Project 3 : Recommender Systems

Sidarth Srinivasan (UID - 005629203), Shweta Katti (UID - 505604846)

March 1, 2022

Course : ECE 219 Large Scale Data Mining
Term : Winter 2022

A recommender system is used to predict the rating or preference of a given user. For example, recommender systems are widely used to generate playlists on spotify, to recommend movies on netflix, or to recommend potential products to buy on amazon.

There are two ways to implement these recommender systems:

- Collaborative Filtering
- Content-based Filtering

Or, we also use a combination of both of the above filtering techniques.

In this project, we will be exploring Collaborative Filtering methods to build a recommender system for a **Synthetic Movie Lens Dataset**. This dataset contains 100836 ratings and 3683 tag applications across 9742 movies. The underlying rating matrix in this dataset is a sparse matrix which is a main challenge for designing collaborative filtering models. Collaborative filtering is a method that filters the preference of a given user by collecting preferences or ratings from several other users.

The two major types of Collaborative Filtering methods are:

- **Neighbourhood-based Collaborative Filtering :** We further classify this method into *item-based models* and *user-based models*. In this project, we implement the *user-based method*. To determine the neighbourhood of a given user, we need to first find the similarity between the ratings of the users, which is done by constructing a similarity function using Pearson-correlation Coefficient. We then implement KNN method to define the neighborhood of the users.
- **Model-based Collaborative Filtering :** In this approach, we deploy ML algorithms to predict the ratings of users for unrated items. In this project, we implement latent factor based models. Here, we implement *NMF filter*, *MF with bias filter*, and *Naive filter*.

To evaluate the performance of these filter models, we use a 10 fold cross validation. We then report the RMSE and MAE values across all these 10 folds and reprt the minimum average RMSE. ROC curves are also plotted to visualize the performance of the various models.

1 Question 1

1.a

The Sparsity is calculated as follows :

$$\text{Sparsity} = \frac{Total number of available ratings}{Total number of possible ratings} = 0.016999683055613623$$

We note that the sparsity is very low, which means that majority of the movie ratings are missing, which makes sense as there is a large number of movies when compared to the number of users and not all users have watched majority of the movies and hence have not rated them. This means that the rating matrix R is sparse in nature.

1.b

Figure 1 below shows the histogram of rating values.

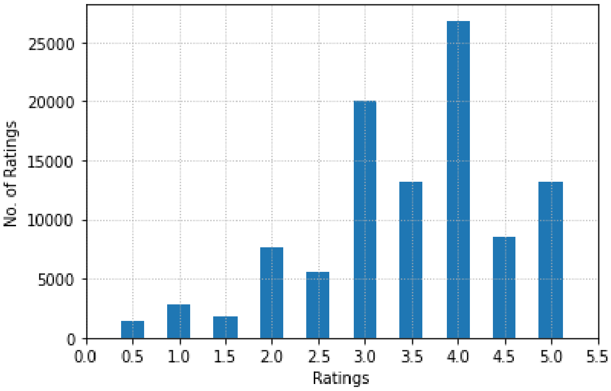


Figure 1: Histogram showing the frequency of the rating values:

From the histogram, we can infer that majority of ratings lie between a 3.0 to a 5.0. There are very few movie ratings that lie below 3.0 as the distribution is more concentrated towards the right. This high distribution for movie rating 3.0 could be attributed to the bias or apriori knowledge among users. This could arise from users perusing through popular movie rating sources and consider those ratings before choosing a movie they would like to watch. Thus, it could be that users would watch movies that they think would like based on the rating sources and hence end up liking most of the times thus explaining the skewed distribution.

We can also see that integer values of ratings have a higher frequency compared to the decimal values. This could be because most people generally rate in whole numbers as it is the most common psychological and basic way of thinking.

1.c

Figure 2 below shows the distribution of the number of ratings received among movies. We observe that the curve is monotonically decreasing as 500 movies (approx) received more than 50 unique user ratings thus explaining the sparsity of the ratings matrix.

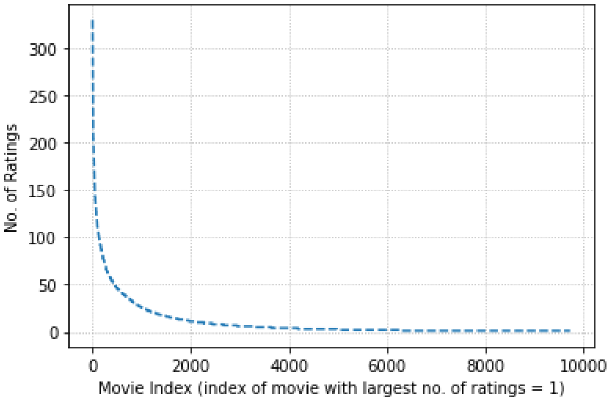


Figure 2: Distribution of the number of ratings received among movies

1.d

Figure 3 below shows the distribution of ratings among users. We also observe a monotonically decreasing trend here with around 50 users (approx) out of the total 610 users to have given ratings to 500 movies or more. Both Figures 2 and 3 explains how the ratings is sparse.

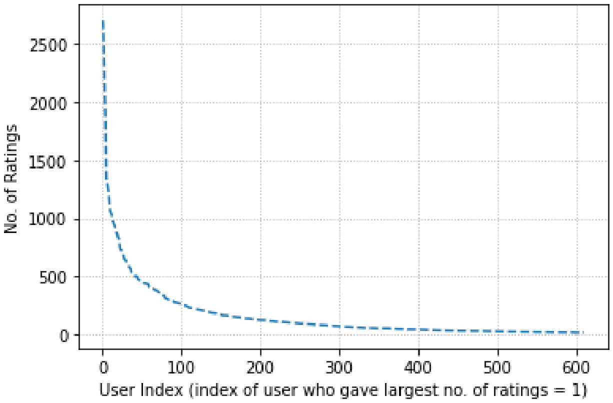


Figure 3: Distribution of ratings among users

1.e Salient features of the distributions:

Both Figure 2 and Figure 3 are monotonically decreasing in nature. In Figure 2, we can observe that few movies (approximately 500) possess more than 50 unique ratings. This denotes that they are popular movies and thus users tend to watch movies that are highly rated and popular and thus leading to more user ratings for the popular movies. From figure 3, we can infer that there are few users (approx 50) that have watched and rated 500 movies or more. Both these figures explain the sparsity of the rating matrix R. This is a serious challenge during prediction as data moves into higher dimensions, it causes the data to be sparse in each dimension due to the volume increase also known as the Curse of Dimensionality. With most of the elements in the representations being 0, thereby contributing to no information during the model training, we observe a poorly trained model on the representation with just too many parameters. This performs poorly as too many parameters overfits the model with just a few movies that posses ratings which will be explored in the coming sections.

1.f

Figure 4 below shows the variance of the rating values. We can see that most movies have a very low variance in their ratings. This makes sense as users tend to watch movies after looking at existing reviews. There are very few movies which receive a variance of 2.5 or greater in their ratings. Since the ratings of a movie do not vary significantly, we can hence use similarity property to predict missing rating and provide better accuracy for user based collaborative filtering.

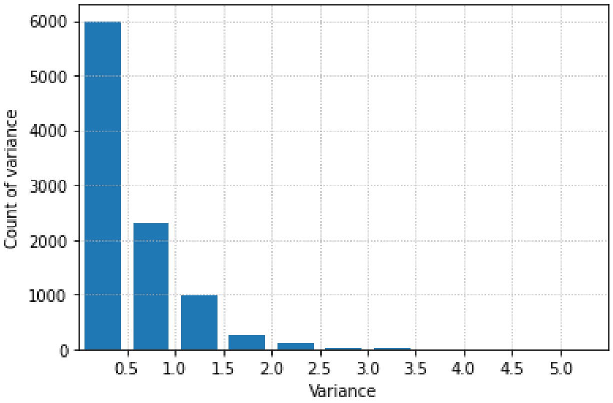


Figure 4: Variance of the rating values

2 Question 2

2.a

The mean rating of a user u can be given by the equation given below:

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

2.b

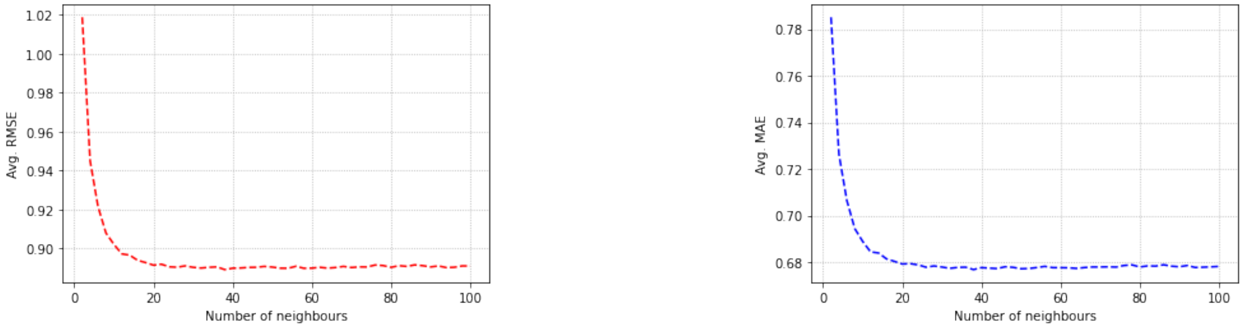
- $I_u \cap I_v$ represents the set of movies that have been rated by both user u and user v.
- This given set can be equal to a null set, i.e $I_u \cap I_v = \emptyset$ as there may be no common movies that have been rated by both the users.

3 Question 3

By mean-centering the raw ratings $(r_{vj}\mu_v)$ in the prediction function we are trying to normalize the user bias. Rating preferences might differ among users, for example, a user might rate a movie they liked by giving it a 5 while another user might rate a movie they liked by giving it a 4. Hence, users who give a movie same rating, might not like or dislike the given movie at the same level and hence, normalization is necessary for prediction. In simpler terms, by mean centering we normalize the scale of ratings among different users. Also, mean centering reduces the variances due to polar ratings by a few users. Overall, it helps us making the data less noisy.

4 Question 4

Figure 5 below shows the plot of average RMSE and MAE for various number of neighbours with Pearson similarity metric. From the figure, we can understand the prediction error is monotonically decreasing until a point after which it the prediction error plateaus as k increases and we see no significant decrease in the prediction error. The same trend is observed for both the RSME and MAE plots with minimum prediction error for different values of k.



(a) Average RMSE against k

(b) Average MAE against k

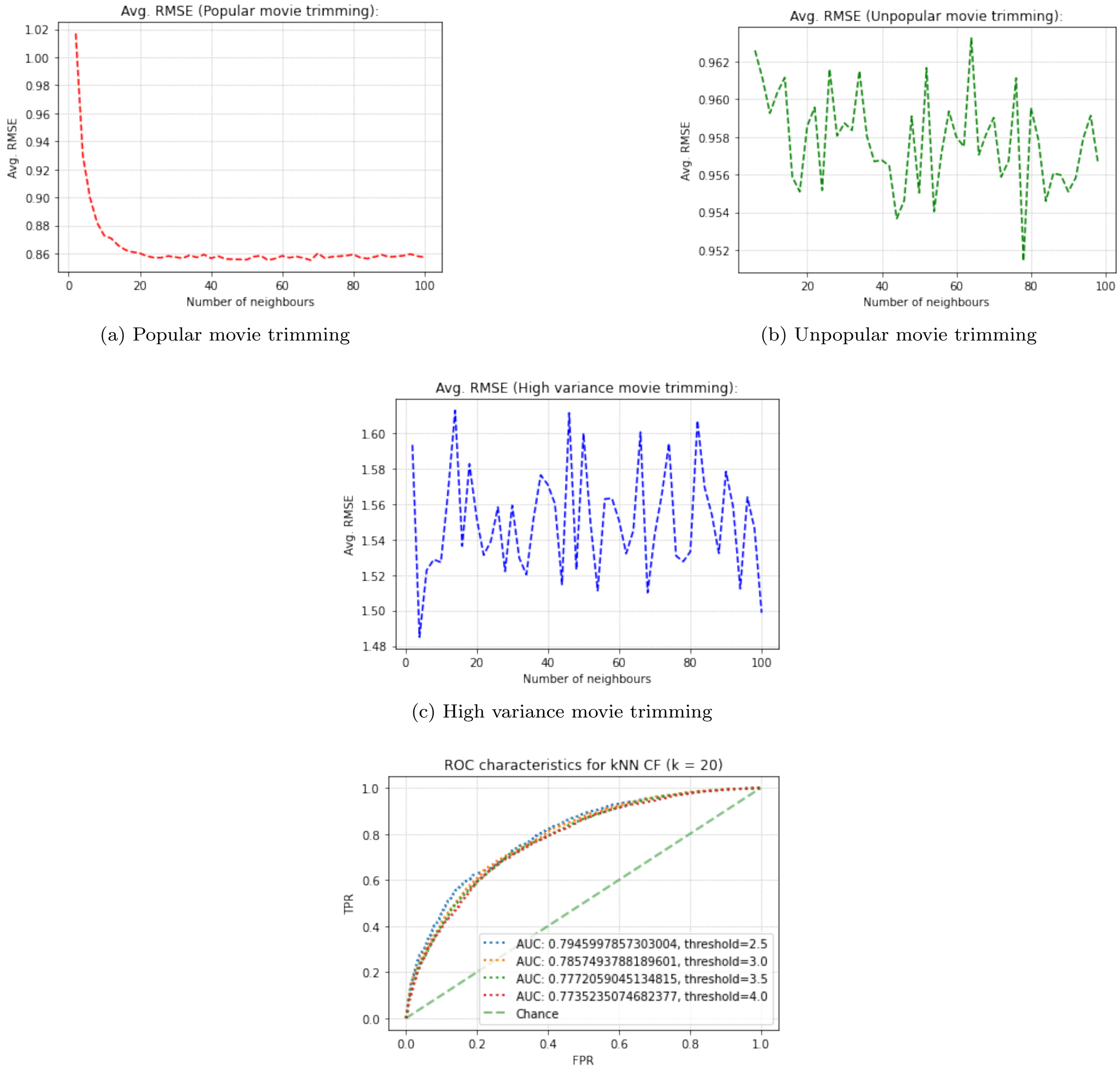
Figure 5: Average RMSE and MAE Plots

5 Question 5

From the result obtained from question 4, we infer that the steady state value or the minimum error occurs for k = 20. The error values are mentioned as follows :
Average RMSE = 0.8890655422448244
Average MAE = 0.6771145304542264

6 Question 6

The following plots as shown in Figure 6 were obtained for kNN collaborative filtering for three trimming models.



(a) Popular movie trimming

(b) Unpopular movie trimming

(c) High variance movie trimming

(d) ROC Curves

Figure 6: Filter performance on trimmed test sets (Average RMSE plots)

- For POPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.8554**. We can note that popular trimmed set performed better than the original set. We can also see from the Figure 6.a that the average RMSE value decreases with increase in k and plateaus beyond a point where there is no significant dip in the error even when k increases.
- For UNPOPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.9528**. This is much higher than that of popular trimmed dataset and the untrimmed set. From figure 6.b we can see that the average RMSE does not follow the general monotonically decreasing trend and hence is non monotonic and erratic. This erratic nature is due to the large number of outliers present. In this case, improving k does not necessarily improve the performance as the predictor does not have enough users for predicting the correct rating for rare items.
- For HIGH VARIANCE TRIMMING: **Minimum avg. RMSE** was found to be **1.4807**.This is way higher than both popular and unpopular sets and is thus, the worst in performance. On top of being non-monotonic, it is also extremely erratic in nature as seen in figure 6c. Since we use movies with high variance of ratings, the predictor becomes extremely sensitive to outliers and hence results in large prediction error.

Figure 6d shows the **ROC curves** for all the three models. The AUC obtained for each threshold is as follows:

- Threshold = 2.5, AUC = 0.7946
- Threshold = 3.0, AUC = 0.7857
- Threshold = 3.5, AUC = 0.7772
- Threshold = 4.0, AUC = 0.7735

We get highest AUC for a threshold of 2.5. The values of AUC reduces with increase in threshold value.

7 Question 7

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

No, the optimization problem denoted by the above equation is not convex. We are dealing with both matrices U and V at the same time as unknown variables. On the other hand, however, if we fix one of these two variables and solve for the other one, the given optimization problem can be formulated to a least-squares problem which will be convex. Hence, the optimization problem is not jointly convex for both U and V due to the existence of multiple local minima in the objective function gradient plane.

Specifically, if we fix the matrix U and solve for the matrix V, we will have the following least-squares problem:

$$\underset{V}{\text{minimize}} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(r_{ij} - (\bar{U}V)_{ij}^T)^2$$

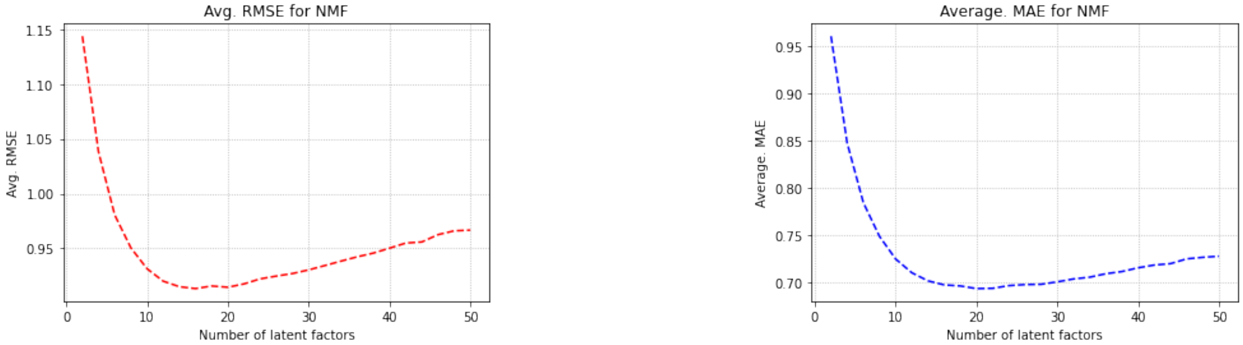
8 Question 8

Figure 5 below shows the RMSE and MAE plots against k for the NMF Colloborative Filter model. Both plots are similar in the sense that the error first decreases with an increase in the number of latent factors. But after a point, it starts increasing linearly again. But increasing the value of k, does not significantly help with the performance due to the curse of dimensionality. As increasing k, results in an increase in the dimension and the rating matrix becomes more noisy.

We obtained the following results:

- Minimum average RMSE: **0.913185**, value of k: **16**
- Minimum average MAE: **0.693954**, value of k: **20**

There are 19 genres in the MovieLens Dataset and the optimal value of k that we obtain is very close to this number. Also, if we take the optimal values of k for both RMSE and MAE, we obtain k = 18 which is close to the number of genres.



(a) Average RMSE against k

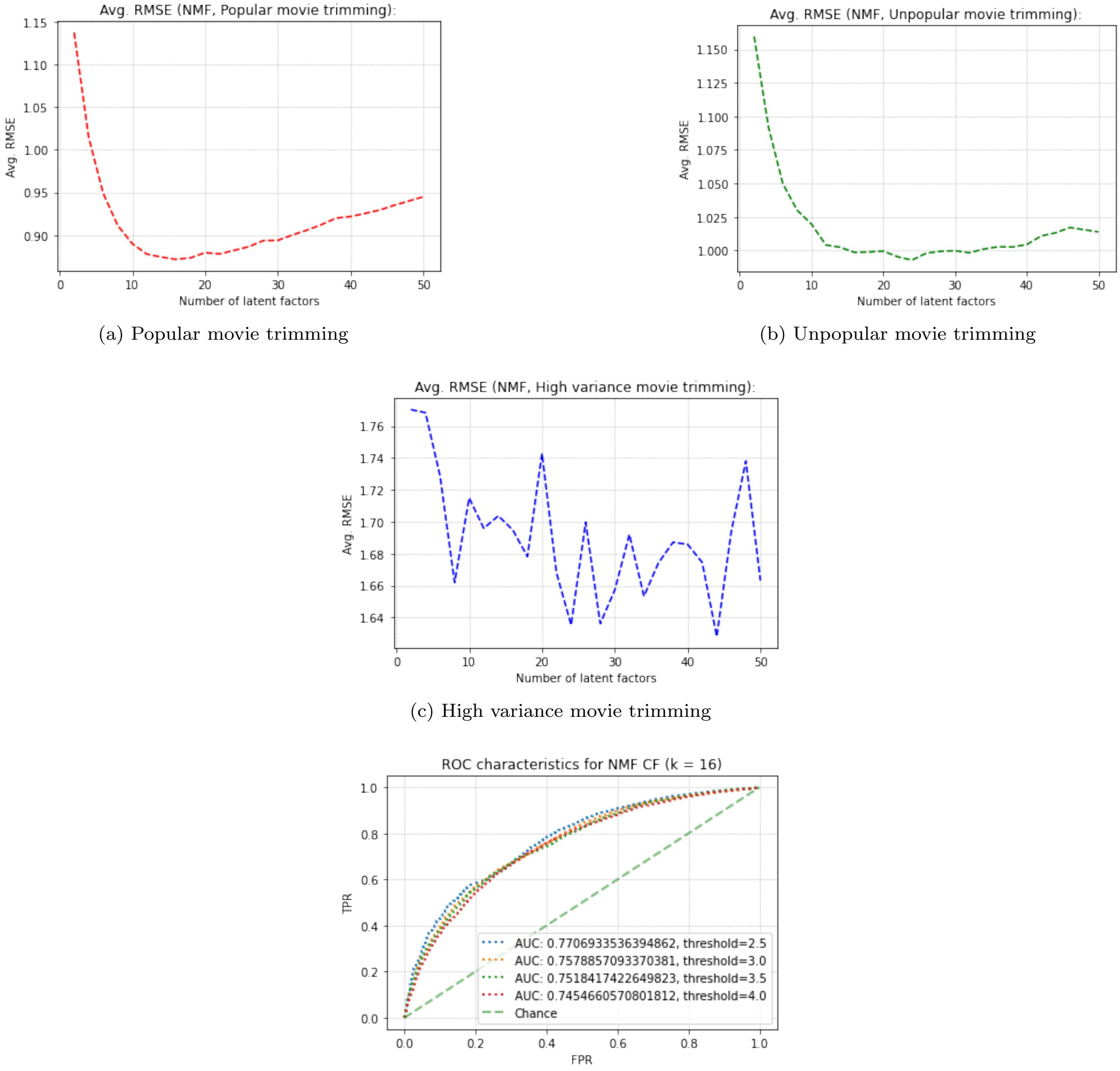
(b) Average MAE against k

Figure 7: Average RMSE and MAE Plots for NMF

The following plots as shown in Figure 8 were obtained for NMF collaborative filtering for three trimming models.

- For POPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.8714**. We can note that popular trimmed set performed better than the original set. We can also see from the Figure 8.a that the average RMSE value initially decreases and then increases with increase in k. This could be attributed to the increases in number of principal components is more than the semantic and complex information contained in the data. For smaller latent factors, we see a dip in the prediction error but as k increases, we observe that the prediction error increases again due to the reason above. Also since NMF allows only positive entries, therefore the factorization matrix suffers from high information loss and also causes the depth of NMF to diminish in higher dimensions.
- For UNPOPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.9927**. This is much higher than that of popular trimmed dataset. From figure 8.b we can see that the average RMSE does not follow the general monotonically decreasing trend and is fluctuating slightly. This eratic nature is due to the large number of outliers present. In this case, improving k does not necessarily improve the performance as the predictor does not have enough users for predicting the correct rating for rare items.

- For HIGH VARIANCE TRIMMING: **Minimum avg. RMSE** was found to be **1.6281**.This is way higher than both popular and unpopular sets and is thus, the worst in performance. On top of being non-monotonic, it is also extremely erratic in nature as seen in figure 6c. Since we use movies with high variance of ratings, the predictor becomes extremely sensitive to outliers and hence results in large prediction error.



(a) Popular movie trimming

(b) Unpopular movie trimming

(c) High variance movie trimming

(d) ROC Curves

Figure 8: Filter performance on trimmed test sets (Average RMSE plots)

Figure 8d shows the **ROC curves** for all the three models. The AUC obtained for each threshold is as follows:

- Threshold = 2.5, AUC = 0.7706

- Threshold = 3.0, AUC = 0.7578
- Threshold = 3.5, AUC = 0.7518
- Threshold = 4.0, AUC = 0.7455

We get highest AUC for a threshold of 2.5. The values of AUC decreases monotonically for the NNMF Collaborative Filter model.

9 Question 9

Here, we explore the interpretation of the NMF model by finding a relation between the movie genres and the latent factors. Ideally, each latent factor should represent one genre. We set k = 20 and obtain the matrix V and pick the top 10 movies. We obtained the following results:

```
Column number of V: 1
Drama
Drama
Adventure—Animation—Comedy
Animation—Drama—Romance
Thriller
Comedy—Drama—Romance
Drama—War—Western
Drama
Fantasy—Horror
Comedy—Romance
————————————————————-

Column number of V: 3
Comedy—Horror
Action—Comedy—Crime
Comedy
Action—Sci-Fi
Drama—Romance
Mystery—Thriller
Comedy—Drama—Romance
Comedy—Drama
Drama—War
Action—Adventure—Sci-Fi
————————————————————-

Column number of V: 5
Action—Comedy—Crime—Drama
Comedy—Romance
Comedy
Adventure—Comedy—Thriller
Drama—Romance
Action—Crime—Sci-Fi—Thriller
Comedy—Romance
Drama—Romance—Sci-Fi
Romance—Sci-Fi
```

```
Comedy—Crime—Drama—Romance—Thriller
————————————————————————-

Column number of V: 7
Comedy—Drama—Romance
Adventure—Comedy—Thriller
Drama—Film-Noir—Mystery—Thriller
Comedy
Comedy—Drama—Romance
Drama—Romance
Crime
Horror—Thriller
Action—Crime—Drama—Thriller
Comedy—Drama—Romance
————————————————————————-

Column number of V: 11
Adventure—Animation—Children—Drama—Musical—IMAX
Drama
Crime—Drama
Animation—Children—Comedy—Musical—Romance
Action—Drama—Romance—Sci-Fi
Drama—Romance
Drama
Crime—Drama—Mystery
Drama—Mystery
Comedy
————————————————————————-

Column number of V: 15
Action—Sci-Fi—Thriller
Comedy—Drama
Comedy—Thriller
Animation—Comedy
Drama—Romance
Comedy
Comedy
Action—Adventure—Drama—War
Drama—Romance
Drama
————————————————————————-

Column number of V: 19
Drama—Film-Noir—Mystery—Romance
Comedy—Drama
Drama
Comedy—Drama—Romance
Documentary
Comedy
Comedy—Horror—Romance
Crime—Drama—Thriller
```

Documentary—Drama
Drama—Romance

From the above result, we note the following:

- We note that the latent factors are closely related to the movie genres. For example group 3 is strongly related to action, thriller, comedy while group 19 is tied to drama, romance, comedy.
- Thus we can note that the top 10 movies belong to a small set of genres for each group.

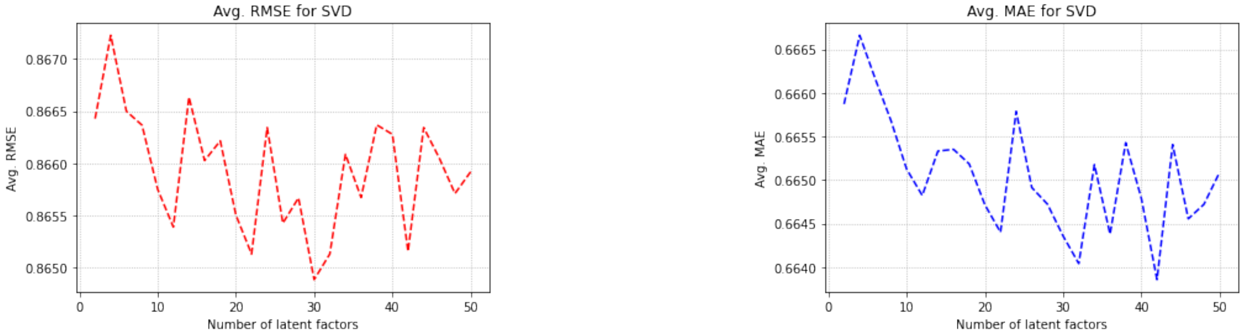
10 Question 10

Figure 9 below shows the RMSE and MAE plots against k for the MF model with bias. Though the plot is erratic in nature, we can note the both RMSE and MAE values are consistent within a tiny range of values for latent factor. The SVD actually outperforms both the NNMF CF and KNN models. This performance can be attributed to the following :

- SVD performs better due to the better factorization of the high dimensional matrix into lower dimensional matrices. This is due to the fact that there are no constraints on U and V and hence we obtain a factorization matrix with low information loss in higher and lower dimensions.
- SVD reduces the sensitivity to outliers by providing a better and more appropriate normalization to the user and movie specific bias information.
- SVD also produces embeddings of features with high relevance in a hierarchical manner and hence is ordered by relevance. Hence for any high value of k, the embeddings produced does not hinder the model due to the ordering of the features based on relevance.

We obtained the following results:

- Minimum average RMSE: **0.864886**, value of k: **30**
- Minimum average MAE: **0.663856**, value of k: **42**



(a) Average RMSE against k

(b) Average MAE against k

Figure 9: Average RMSE and MAE Plots

The following plots as shown in Figure 8 were obtained for NMF collaborative filtering for three trimming models.

- For POPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.8464**. We can note that popular trimmed set performed better than the original set. We can also see from the Figure 8.a that the average RMSE value decreases with increase in k. Also, the performance reduces as k increases and better performance is seen for lower values of k.
- For UNPOPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.8987**. This is much higher than that of popular trimmed dataset. From figure 8b we can see that the average RMSE does not follow the general monotonically decreasing trend and is fluctuating slightly. This erratic nature is due to the large number of outliers present. In this case, improving k does not necessarily improve the performance as the predictor does not have enough users for predicting the correct rating for rare items.
- For HIGH VARIANCE TRIMMING: **Minimum avg. RMSE** was found to be **1.4252**.This is way higher than both popular and unpopular sets and is thus, the worst in performance. On top of being non-monotonic, it is also extremely erratic in nature as seen in figure 6c. Since we use movies with high variance of ratings, the predictor becomes extremely sensitive to outliers and hence results in large prediction error.

Figure 10.d shows the **ROC curves** for all the three models. The AUC obtained for each threshold is as follows:

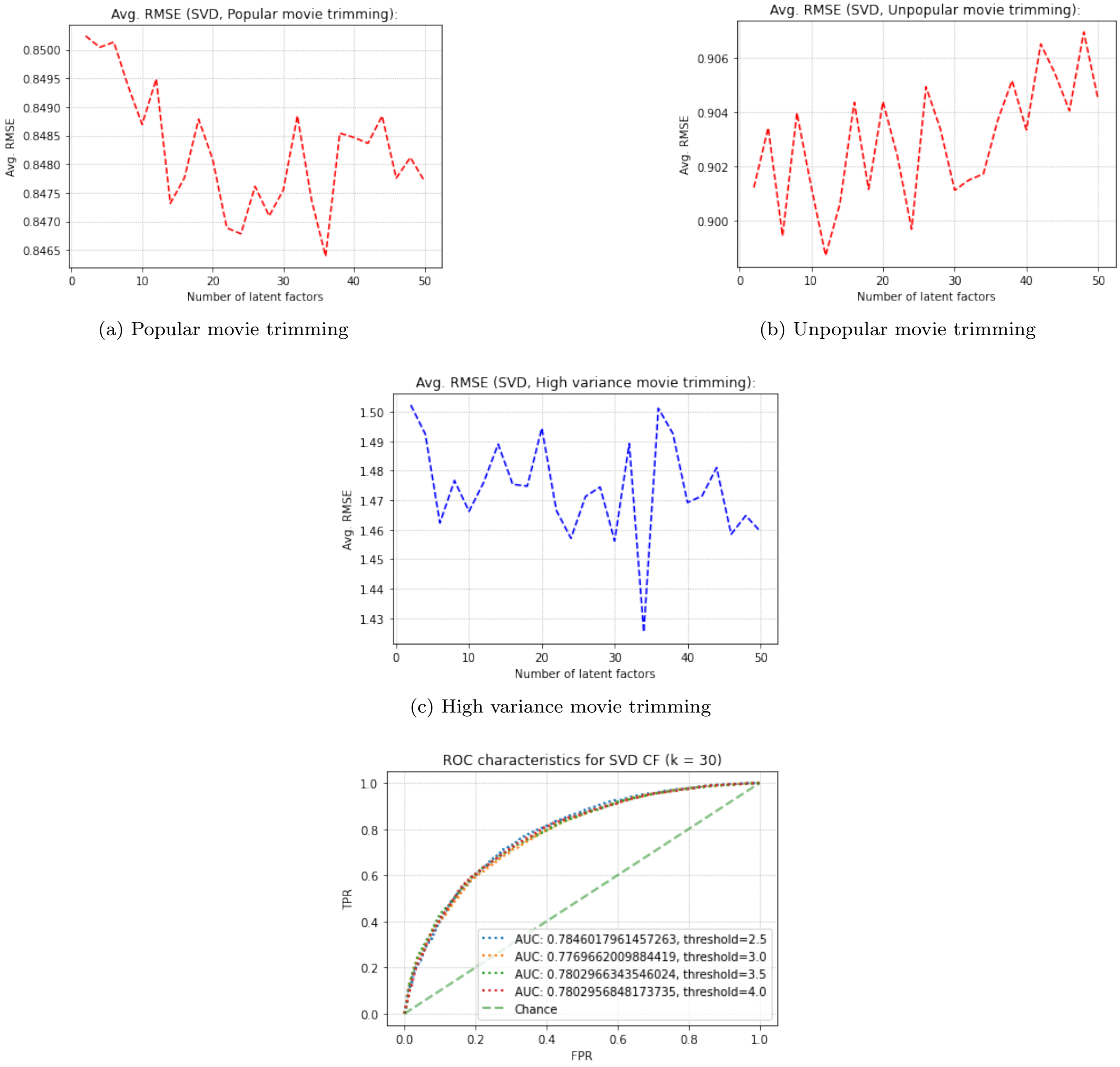
- Threshold = 2.5, AUC = 0.7846
- Threshold = 3.0, AUC = 0.7769
- Threshold = 3.5, AUC = 0.7803
- Threshold = 4.0, AUC = 0.7803

We get highest AUC for a threshold of 2.5. The values of AUC reduces with increase in threshold value and then increases again.

11 Question 11

The **Avg. RMSE** for Naive Filtering obtained was equal to **0.93471** across 10 folds.
For various trimming models, we obtained the following results:

- FOR POPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.9242**.
- FOR UNPOPULAR TRIMMING: **Minimum avg. RMSE** was found to be **0.9544**.
- FOR HIGH VARIANCE TRIMMING: **Minimum avg. RMSE** was found to be **1.4615**.



(a) Popular movie trimming

(b) Unpopular movie trimming

(c) High variance movie trimming

(d) ROC Curves

Figure 10: Filter performance on trimmed test sets (Average RMSE plots)

12 Question 12

From the figure we have compared the ROC curves for SVD CF (MF with Bias), KNN CF and NNMF CF for the threshold = 3. The AUC for each of the model are as follows :

- MF with Bias (SVD CF) - 0.7821
- NNMF CF - 0.7626
- KNN CF - 0.7799

Hence, from this we can infer that the MF with Bias performs the best followed by KNN CF and NNMF CF. The performances could be attributed to the following:

- Comparison of SVD and NMF models:
  - Comparing the SVD and NMF models, we can infer that the SVD is able to better represent the higher dimensional feature matrix as it does not have any constrains on U and V and thus provides a better matrix factorization with less information loss. On the other hand, the NMF imposes condition on the U and V and thus results in less number of optimal choices.
  - As discussed in the previous sections, we noted that the SVD produces embeddings of features with high relevance in a hierarchical manner and hence is ordered by relevance. Hence for any high value of k, the embeddings produced does not hinder the model due to the ordering of the features based on relevance. Due to this reason, they are robust to outliers and noise in the data when compared to the NMF that does not consider the geometry of the feature matrix.
- Comparison of SVD and KNN models:
  - The KNN predicts directly on the sprase rating matrix and thus yields poor prediction accuracy in higher dimensions due to the curse of dimensionality problem discussed in the above section. Thus, KNN models would be harder to scale and deploy as they need more data to perform on higher dimensions due to the sparseness of the matrix.
  - We can say that the KNN is more sensitive to outliers or rarely rated items as it does not consider the bias information for each user/item.

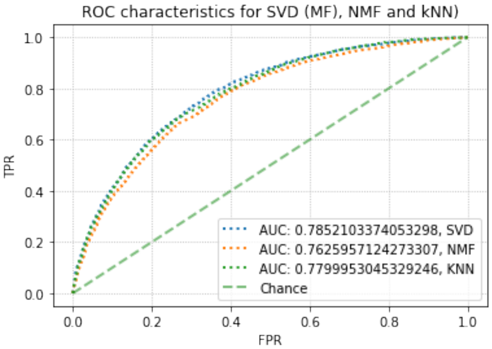


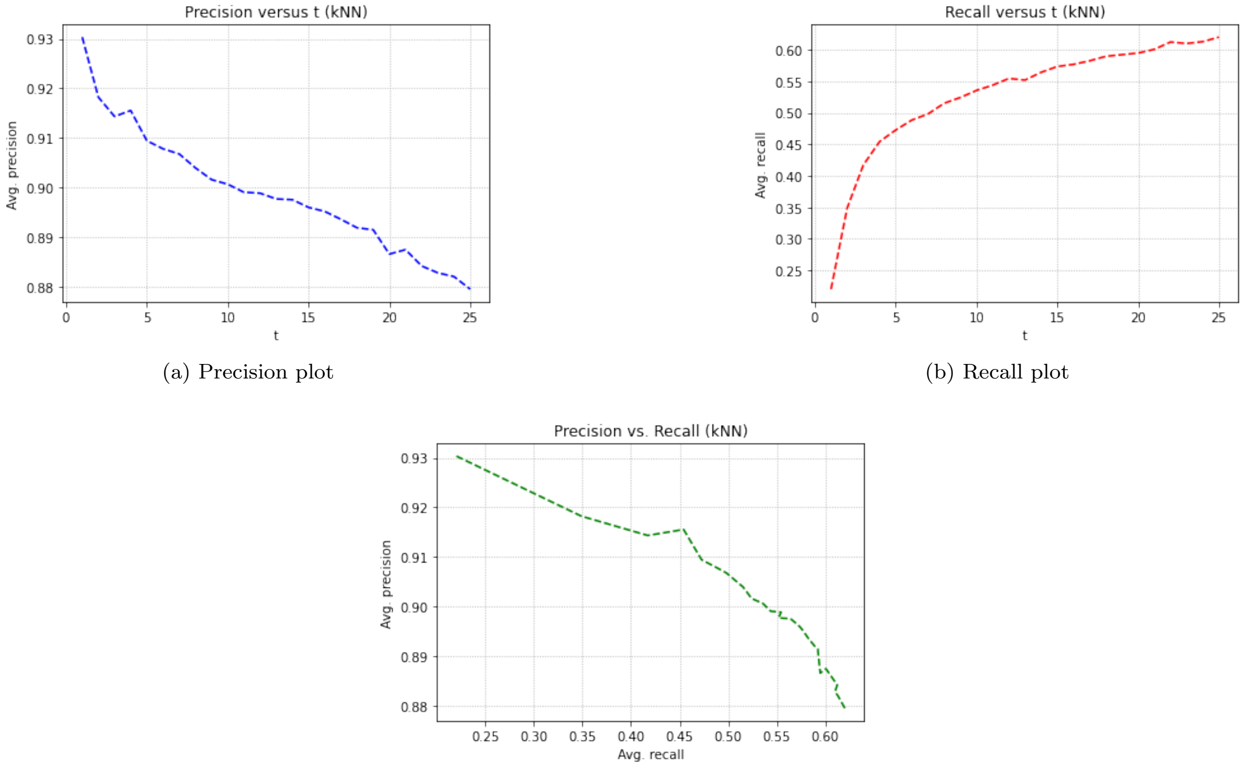
Figure 11: ROC curves (threshold = 3) for the k-NN, NMF, and MF with bias

13 Question 13

- Precision is the percentage of recommended items that are relevant (ground-truth positives), while Recall is the percentage of the relevant items (ground-truth positives) that have been recommended.
- In other words, Precision is the percentage of correct results out of all predicted results (all items recommended to users), while Recall is the percentage of correct results out of the relevant results (here, relevant items are those liked by users).
- In our context of recommendation systems, precision denotes the percentage of items that the user liked out of the set of items recommended to him. But, whereas, recall denotes all the items the user likes irrespective of it was recommended to the user or not.

14 Question 14

- **Explained plots for KNN model - Figure 12**
  - Figure 12 shows 3 plots for the KNN CF model for t values ranging from 1 to 25. From the figure we can infer that as t increases, the average precision decreases. This could be due to the increase in the number of false positive cases as the number of items recommended to the user increases. Although the precision decreases as t increases, the drop is around 5% for substantial increase in k and hence we could say the precision remains consistent with t.
  - With increase in t, we can infer that the recall increases as with a high value of t, the probability of its predicted ratings will include all the movies liked by user and hence increases. We can further say that, the recall is more sensitive to the number of items suggested to the user when compared to precision.
  - As recall increases, we can see that the average precision decreases as well. With a high recall, along with the inclusion of true positives in the recommendation list that the user would like, the model is also likely to include items that the users might not like and thus leading to lower precision. Hence, one can say that higher recall leads to lower precision and vice versa.



(a) Precision plot

(b) Recall plot

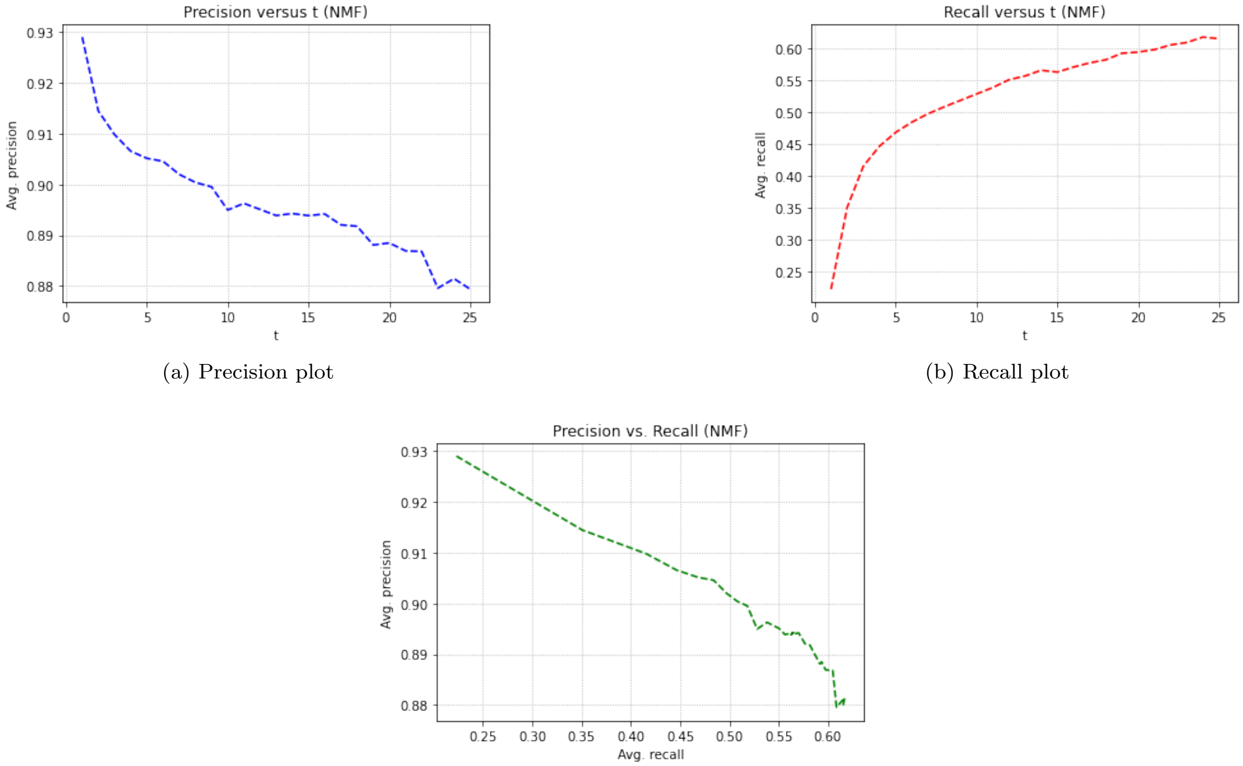
(c) Precision vs Recall

Figure 12: Precision-Recall plots obtained for kNN

- **Explained plots for NNMF CF - Figure 13**
  - From the figure we can infer that as t increases, the average precision decreases. This could be due to the increase in the number of false positive cases as the number of items recommended

to the user increases. Although the precision decreases as t increases, the drop is around 4% for substantial increase in k and hence we could say the precision remains consistent with t. However, this range is lower that what was observed for KNN CF.

- With increase in t, we can infer that the recall increases as with a high value of t, the probability of its predicted ratings will include all the movies liked by user and hence increases. We can further say that, the recall is more sensitive to the number of items suggested to the user when compared to precision.
- As recall increases, we can see that the average precision decreases as well. With a high recall, along with the inclusion of true positives in the recommendation list that the user would like, the model is also likely to include items that the users might not like and thus leading to lower precision. Hence, one can say that higher recall leads to lower precision and vice versa.



(a) Precision plot

(b) Recall plot

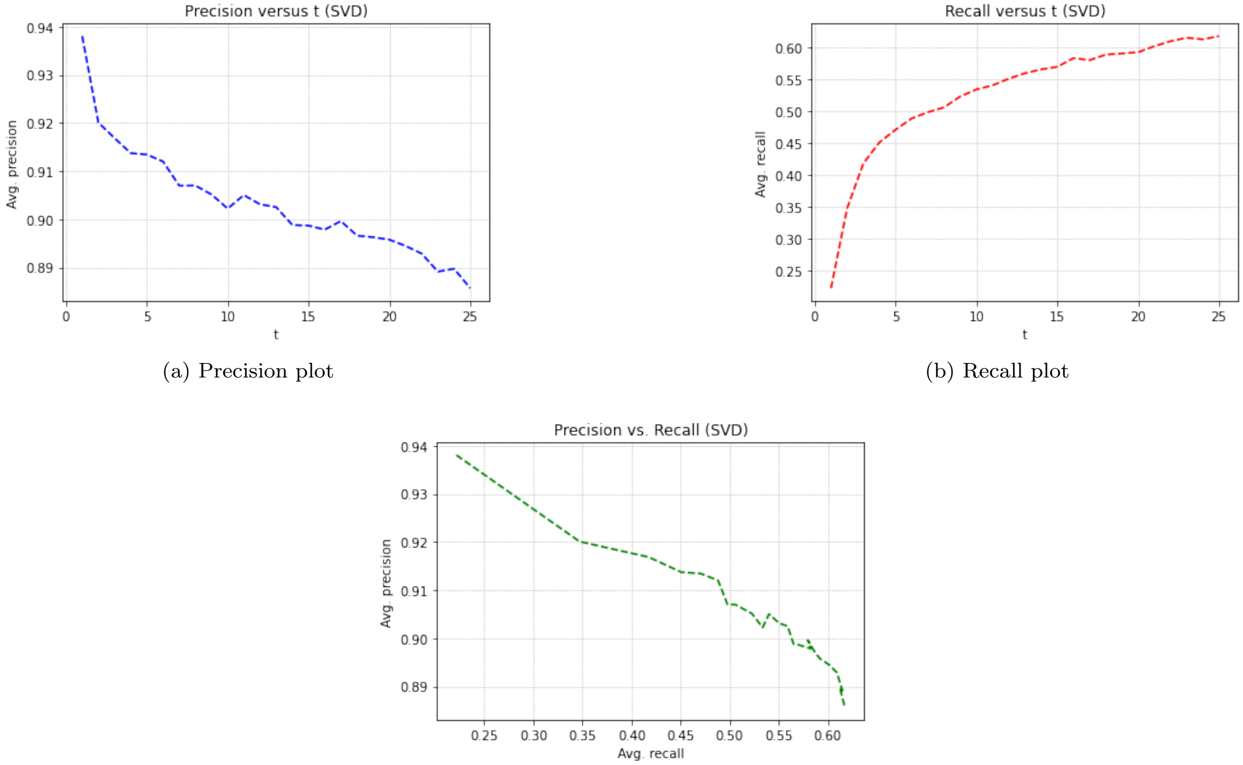
(c) Precision vs Recall

Figure 13: Precision-Recall plots obtained for NMF

- **Explained plots for SVD CF (MF with Bias)**
  - From the figure we can infer that as t increases, the average precision decreases. This could be due to the increase in the number of false positive cases as the number of items recommended to the user increases. Although the precision decreases as t increases, the drop is around 4% for substantial increase in k and hence we could say the precision remains consistent with t.
  - With increase in t, we can infer that the recall increases as with a high value of t, the probability of its predicted ratings will include all the movies liked by user and hence increases. We can

further say that, the recall is more sensitive to the number of items suggested to the user when compared to precision.

– As recall increases, we can see that the average precision decreases as well. With a high recall, along with the inclusion of true positives in the recommendation list that the user would like, the model is also likely to include items that the users might not like and thus leading to lower precision. Hence, one can say that higher recall leads to lower precision and vice versa.



(a) Precision plot

(b) Recall plot

(c) Precision vs Recall

Figure 14: Precision-Recall plots obtained for MF with bias

- **Explained Plot for Figure 15**

– From Figure 15, we observe that the SVD CF (MF with Bias) provides the best performance due to its slower drop in precision compared to the other two. It also maintains a higher precision for its corresponding recall value when compared to the rest two. The KNN CF performs second better and then followed by the NNMF CF. Hence we could say that the SVD CF model provides the most relevant set of items that is likely to be liked by the user) followed by KNN CF and NNMF CF.

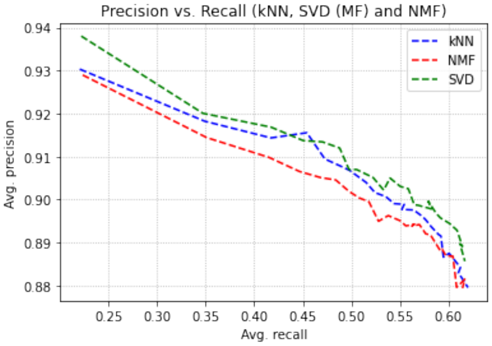


Figure 15: Precision-Recall curves obtained for the three models (kNN, NMF, MF)