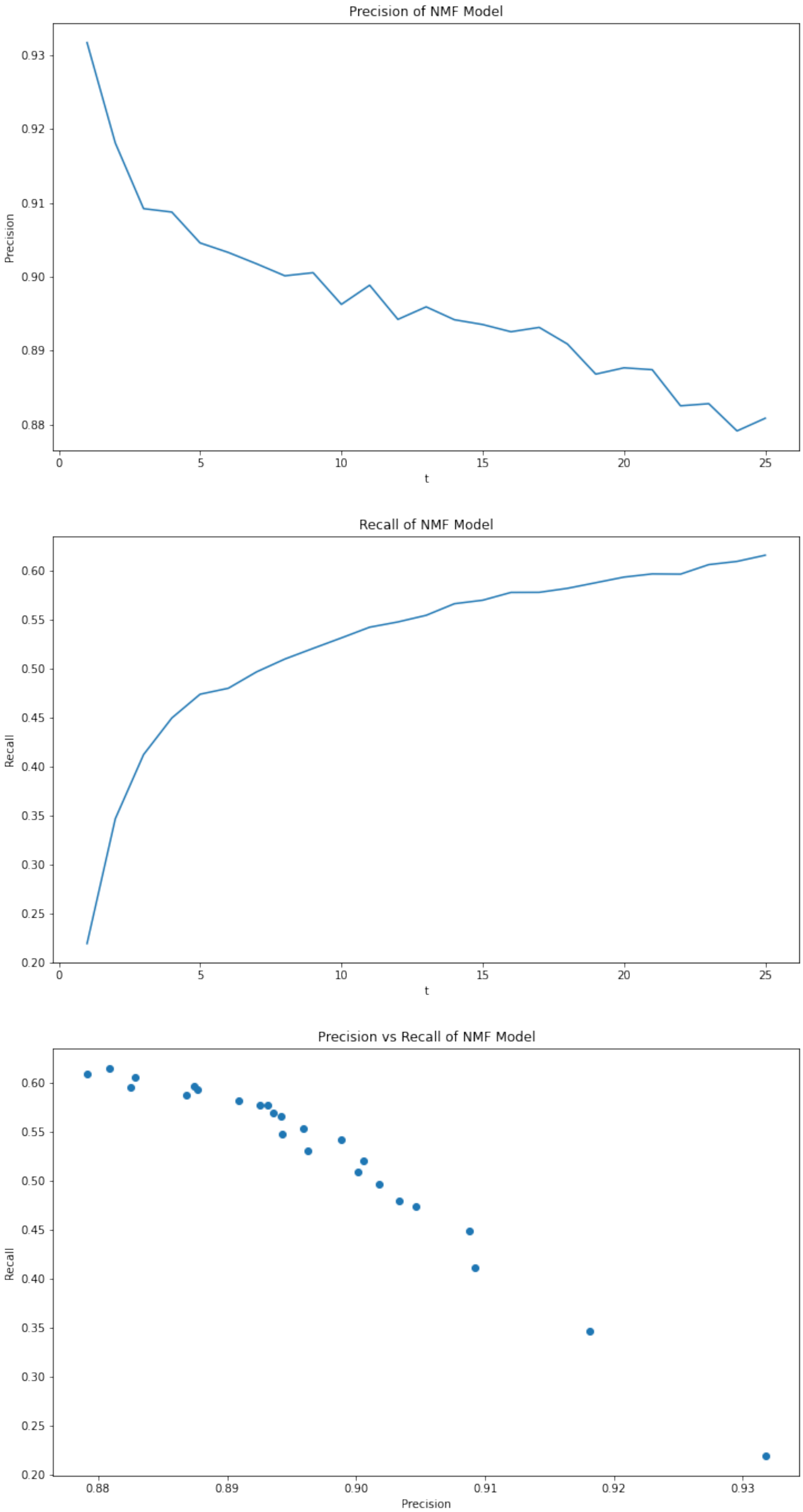

Figure 28: Precision, Recall, and Their Correlation of the NMF Model

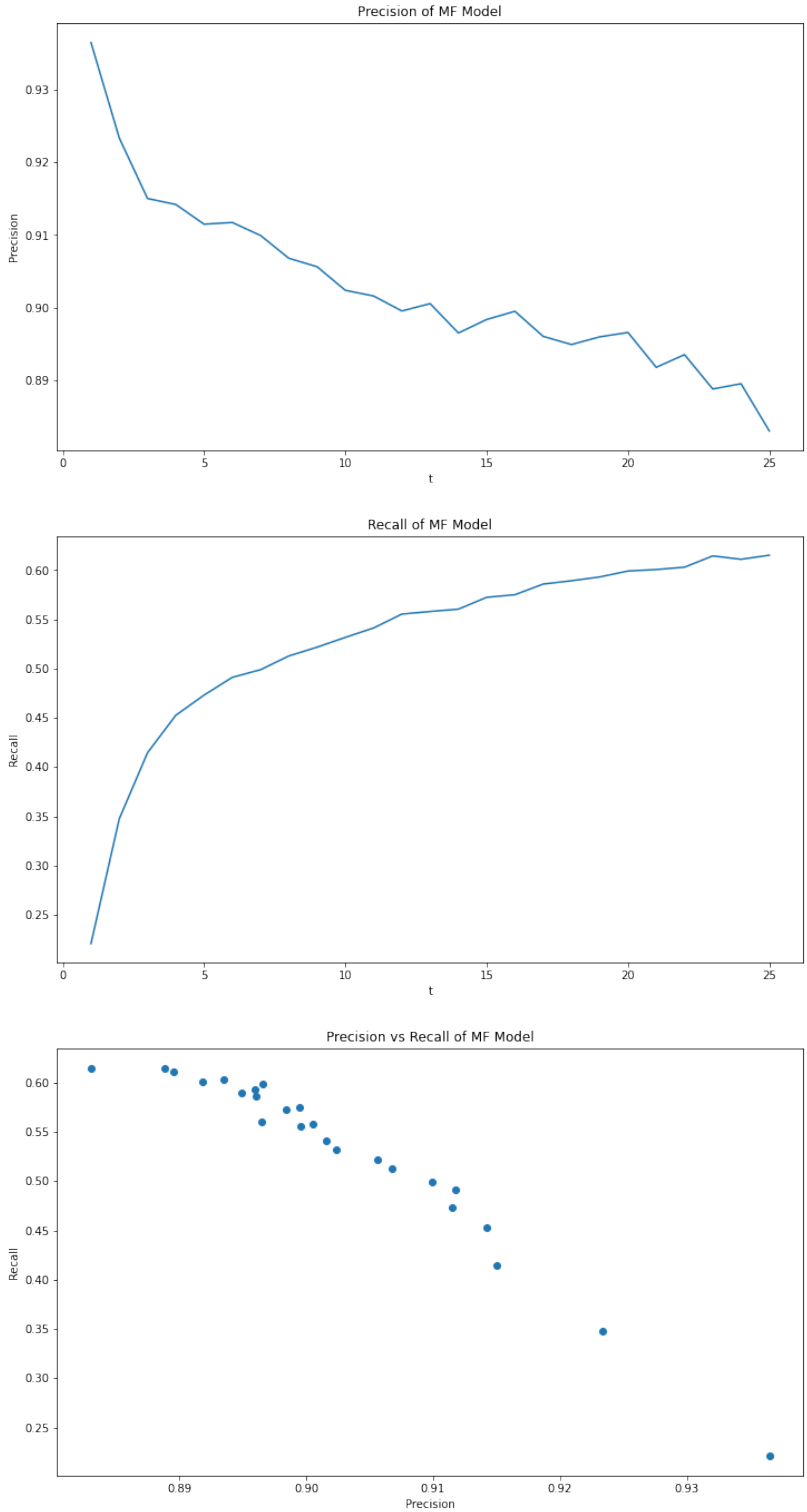

Figure 29: Precision, Recall, and Their Correlation of the MF Model

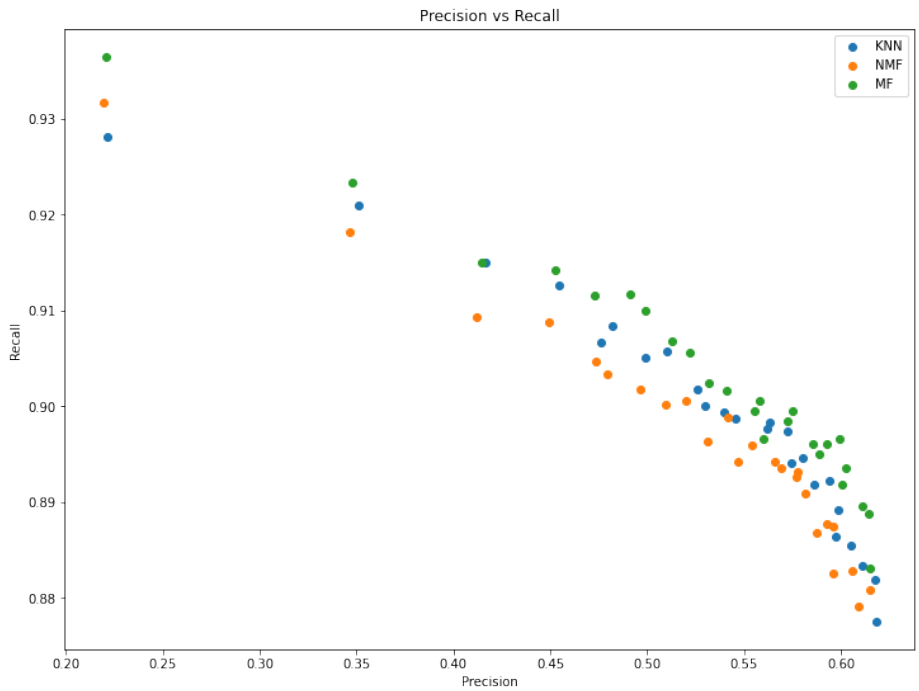

Figure 30: Precision vs Recall of k-NN, NMF, and MF Models

It is obtained from the result that MF filter turned out to have a higher recall score compared with the other two. k-NN outperformed NMF and this result conforms with the one we obtained from the ROC curve comparison.