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

Project 3: Data Representations and Clustering

Sudeeksha Agrawal, Tazeem Khan, Vamsi Krishna Pamidi

UID: 305928941,105946724,805945580

Introduction:

The increasing importance of the web as a medium for electronic and business transactions and advertisement, and social media has served as a driving force behind the development of recommender systems technology. Among the benefits, recommender systems provide a means to prioritize data for each user from the infinite information available on the internet. Such systems are critical to ensuring (among others): (a) the detection of hate speech, (b) user retention on a web service, and (c) fast and high-quality access to relevant information. An important catalyst is the ease with which the web enables users to provide feedback about a small portion of the web that they traverse. Such user-driven sparse feedback poses the following challenge in the design of recommender systems: Can we utilize these sparse user datapoints to infer generalized user interests? We define some terms:

• The entity to which the recommendation is provided is referred to as the user ;

• The product being recommended is an item.

The basic models for recommender systems works with two kinds of data:

A User-Item interactions such as ratings (a user (you) provides ratings about a movie (item));

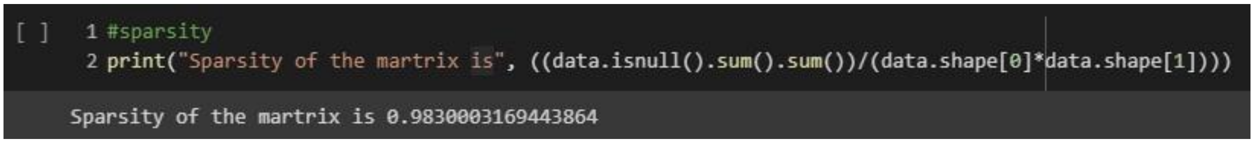
B Attribute information about the users and items such as textual profiles or relevant keywords (deep representations about a user or item).

Models that use type A data are referred to as collaborative filtering methods, whereas models that use type B data are referred to as content-based methods. In this project, we will build a recommendation system using collaborative filtering methods.

Dataset:

QUESTION 1: Explore the Dataset: In this question, we explore the structure of the data.

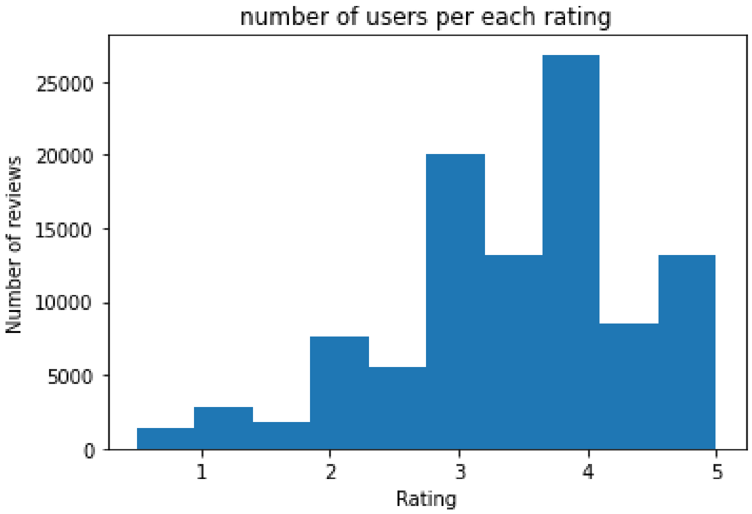
A) Compute the sparsity of the movie rating dataset :
Sparsity of the matrix is **0.9830003169443864**



```
#sparsity
print("Sparsity of the martrix is", ((data.isnull().sum().sum())/(data.shape[0]*data.shape[1])))
```

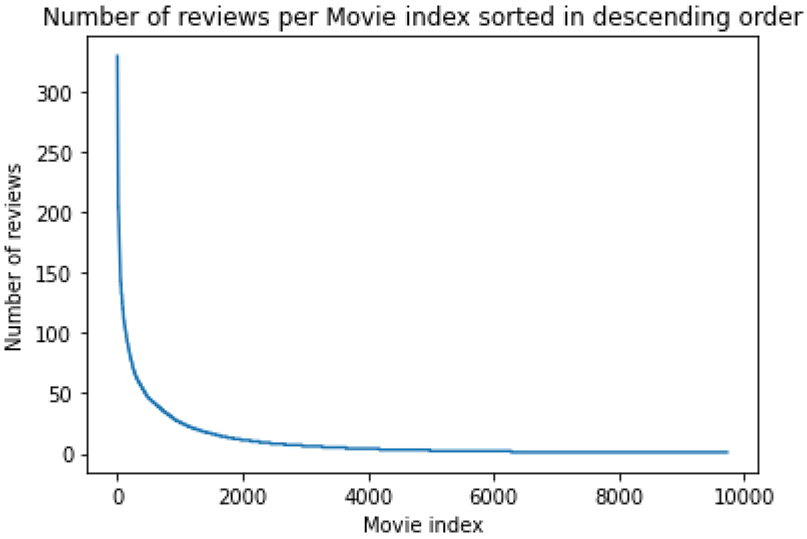
```
Sparsity of the martrix is 0.9830003169443864
```

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



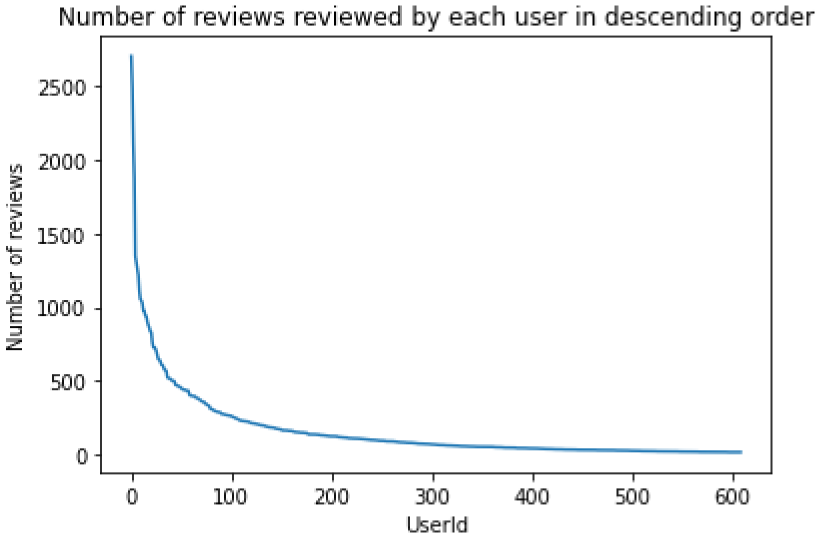
The distribution of rating values across the intervals is not uniform, indicating that movies were rated more highly than poorly. A majority of users rated movies in the range of 3.0 to 5.0, suggesting that most users enjoyed the movies they watched. This could be attributed to the fact that popular movies, hyped movies, and movies with high ratings from reputable sources are typically watched by more people and hence receive more ratings than unpopular movies. As a result, the recommendation system is biased towards recommending popular movies to users. Users may also have a tendency not to rate movies if they did not like them, which could explain the shape of the histogram. Additionally, integer ratings have higher counts than fractional ratings, as most people think of numbers as integers rather than fractions.

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



A monotonically decreasing trend can be seen

D) Plot the distribution of ratings among users:



A monotonically decreasing trend can be seen

E) Discuss the salient features of the distributions:

Sections C and D of the given text highlight an important trend observed in the data related to the MovieLens dataset. Specifically, it is evident that there is a significant decrease in the number of ratings received by movies and users in the dataset. This trend is reflected in the data showing that only a small number of movies and users received the majority of ratings, while the majority of movies and users received very few ratings.

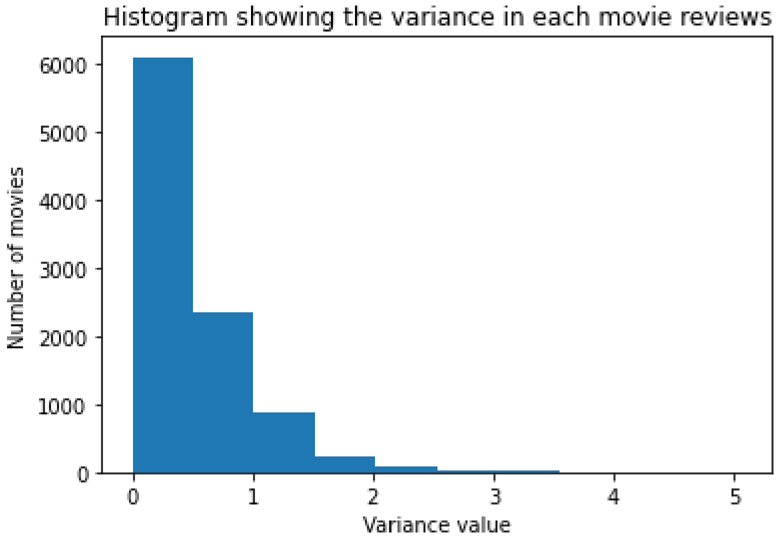
Part C of the text notes that the number of ratings has a reciprocal relationship with the movie index. This means that a small number of movies received a majority of the ratings, with only about 500 movies out of the total 9742 movies receiving more than 50 user ratings. Additionally, it is observed that around 20% of the movies received fewer than 10 ratings. This indicates that the majority of movies in the dataset received very few ratings, while a few popular movies attracted a large number of ratings.

Similarly, part D of the text indicates that the number of ratings also has a reciprocal relationship with the user index. This implies that a small number of users accounted for most of the ratings, with the highest number of ratings given by a user being a staggering 2698. Moreover, less than 50 users out of the total 610 provided ratings for more than 500 movies. This indicates that a majority of users in the dataset hardly rated movies, while a few users provided most of the ratings.

This sparsity in the data suggests that heavy regularization needs to be added to the recommendation process to prevent overfitting and false links. Moreover, the recommendation system tends to recommend popular movies to other users as popular movies attract a large number of ratings. The situation is similar from a user perspective, where only a few users are very active in terms of watching movies and rating them.

However, this bias towards popular movies and active users poses a challenge for the recommendation system. Specifically, it can result in a problem known as the "cold start problem" where the system struggles to make recommendations in the absence of sufficient data. Collaborative algorithms can be used to make recommendations, but the quality of recommendations may be poor due to the lack of data. Therefore, understanding this trend is crucial for developing accurate and effective recommendation systems.

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



We observed that there is a decline in the variance, indicating that a large number of movies were rated similarly with almost no significant differences in the ratings, differing by only 2.5 or less. This trend is reasonable as users tend to have similar opinions on movies. Since most of the variance falls between 0 and 2, we can conclude that the majority of the ratings are dependable and consistent.

Pearson-correlation coefficient

QUESTION 2: Understanding the Pearson Correlation Coefficient:

A) **Write down the formula for µu in terms of Iu and ruk;**

- Iu : Set of item indices for which ratings have been specified by user u;
- Iv : Set of item indices for which ratings have been specified by user v;
- µu: Mean rating for user u computed using her specified ratings;
- ruk: Rating of user u for item k.

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

B) **In plain words, explain the meaning of Iu ∩ Iv. Can Iu ∩ Iv = ∅? (Hint: Rating matrix R is sparse)**

Iu ∩ Iv represents the set of movies that both user *u* and user *v* have rated in the MovieLens dataset. Due to the sparsity of the ratings matrix, there is a high probability that there will be many movies that one user has rated but the other has not, meaning Iu ∩ Iv can be null. In fact, it is possible for two users to have no overlap in the movies they have rated. This is because the MovieLens dataset contains a large number of movies, but only a small fraction of them have been rated by a significant number of users. Additionally, many users have only rated a few movies, so it is possible that two users have no rated movies in common. In such cases, Iu ∩ Iv would be an empty set.

QUESTION 3: Understanding the Prediction function: Can you explain the reason behind mean-centering the raw ratings ($rvj - \mu v$) in the prediction function? (Hint: Consider users who either rate all items highly or rate all items poorly and the impact of these users on the prediction function.) :

- Mean-centering is a process that can minimize the impact of different rating behaviors among users. For instance, two users with similar movie preferences may rate differently, where one may give higher ratings more frequently than the other. Without mean-centering, this difference in rating behaviors may lead to inaccurate predictions by the Recommender System. By removing the mean of the ratings, we can reduce user-specific bias and outliers, resulting in less noisy data. This is important because users who give extreme opinions by rating all items either highly or poorly can create a biased view of the data, leading to inaccurate predictions. Therefore, mean-centering the ratings can help us obtain a more accurate prediction by providing a better understanding of the significance of the user's rating.

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. Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis).

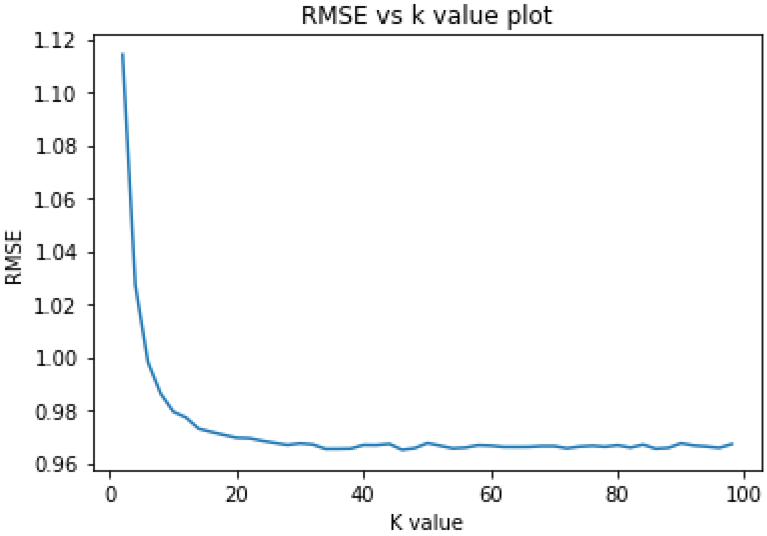


Figure : Average RMSE Vs Number of neighbors (k)

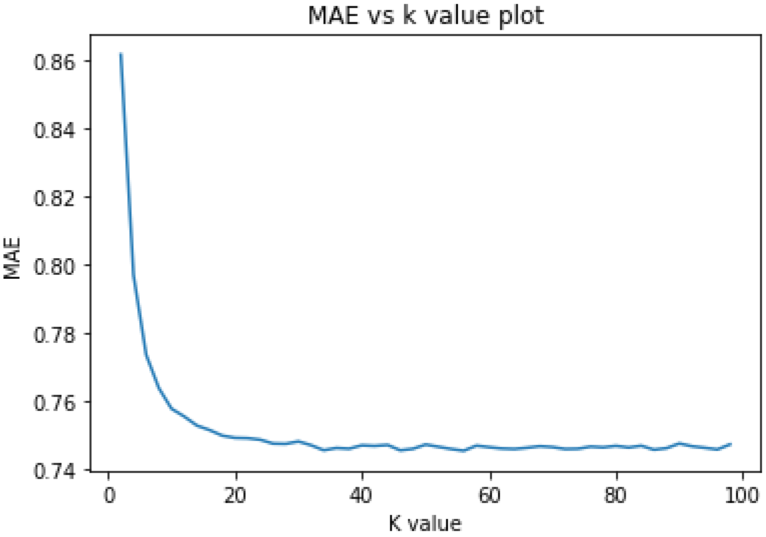


Figure : Average MAE Vs Number of neighbors (k)

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.

Minimum value of k seems to be = 22
Where Steady State RMSE = 0.9679758758885194
Steady State MAE = 0.7479665656727634

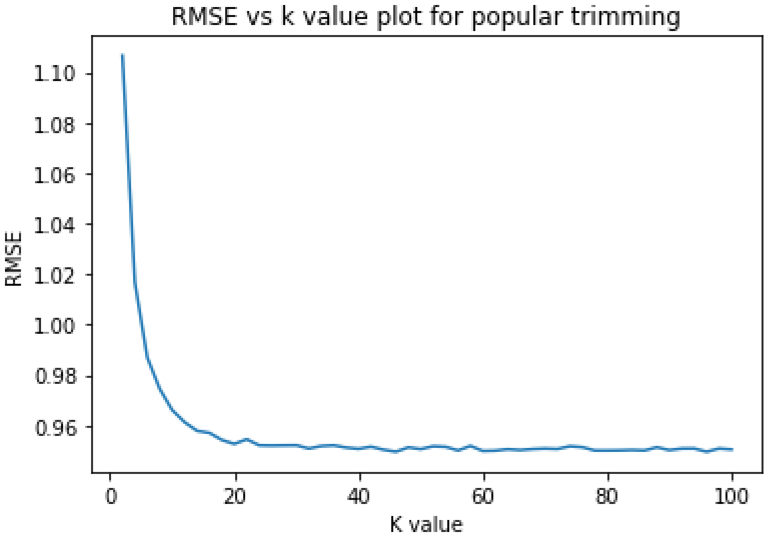
\

After observing the results, it is evident that increasing the value of k beyond 22 does not provide a significant improvement in reducing the error. It is just additional computation that can be avoided because when k is equal to 22, there are enough neighbors to generate a precise prediction even when there are outliers and noise present in the data.

QUESTION 6: Within EACH of the 3 trimmed subsets in the dataset, design (train and validate): A k-NN collaborative filter on the ratings of the movies (i.e Popular, Unpopular or High-Variance) and evaluate each of the three models’ performance using 10-fold cross validation:

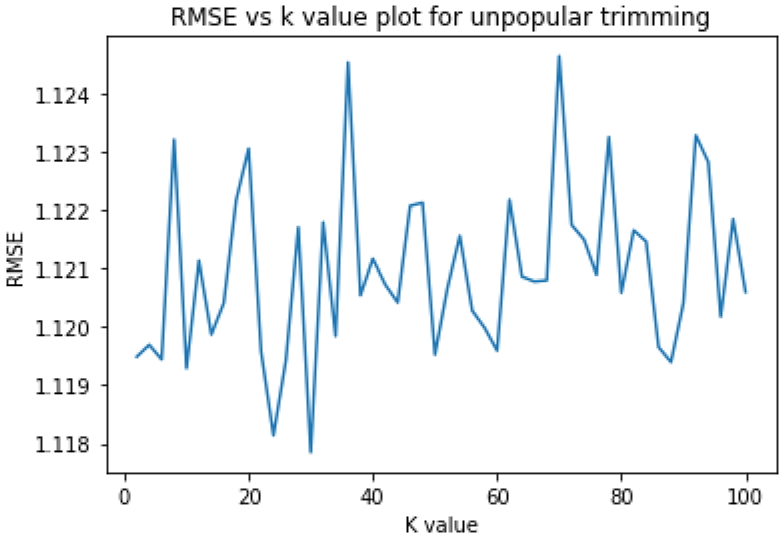
• Sweep k (number of neighbors) from 2 to 100 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.

Popular movie trimming:



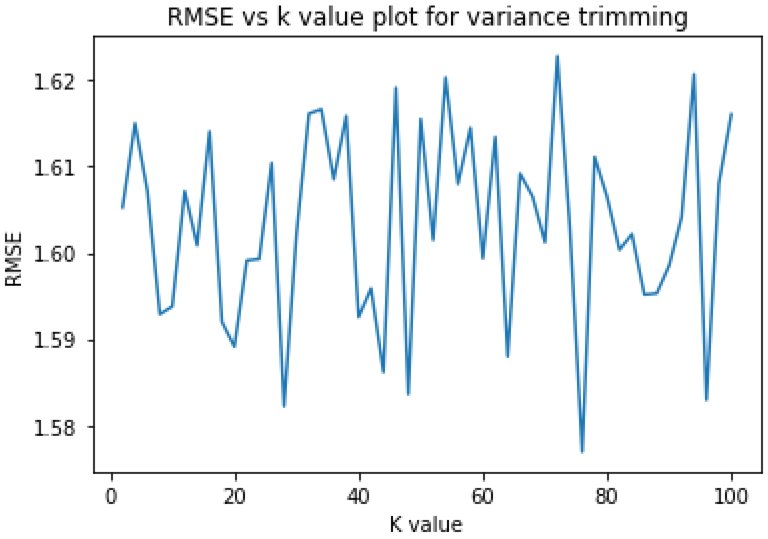
The minimum value of RMSE and the corresponding k value for popular trimming are 0.9494637857516519 and 96

Unpopular movie trimming:



The minimum value of RMSE and the corresponding k value for unpopular trimming are 1.115625452427866 and 96

High variance movie trimming:



The minimum value of RMSE and the corresponding k value for high variance trimming are 1.5769346730662754 and 76

- Plot the ROC curves for the k-NN collaborative filters for threshold values [2.5, 3, 3.5, 4]. These thresholds are applied only on the ground truth labels in the held-out validation set. For each of the plots, also report the area under the curve (AUC) value. You should have 4 × 4 plots in this section (4 trimming options – including no trimming times 4 thresholds) - all thresholds can be condensed into one plot per trimming option yielding only 4 plots.

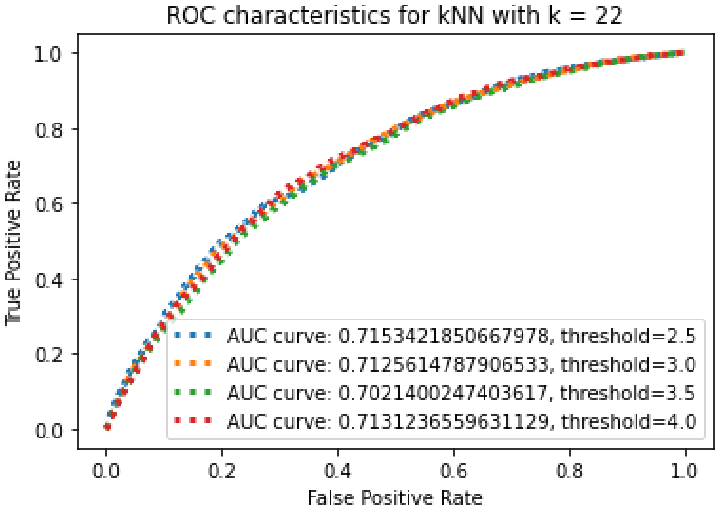


Figure : ROC curves for various thresholds for k-NN user-based CF with 22 neighbors

Threshold 2.5: 0.7153
Threshold 3: 0.71256
Threshold 3.5: 0.7021
Threshold 4: 0.7131

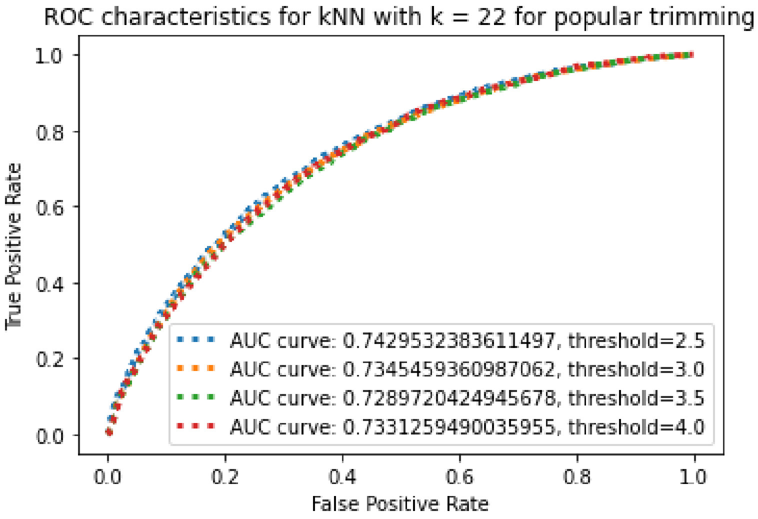


Figure : ROC curves for popular k-NN user-based CF with 22 neighbors

Threshold 2.5: 0.7429
Threshold 3: 0.7345
Threshold 3.5: 0.7289
Threshold 4: 0.7331

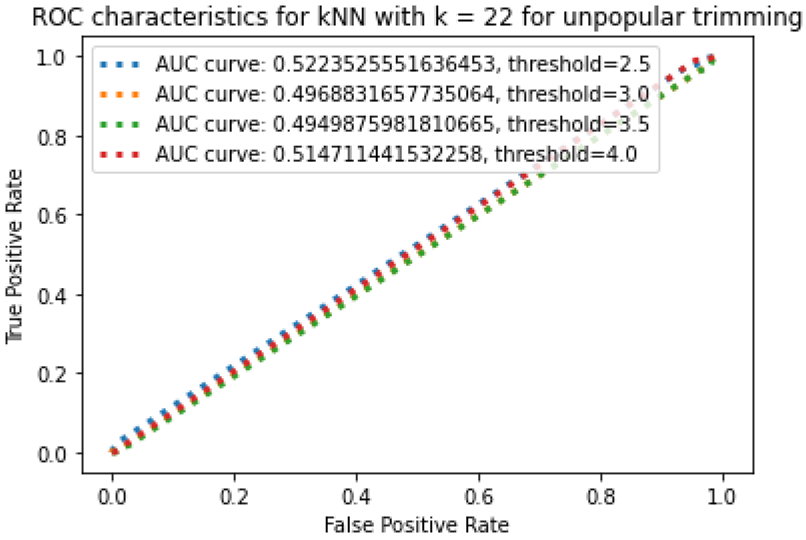


Figure : ROC curves for unpopular k-NN user-based CF with 22 neighbors

Threshold 2.5: 0.5223
Threshold 3: 0.4968
Threshold 3.5: 0.4949
Threshold 4: 0.5147

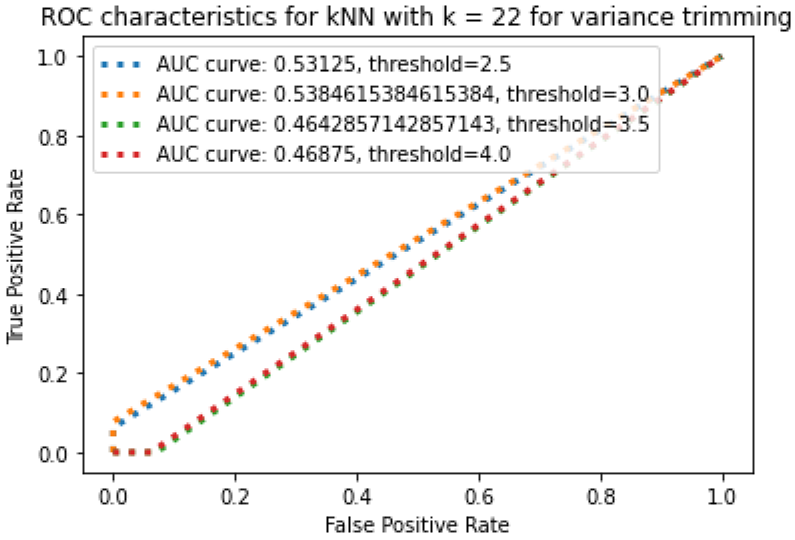
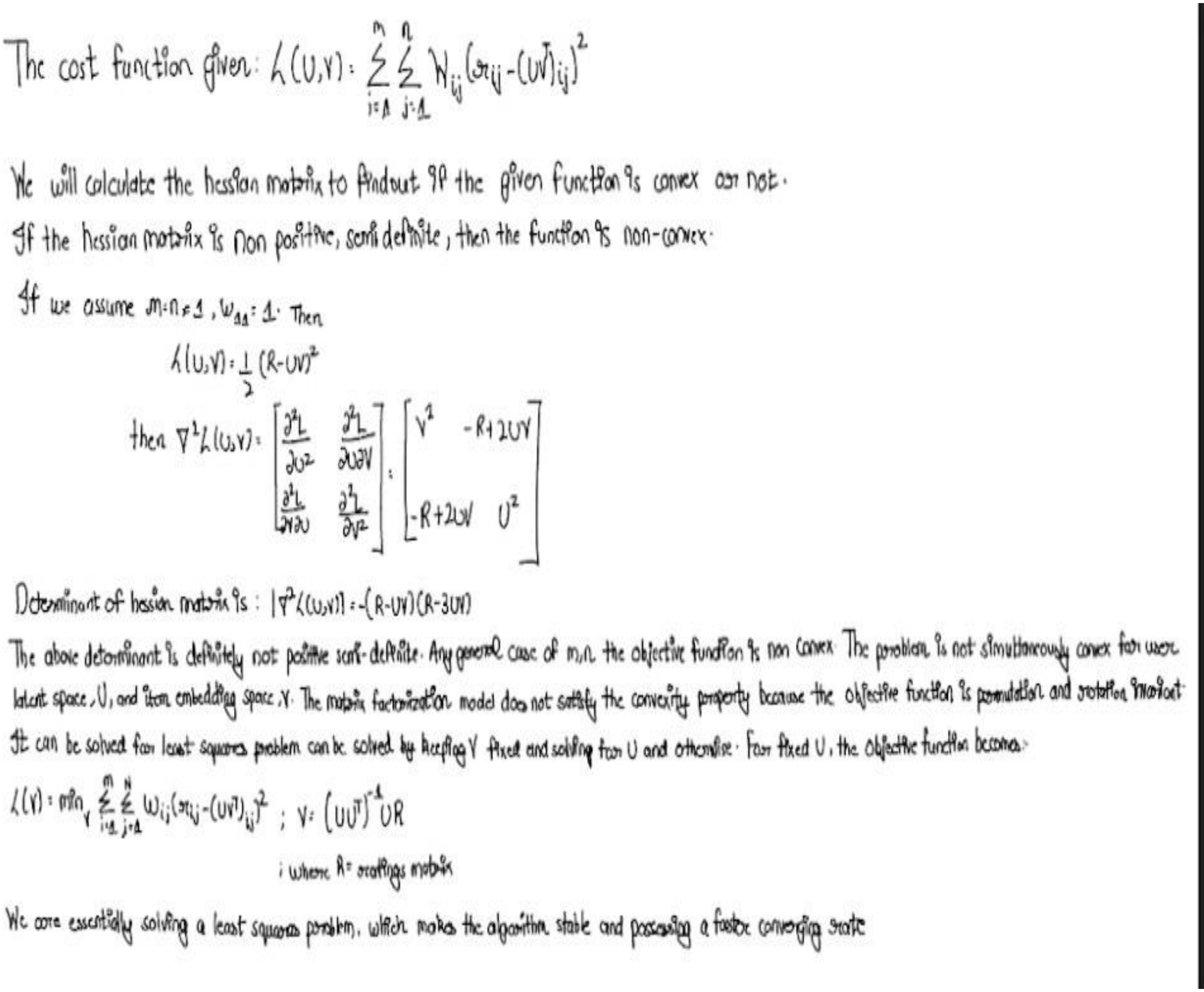


Figure : ROC curves for High Variance k-NN user-based CF with 22 neighbors

Threshold 2.5: 0.5312
Threshold 3: 0.5384
Threshold 3.5: 0.4643
Threshold 4: 0.4687

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.



The cost function given: $L(U,V) = \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2$

We will calculate the hessian matrix to findout if the given function is convex or not.

If the hessian matrix is non positive, semi definite, then the function is non-convex.

If we assume $m = n = 1$, $W_{11} = 1$. Then

$$L(U,V) = \frac{1}{2}(R - UV)^2$$

$$\text{then } \nabla^2 L(U,V) = \begin{bmatrix} \frac{\partial^2 L}{\partial U^2} & \frac{\partial^2 L}{\partial U \partial V} \\ \frac{\partial^2 L}{\partial V \partial U} & \frac{\partial^2 L}{\partial V^2} \end{bmatrix} = \begin{bmatrix} V^2 & -R + 2UV \\ -R + 2UV & U^2 \end{bmatrix}$$

Determinant of hession matrix is : $|\nabla^2 L(U,V)| = -(R - UV)(R - 3UV)$

The above determinant is definitely not positive semi-definite. Any general case of m,n the objective function is non convex. The problem is not simultaneously convex for user latent space, U, and item embedding space, V. The matrix factorization model does not satisfy the convexity property because the objective function is permutation and rotation invariant. It can be solved for least squares problem can be solved by keeping V fixed and solving for U and otherwise. For fixed U, the objective function becomes

$$L(V) = \min_V \sum_{i=1}^{m} \sum_{j=1}^{N} W_{ij}(r_{ij} - (UV^T)_{ij})^2 \;;\; V = (UU^T)^{-1} UR$$

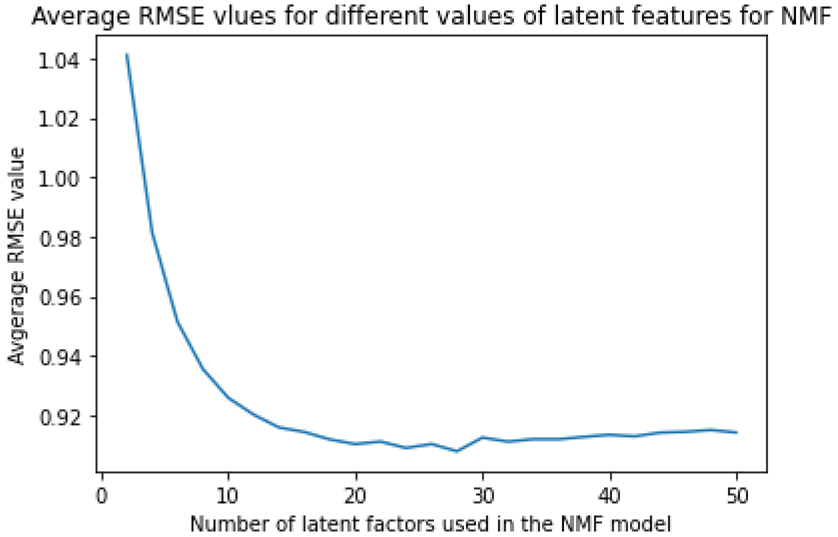
; where R = ratings matrix

We are essentially solving a least squares problem, which makes the algorithm stable and possessing a faster converging state

QUESTION 8: Designing the NMF Collaborative Filter:

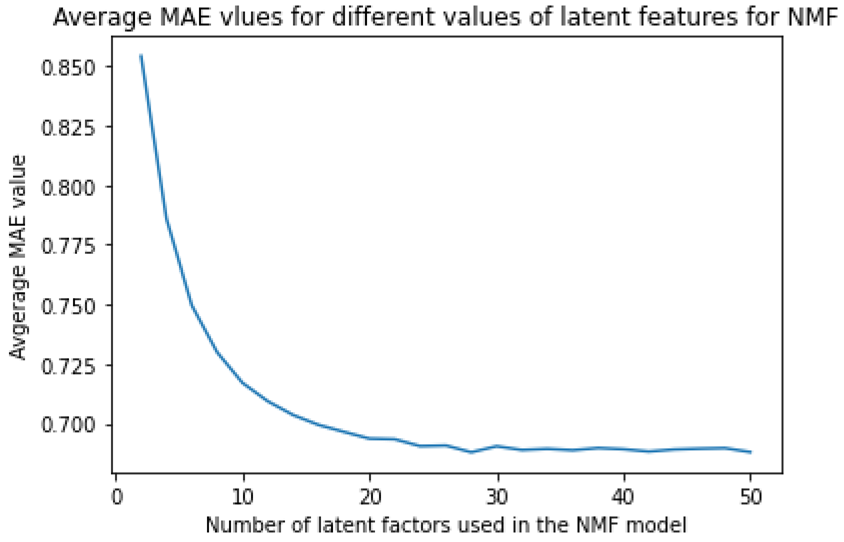
A) Design a NMF-based collaborative filter to predict the ratings of the movies in the original dataset and evaluate its performance using 10-fold cross-validation. Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE and average MAE obtained by averaging the RMSE and MAE across all 10 folds. If NMF takes too long, you can increase the step size. Increasing it too much will result in poorer granularity in your results. Plot the average RMSE (Y-axis) against k (X-axis) and the average MAE (Yaxis) against k (X-axis). For solving this question, use the default value for the regularization parameter.

B) Use the plot from the previous part to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE. Is the optimal number of latent factors same as the number of movie genres?



The minimum value of RMSE and the corresponding number of latent factors are 0.9079134722068547 and 28

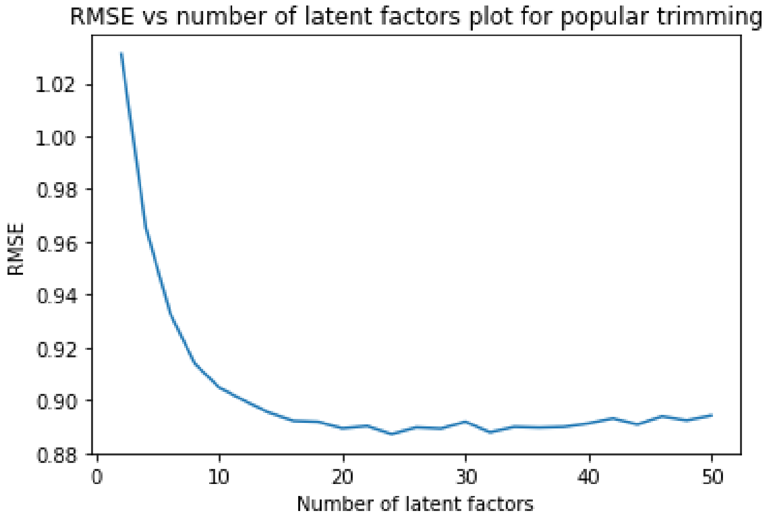
We observe the values decrease with the minimum at around k = 28, and then the values start increasing slightly.



The minimum value of MAE and the corresponding number of latent factors are 0.6879737058262434 and 28.
We observe the values decrease with the minimum at around k = 28, and remain around the same for k's after that.

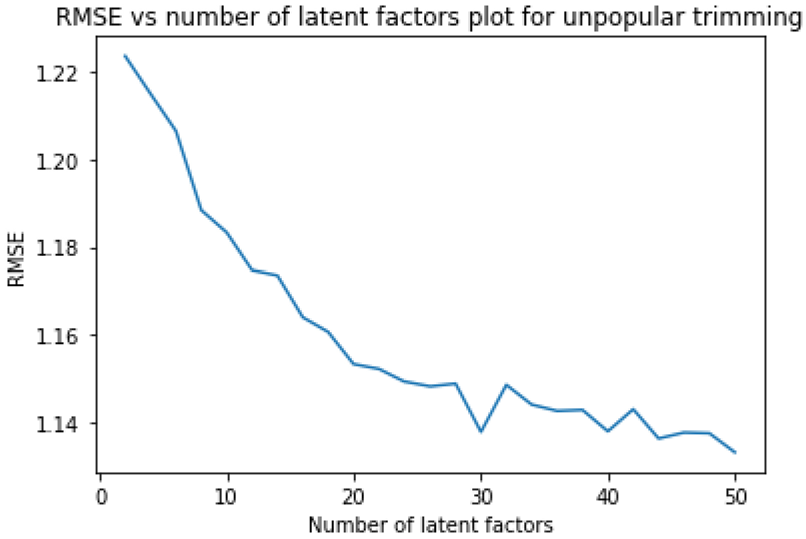
C) C Performance on trimmed dataset subsets: For each of Popular, Unpopular and HighVariance subsets - – Design a NMF collaborative filter for each trimmed subset and evaluate its performance using 10-fold cross validation. Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds. – Plot average RMSE (Y-axis) against k (X-axis); item Report the minimum average RMSE.

Popular movie trimming:



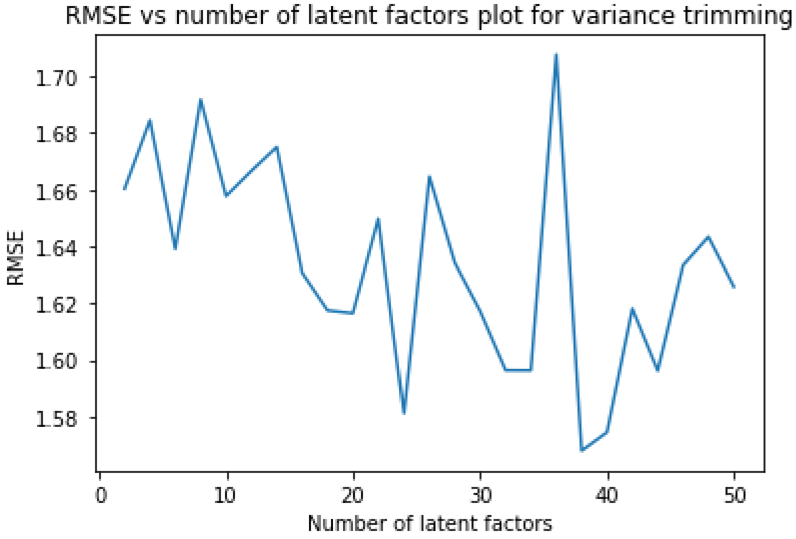
The minimum value of RMSE and the corresponding number of latent factors for popular trimming are 0.8871521626739508 and 24

Unpopular movie trimming:



The minimum value of RMSE and the corresponding number of latent factors for unpopular trimming are 1.133099431870318 and 50

High variance movie trimming:



The minimum value of RMSE and the corresponding number of latent factors for variance trimming are 1.56800591777239 and 38

Plot the ROC curves for the NMF-based collaborative filter and also report the area under the curve (AUC) value as done in Question 6.

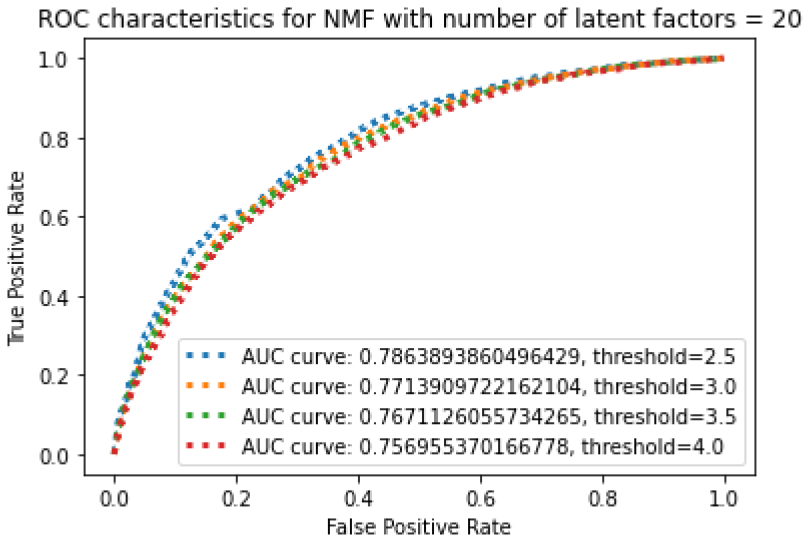


Figure : ROC curves for various thresholds for NMF user-based CF with 20 neighbors

Area Under the Curve Values (AUC):
Threshold 2.5: 0.7864
Threshold 3: 0.7714
Threshold 3.5: 0.7671
Threshold 4: 0.7569

Popular Trimming:

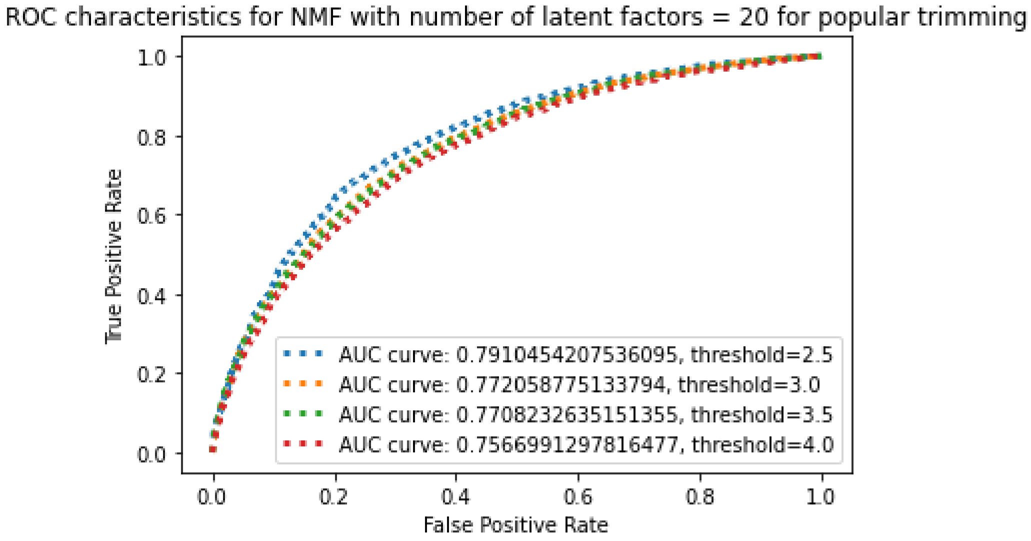


Figure : ROC curves for popular NMF user-based CF with 20 neighbors

Area Under the Curve Values (AUC):

Threshold 2.5: 0.7910

Threshold 3: 0.7721

Threshold 3.5: 0.771

Threshold 4: 0.7567

Unpopular movie trimming:

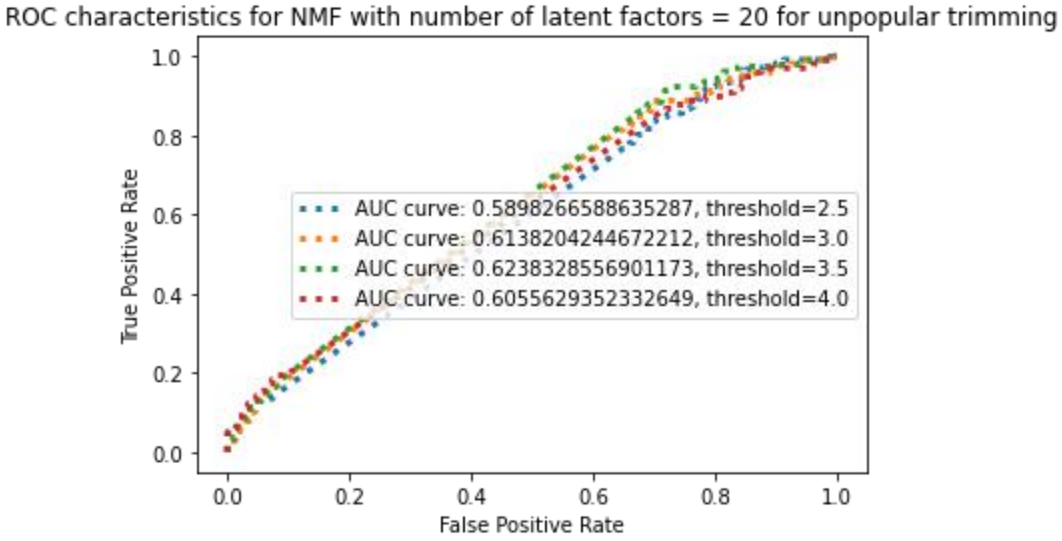


Figure : ROC curves for unpopular NMF user-based CF with 20 neighbors

Area Under the Curve Values (AUC):

Threshold 2.5: 0.5898

Threshold 3: 0.6138

Threshold 3.5: 0.6238

Threshold 4: 0.6055

High Variance movie trimming:

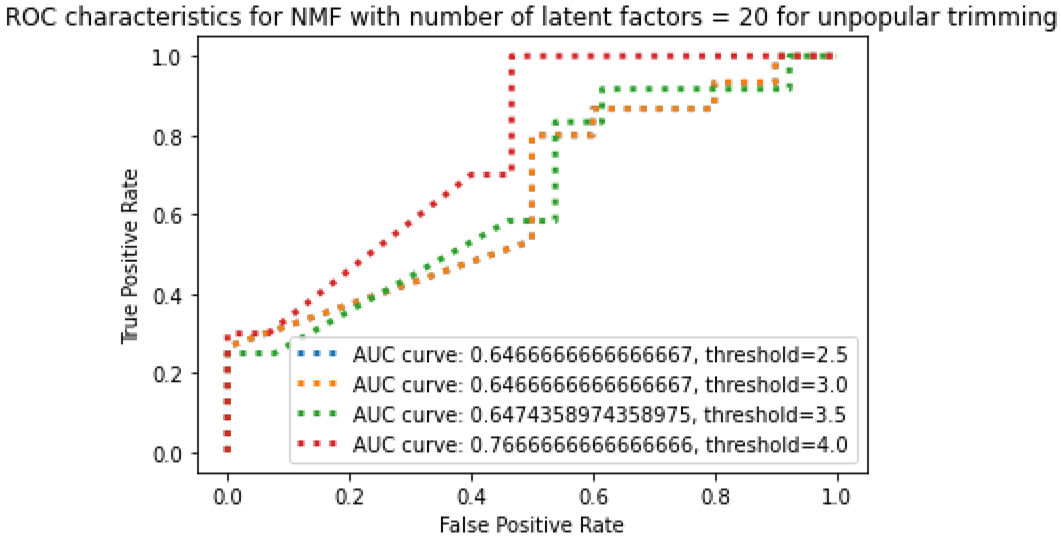


Figure : ROC curves for high variance NMF user-based CF with 20 neighbors

Area Under the Curve Values (AUC):

Threshold 2.5: 0.6466

Threshold 3: 0.6466

Threshold 3.5: 0.6474

Threshold 4: 0.7666

QUESTION 9: Interpreting the NMF model: Perform Non-negative matrix factorization on the ratings matrix R to obtain the factor matrices U and V , where U represents the user-latent factors interaction and V represents the movie-latent factors interaction (use k = 20). For each column of V , sort the movies in descending order and report the genres of the top 10 movies. Do the top 10 movies belong to a particular or a small collection of genre? Is there a connection between the latent factors and the movie genres?

-

The top 10 movies genre list for the 0 th column are
Comedy|Drama|Romance
Action|Adventure|Fantasy
Drama|Mystery|Thriller
Drama
Drama
Action|Adventure|Comedy|Fantasy|Horror|Thriller
Mystery|Thriller
Drama
Action|Adventure|Fantasy|War
Comedy|Horror|Mystery|Thriller
The unique genres in this column are {**'Fantasy', 'Thriller', 'Comedy', 'Horror', 'Romance', 'Mystery', 'Drama', 'War', 'Action', 'Adventure'**}
Their number is 10

The top 10 movies genre list for the 1 th column are
Crime|Drama|Romance|Thriller
Drama|War
Crime|Drama
Action|Drama|War
Adventure|Animation|Comedy|Fantasy|Romance
Comedy|Horror|Sci-Fi
Comedy|Musical|Romance
Comedy|Romance
Drama
Drama
The unique genres in this column are {**'Fantasy', 'Thriller', 'Animation', 'Adventure', 'Comedy', 'Horror', 'Musical', 'Sci-Fi', 'Romance', 'Action', 'Drama', 'Crime', 'War'**}
Their number is 13

The top 10 movies genre list for the 2 th column are
Action|Crime|Drama|Thriller
Action|Adventure|Romance
Comedy|Romance
Adventure|Children
Drama|Romance
Drama
Comedy|Drama|Romance
Documentary
Animation|Children|Comedy
Adventure|Children|Comedy|Fantasy
The unique genres in this column are {**'Fantasy', 'Thriller', 'Animation', 'Comedy', 'Children', 'Romance', 'Documentary', 'Drama', 'Action', 'Adventure', 'Crime'**}
Their number is 11

The top 10 movies genre list for the 3 th column are
Documentary
Animation|Comedy|Fantasy|Sci-Fi
Drama|Fantasy|Mystery|Sci-Fi
Thriller
Adventure|Drama
Crime|Horror|Sci-Fi
Comedy|Musical
Crime|Drama
Drama|Romance|Sci-Fi
Crime|Drama
The unique genres in this column are **{'Fantasy', 'Thriller', 'Animation', 'Comedy', 'Horror', 'Musical', 'Sci-Fi', 'Romance', 'Mystery', 'Documentary', 'Drama', 'Crime', 'Adventure'}**
Their number is 13

The top 10 movies genre list for the 4 th column are
Comedy
Crime|Drama
Comedy|Romance|Thriller
Drama|Mystery
Comedy|Romance
Horror
Adventure|Children|Comedy
Drama|Romance|Sci-Fi
Drama

Comedy|Drama|Romance
The unique genres in this column are **{'Thriller', 'Comedy', 'Adventure', 'Children', 'Sci-Fi', 'Romance', 'Mystery', 'Drama', 'Crime', 'Horror'}**
Their number is 10

The top 10 movies genre list for the 5 th column are
Drama|Fantasy|Sci-Fi
Action|War
Action
Drama
Comedy|Drama
Drama|Musical|Romance
Action|Adventure|Sci-Fi
Action|Comedy|Crime
Drama
Horror
The unique genres in this column are **{'Fantasy', 'Comedy', 'Adventure', 'Horror', 'Musical', 'Sci-Fi', 'Romance', 'Drama', 'Action', 'War', 'Crime'}**
Their number is 11

The top 10 movies genre list for the 6 th column are
Children|Comedy|Drama
Comedy|Crime|Romance
Romance
Animation|Children|Comedy|Musical
Comedy|Drama
Action|Drama|War
Documentary
Comedy
Comedy
Comedy
The unique genres in this column are **{'Animation', 'Comedy', 'Musical', 'Children', 'Romance', 'Action', 'Documentary', 'Drama', 'Crime', 'War'}**
Their number is 10

The top 10 movies genre list for the 7 th column are
Drama
Action|Adventure|Sci-Fi|Thriller
Drama
Comedy

Documentary

Comedy|Musical

Comedy

Action|Adventure|Drama|Thriller

Action|Comedy|Crime|Thriller

Comedy|Drama|Romance

The unique genres in this column are **{'Thriller', 'Comedy', 'Musical', 'Sci-Fi', 'Romance', 'Documentary', 'Drama', 'Action', 'Adventure', 'Crime'}**

Their number is 10

The top 10 movies genre list for the 8 th column are

Animation|Comedy|Fantasy|Sci-Fi

Action|Sci-Fi

Adventure

Fantasy|Western

Action|Drama|Romance

Comedy

Drama|Thriller

Crime|Drama

Crime|Horror|Mystery|Thriller

Fantasy|Mystery|Western

The unique genres in this column are **{'Fantasy', 'Thriller', 'Animation', 'Comedy', 'Horror', 'Sci-Fi', 'Romance', 'Western', 'Mystery', 'Drama', 'Action', 'Adventure', 'Crime'}**

Their number is 13

The top 10 movies genre list for the 9 th column are

Action|Drama|Sci-Fi

Documentary

Drama

Comedy

Action|Comedy|Crime|Thriller

Drama|Romance|War

Comedy|Drama|Romance

Thriller

Comedy|Documentary

Drama

The unique genres in this column are **{'Thriller', 'Comedy', 'Sci-Fi', 'Romance', 'Documentary', 'Drama', 'Action', 'War', 'Crime'}**

Their number is 9

The top 10 movies genre list for the 10 th column are
Drama
Action|Adventure|Comedy|Crime
Crime|Drama
Comedy|Drama
Action|Comedy|Crime|Mystery
Comedy|Drama
Comedy
Drama
Action|Animation|Sci-Fi|Thriller
Action|Adventure|Comedy|Thriller
The unique genres in this column are **{'Thriller', 'Animation', 'Comedy', 'Sci-Fi', 'Mystery', 'Drama', 'Action', 'Adventure', 'Crime'}**
Their number is 9

The top 10 movies genre list for the 11 th column are
Drama
Comedy|Drama
Drama
Comedy
Comedy|Romance
Crime|Drama|Mystery|Thriller
Horror|Western
Action|Thriller|War
Horror|Mystery|Thriller
Crime|Drama|Thriller
The unique genres in this column are **{'Thriller', 'Comedy', 'Romance', 'Action', 'Western', 'Mystery', 'Drama', 'War', 'Crime', 'Horror'}**
Their number is 10

The top 10 movies genre list for the 12 th column are
Drama|Romance
Comedy|Crime|Romance
Sci-Fi
Crime|Drama|Thriller
Crime|Horror|Mystery|Thriller
Drama|Romance
Action|Adventure|Animation|Fantasy|Sci-Fi
Crime|Mystery|Thriller
Drama|Thriller

Action|Comedy|Crime
The unique genres in this column are **{'Fantasy', 'Thriller', 'Animation', 'Comedy', 'Adventure', 'Sci-Fi', 'Romance', 'Action', 'Mystery', 'Drama', 'Crime', 'Horror'}**
Their number is 12

The top 10 movies genre list for the 13 th column are
Drama
Crime|Film-Noir
Adventure|Children|Fantasy
Crime|Drama
Drama|Mystery
Action|Thriller|War
Drama
Action|Crime|Sci-Fi|Thriller
Crime|Drama|Thriller
Action|Drama|War
The unique genres in this column are **{'Fantasy', 'Thriller', 'Film-Noir', 'Children', 'Sci-Fi', 'Action', 'Mystery', 'Drama', 'War', 'Crime', 'Adventure'}**
Their number is 11

The top 10 movies genre list for the 14 th column are
Comedy|Romance
Comedy
Action|Drama|Thriller
Action|Crime|Drama
Drama
Comedy|Romance
Horror|Sci-Fi
Comedy|Romance
Action|Thriller
Action|Drama
The unique genres in this column are **{'Thriller', 'Comedy', 'Sci-Fi', 'Romance', 'Drama', 'Action', 'Horror', 'Crime'}**
Their number is 8

The top 10 movies genre list for the 15 th column are
Crime|Drama|Mystery|Thriller
Comedy|Documentary
Comedy|Drama|Romance
Drama|Horror|Sci-Fi

Drama
Drama|Romance
Drama|War
Documentary|Drama
Romance
Action|Adventure|Fantasy
The unique genres in this column are **{'Fantasy', 'Thriller', 'Comedy', 'Adventure', 'Sci-Fi', 'Romance', 'Action', 'Mystery', 'Documentary', 'Drama', 'War', 'Crime', 'Horror'}**
Their number is 13

The top 10 movies genre list for the 16 th column are
Action|War
Action|Adventure|Thriller
Mystery|Thriller
Drama
Comedy|Romance
Comedy|Musical|Romance
Action|Adventure|Mystery|Thriller
Action|Fantasy|Horror|Sci-Fi|Thriller
Comedy|Drama
Comedy
The unique genres in this column are **{'Fantasy', 'Thriller', 'Comedy', 'Horror', 'Musical', 'Sci-Fi', 'Romance', 'Mystery', 'Drama', 'War', 'Action', 'Adventure'}**
Their number is 12

The top 10 movies genre list for the 17 th column are
Children|Comedy|Fantasy|Horror
Comedy|Drama|Musical
Drama|Thriller
Drama|Romance
Comedy|Fantasy
Drama|Sci-Fi
Adventure|Western
Comedy|Drama|Mystery|Thriller
Comedy|Drama|Fantasy|Romance
Crime|Drama|Thriller
The unique genres in this column are **{'Fantasy', 'Thriller', 'Comedy', 'Adventure', 'Musical', 'Children', 'Sci-Fi', 'Romance', 'Western', 'Mystery', 'Drama', 'Crime', 'Horror'}**
Their number is 13

The top 10 movies genre list for the 18 th column are

```
Horror|Mystery|Thriller
Comedy
Adventure
Horror|Mystery|Thriller
Drama|Fantasy|Musical|Romance
Drama|Romance
Drama
Drama|Horror|Thriller
Drama
Action|Sci-Fi|Thriller
```

The unique genres in this column are **{'Fantasy', 'Thriller', 'Comedy', 'Horror', 'Musical', 'Sci-Fi', 'Romance', 'Mystery', 'Drama', 'Action', 'Adventure'}**

Their number is 11

The top 10 movies genre list for the 19 th column are

```
Comedy
Drama
Comedy|Drama
Adventure|Comedy|Drama|Romance
Comedy
Comedy|Crime|Mystery|Romance
Animation|Children|Fantasy|Mystery
Action|Sci-Fi|Thriller|Western|IMAX
Action|Comedy|Crime|Thriller
Comedy|Romance
```

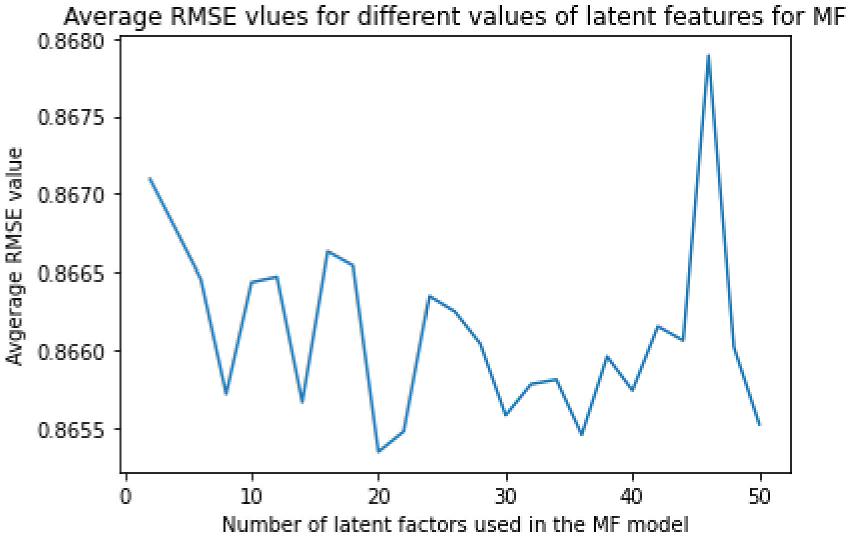
The unique genres in this column are **{'Fantasy', 'Thriller', 'Animation', 'Comedy', 'IMAX', 'Children', 'Sci-Fi', 'Romance', 'Action', 'Western', 'Mystery', 'Drama', 'Crime', 'Adventure'}**

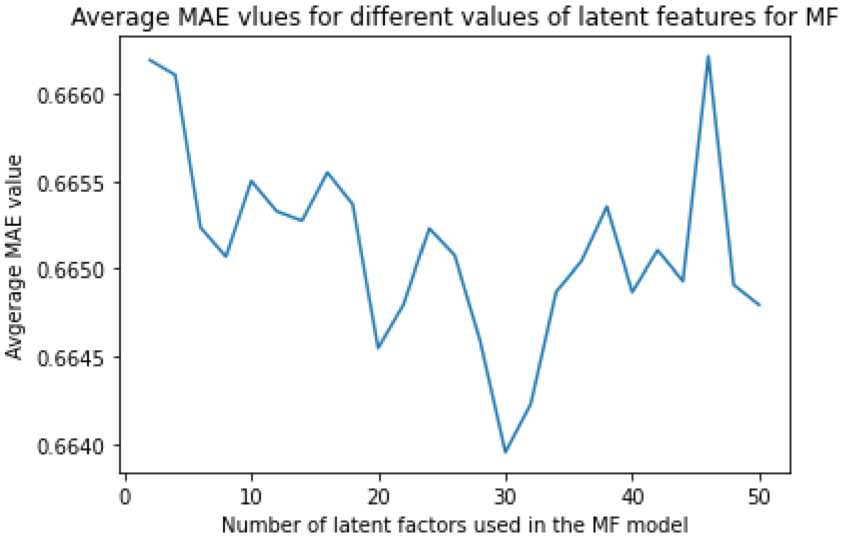
Their number is 14

The above results describe the genres of the top 10 movies for different latent factors. For example, the top 10 movies for latent factor 1 are mostly dramas, crimes, and action movies, indicating that this latent factor combines a few genres. Similarly, latent factor 13 represents fantasy. For all latent factors, we observe that the top 10 movies belong to a small set of genres. As the column number of the latent factor increases, the number of distinct movie genres decreases or remains the same. This suggests that movies of similar genres tend to cluster together for higher column numbers. Overall, NMF seems to cluster movies based on their genre, and each latent factor represents a particular genre type or a small set of related genres.

QUESTION 10: Designing the MF Collaborative Filter:

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





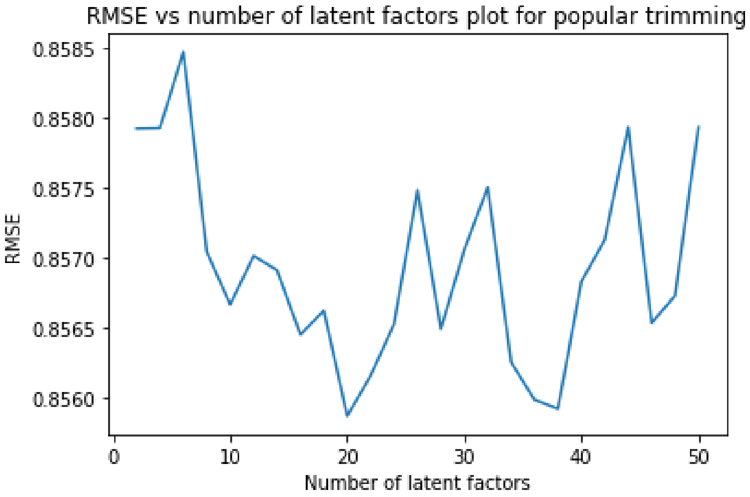
\

B) Use the plot from the previous part to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE. Is the optimal number of latent factors same as the number of movie genres?

The minimum value of RMSE and the corresponding number of latent factors are 0.8653446481705158 and 20
The minimum value of MAE and the corresponding number of latent factors are 0.6639542252974552 and 30
The optimal number of latent components is close to the number of movie genres based on both average RMSE and MAE criteria in the MF with bias collaborative filter, but this is not always true. It is difficult to interpret the latent components in this model and the results are similar for a wide range of latent component counts. Therefore, if we rerun cross-validation with a new random seed, the optimal number of latent factors is likely to be different. It is assumed that the optimal number of factors is 20, which is quite similar to the number of movie genres (19) in this case.

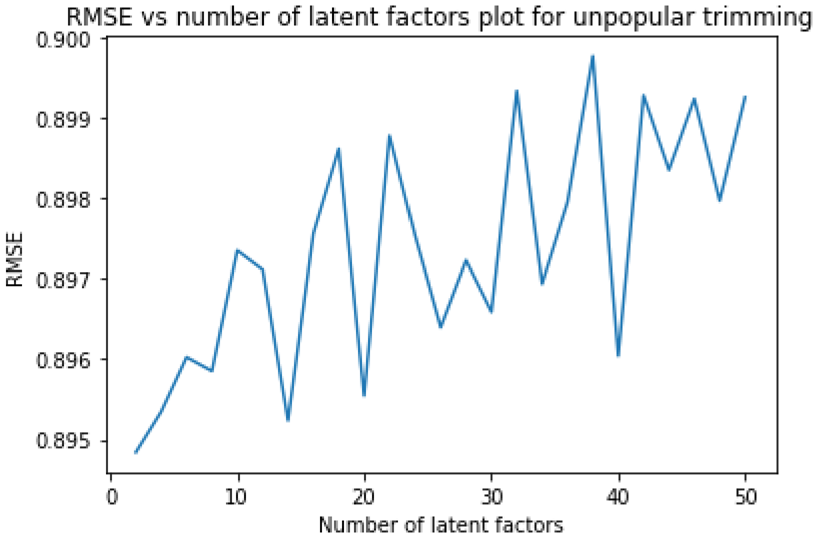
C) Performance on dataset subsets: For each of Popular, Unpopular and High-Variance subsets -
– Design a MF collaborative filter for each trimmed subset and evaluate its performance using 10-fold cross validation. Sweep k (number of latent factors) from 2 to 50 in step sizes of 2, and for each k compute the average RMSE obtained by averaging the RMSE across all 10 folds.
– Plot average RMSE (Y-axis) against k (X-axis); item Report the minimum average RMSE. 9

Popular Trimming:



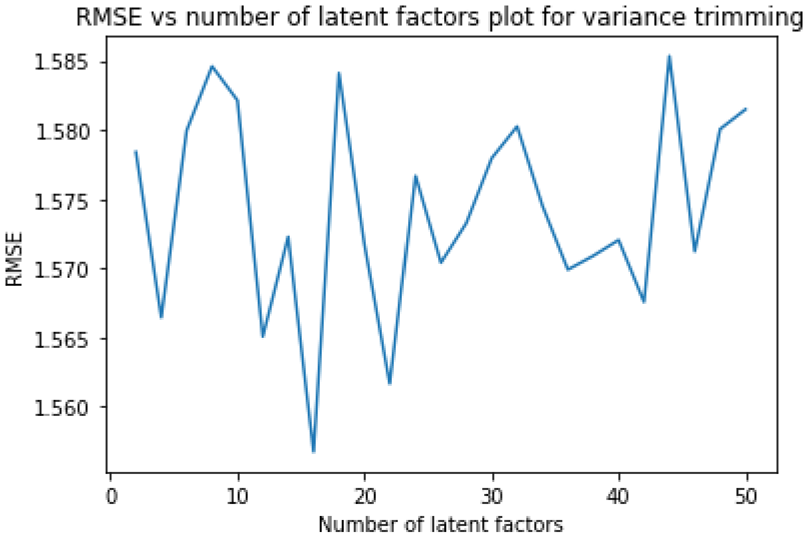
The minimum value of RMSE and the corresponding number of latent factors for popular trimming are 0.8558712762958555 and 20

Unpopular movie trimming:



The minimum value of RMSE and the corresponding number of latent factors for unpopular trimming are 0.8948415072306307 and 2

High Variance trimming:



The minimum value of RMSE and the corresponding number of latent factors for variance trimming are 1.5567397596512522 and 16

Plot the ROC curves for the MF-based collaborative filter and also report the area under the curve (AUC) value as done in Question 6.

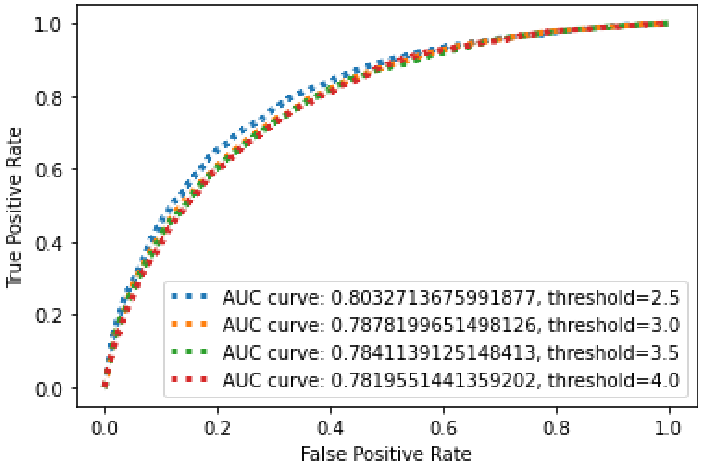


Figure : ROC curves for various thresholds for SVD - NMF user-based CF with 20 latent factors

Area Under the Curve Values (AUC):

Threshold 2.5: 0.8033

Threshold 3: 0.7878

Threshold 3.5: 0.7841

Threshold 4: 0.7819

Popular Trimming:

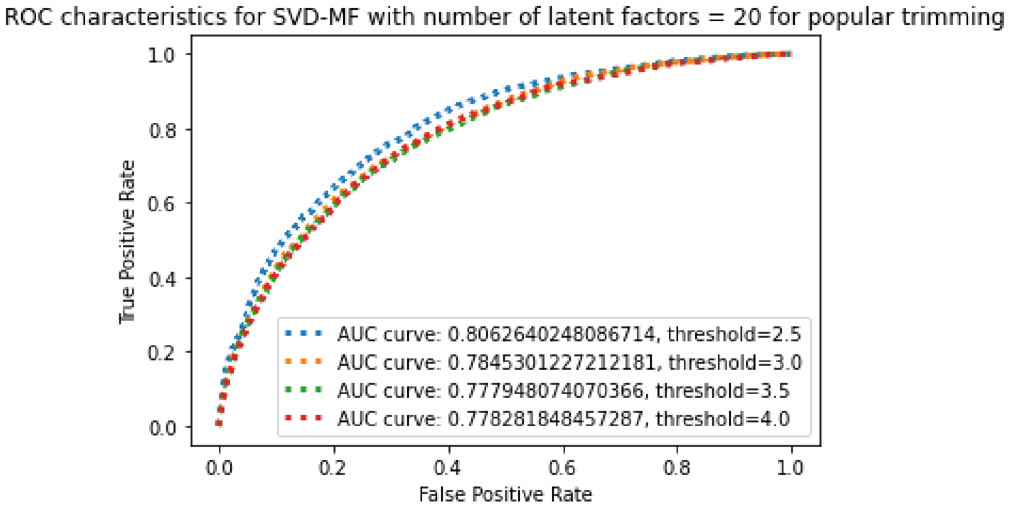


Figure : ROC curves for popular trimming various thresholds for SVD - NMF user-based CF with 20 latent factors

Area Under the Curve Values (AUC):

Threshold 2.5: 0.8062

Threshold 3: 0.7845

Threshold 3.5: 0.7779

Threshold 4: 0.7783

Unpopular Trimming:

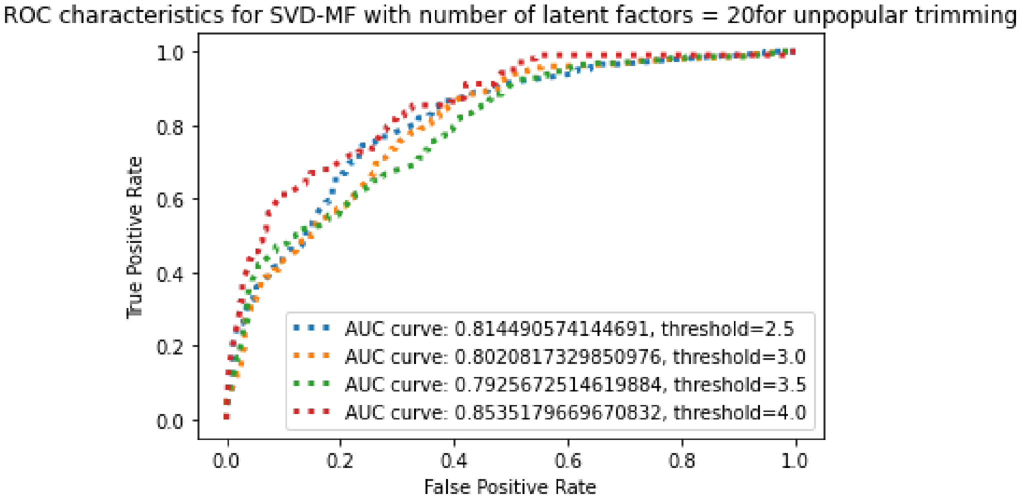


Figure : ROC curves for unpopular trimming various thresholds for SVD - NMF user-based CF with 20 latent factors

Area Under the Curve Values (AUC):

Threshold 2.5: 0.8145

Threshold 3: 0.8021

Threshold 3.5: 0.7925

Threshold 4: 0.8535

High Variance Trimming:

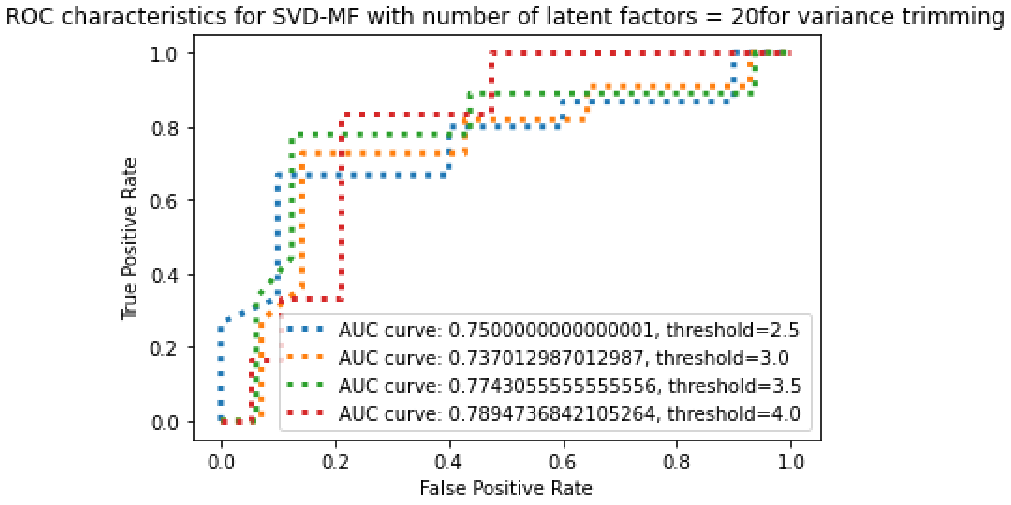


Figure : ROC curves for high variance trimming various thresholds for SVD - NMF user-based CF with 20 latent factors

Area Under the Curve Values (AUC):

Threshold 2.5: 0.75

Threshold 3: 0.7370

Threshold 3.5: 0.7743

Threshold 4: 0.7894

QUESTION 11: Designing a Naive Collaborative Filter:

A. 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.

   We have been tasked with creating a Collaborative Filter using a naive approach for predicting movie ratings based on a given dataset. The naive method involves predicting the rating for a new item as the mean rating of the user on past items, without any training. The prediction function is very simple, as it only involves calculating the mean. This method is expected to have a higher Root Mean Squared Error (RMSE) compared to other Collaborative Filters like neighborhood-based and model-based collaborative filters. Using surprise.AlgoBase, we have found that the average RMSE for this naive method is **1.4390357900986817**

B. 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

   Average RMSE value for Naive Filtering after **Popular movie trimming** is: **1.4598267512126841**
   Average RMSE value for Naive Filtering after **Unpopular movie trimming** is**: 1.3777279723658538**

   Average RMSE value for Naive Filtering after **Variance movie trimming** is**: 1.8616498928224083**

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.

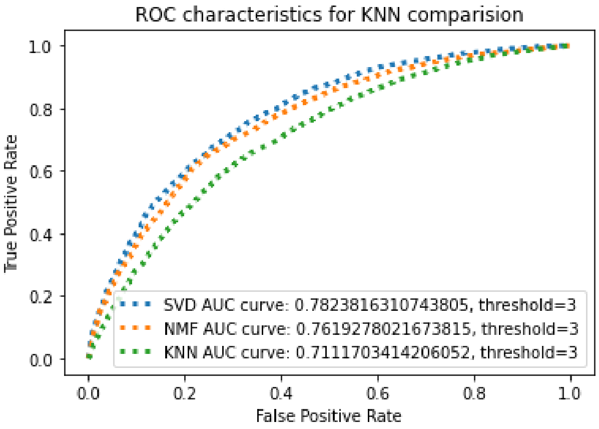


Figure: Comparison of ROC characteristics for MF with bias (SVD), NMF and kNN.

- **k-NN**: 0.77964
- **NMF**: 0.76163
- **MF with bias (SVD)**: 0.78091

According to the above figure, when the threshold is set to 3, MF with bias has the smoothest curve and the largest area under the curve compared to k-NN CF and NMF. This indicates that MF with bias is the best model for predicting movie ratings among the three collaborative filters. The reason for this could be that MF with bias is more versatile compared to NMF as it considers optimization variables for both user and movie bias. This helps in predicting ratings for users who consistently give high or poor ratings. However, KNN performs the worst in terms of movie rating prediction.

MF with bias (SVD) vs NMF:

- MF with bias is better than NMF at representing high-dimensional feature matrices because it does not have any restrictions on the values of U and V. This allows for a deeper factorization process with minimal loss of information. NMF, on the other hand, requires that U and V must be positive, and it has fewer optimal components in U and V than MF with bias. MF with bias creates a hierarchy of geometric basis that is arranged based on their importance, which results in embeddings that have the most important features and characteristics in the ratings matrix positioned higher in the hierarchy. Due to this hierarchical ordering, the embeddings generated by MF with bias are robust to noise and outliers in the ratings. However, NMF does not consider the geometric structure of the ratings matrix.The embeddings generated by MF with bias are distinct and predictable, while the embeddings produced by NMF are not definite and random, with no guarantee of reaching the best U and V every time the function is used. MF with bias uses information about user and movie-specific bias to reduce the impact of outliers and noise by normalizing the ratings accordingly.

MF with bias (SVD) vs k-NN:

- The k-NN algorithm does not model bias information independently for each user or item, making it more vulnerable to outliers and infrequently rated products. Since k-NN directly infers from the sparse ratings matrix, it has poor prediction performance in high-dimensional space due to the curse of dimensionality. This also limits the scalability of the recommender system. In high-dimensional inference, a substantial amount of training data is required for effective operation. Compared to latent-factor models, k-NN is much less versatile because it cannot identify semantic information and relationships within the user-item ratings matrix, and is sensitive to infrequently rated items.

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.

1) Precision:
   The accuracy of a model's predictions is indicated by precision, which is the percentage of recommended items that the user likes out of all the recommendations made. If precision is low, it implies that there are many false positives. Precision at k measures the proportion of relevant items in the top-k recommendations. For example, if the precision at 10 for a top-10 recommendation problem is 90%, it means that 90% of the recommendations are relevant to the user.
2) Recall
   Recall is a measure of the effectiveness of a recommendation model. Recall is the proportion of relevant recommended items among all the items that the user actually likes. When considering the top-k recommendations, recall at k is the percentage of relevant items that appear in the top-k recommendations. For example, if recall at 10 is 20%, this means that 20% of the relevant items are present in the top-10 recommendations.
3) Precision Recall Curve:
   The precision-recall curve is a useful tool for evaluating the effectiveness of a ranked list. To understand the expressions for precision and recall in the ranking context, we need to introduce some notation. S(t) refers to the set of t items recommended to the user, and any items without a ground truth rating are disregarded. G represents the set of items that the user actually likes, which are the true positives.

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

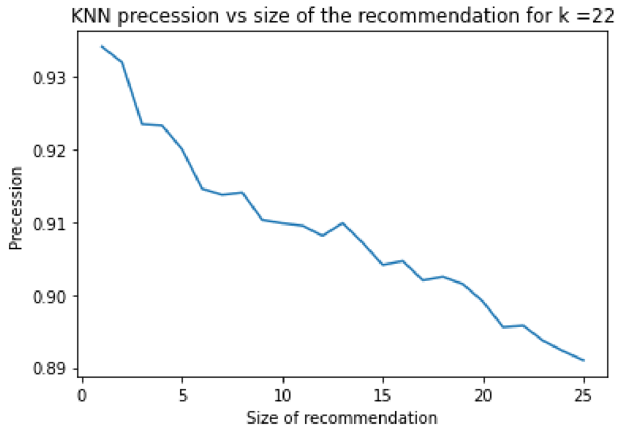
$$Recall(t) = \frac{|S(t) \cap G|}{|G|}$$

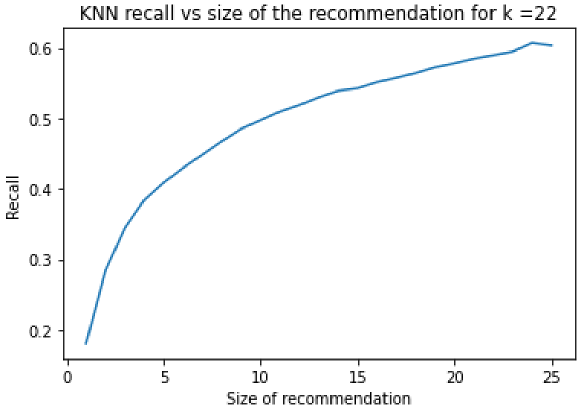
QUESTION 14: Comparing the precision-recall metrics for the different models:

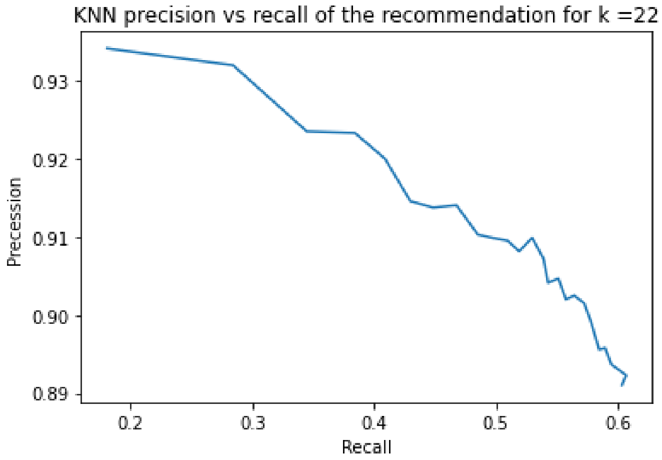
A) For each of the three architectures:

- 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.

KNN





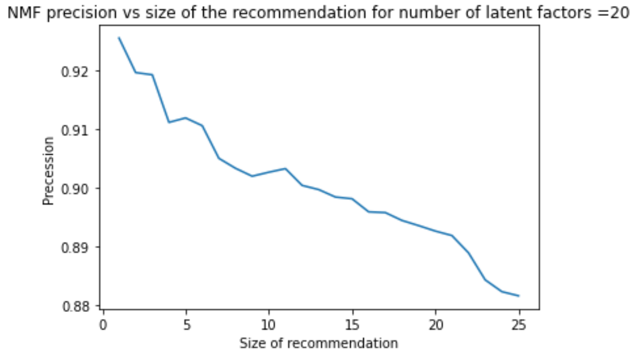


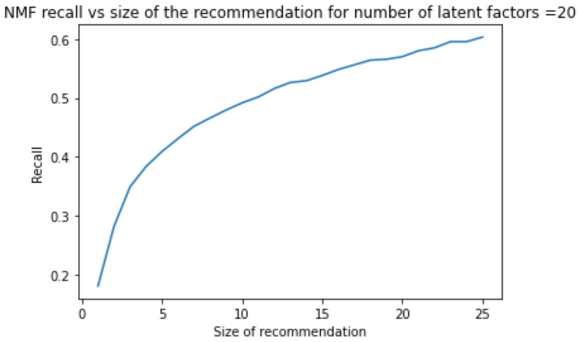
1) Precision is a metric that measures the accuracy of recommended items that are actually liked by the user among all the items suggested to them. As the number of recommended items increases, the precision decreases. This is because when only one item is recommended, it has a higher chance of being a ground-truth positive. However, when more items are recommended, it becomes more challenging for all of them to be relevant to the user, even if they have high predicted ratings close to the threshold. Therefore, with a larger number of recommendations, the precision metric becomes more difficult to optimize.

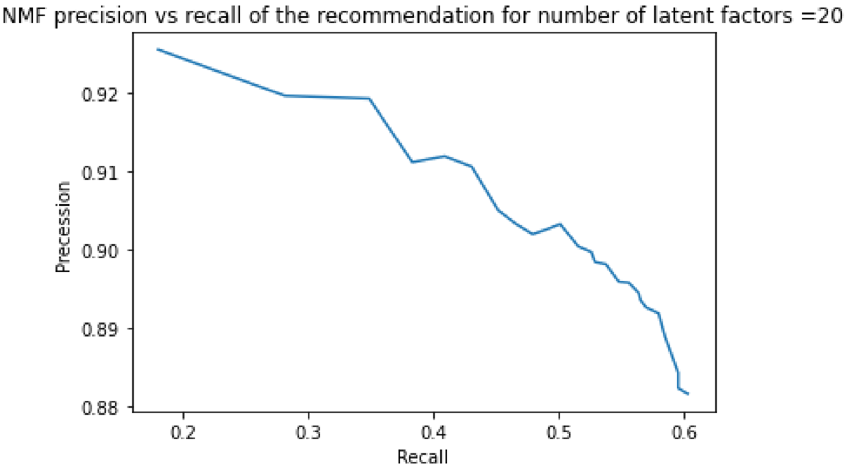
2) Recall evaluates the percentage of recommended items that the user likes among all the items that the user actually likes. As the number of recommended items increases, the probability of recommending more items that are liked by the user also increases. This explains why recall increases as we increase the number of recommended items. We can also understand this increase in recall by looking at the recall formula, which uses a constant denominator representing the number of items liked by the user. As t increases, the intersection between S(t) and G increases, resulting in a steady increase in recall. Additionally, there is a significant increase in recall in the initial stages. For instance, if |G| is 5, recall will be low for t < 5 because we are not even recommending 5 items yet. Therefore, we observe a steep increase in recall for small t. However, once t exceeds |G|, the rate of increase in recall starts to decrease.

3) The relationship between precision and recall is negative, which means that as precision decreases, recall increases and vice versa. This is because precision is inversely proportional to t, while recall is directly proportional to t. Thus, t acts as a latent variable that connects precision and recall, and the resulting curve illustrates the tradeoff between the two. It is impossible to maximize both precision and recall at the same time.

NMF


NMF precision vs size of the recommendation for number of latent factors =20


NMF recall vs size of the recommendation for number of latent factors =20

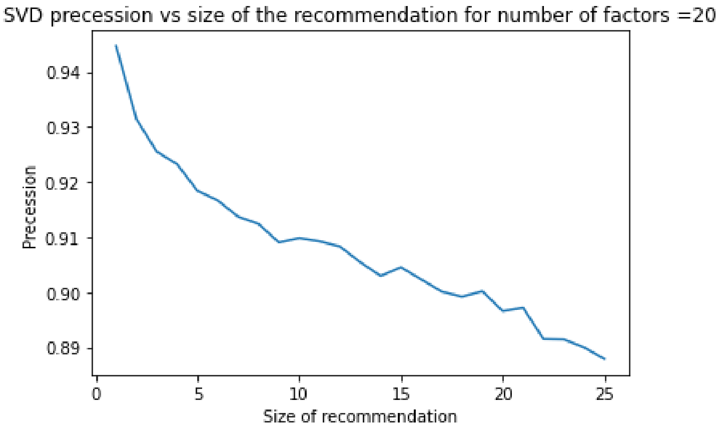

NMF precision vs recall of the recommendation for number of latent factors =20

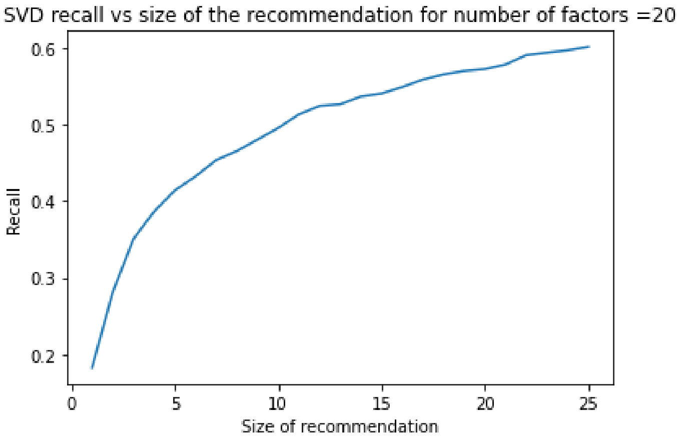
1) When evaluating a recommendation system, precision indicates how many of the recommended items are actually liked by the user. As the number of recommended items increases, precision tends to decrease. This is because when only one item is recommended, the system can choose the one with the highest predicted rating, which is more likely to be liked by the user. However, as the number of recommendations increases, it becomes more difficult to accurately predict all of the items that the user will like, even if the predicted ratings are close to the threshold of liking. Therefore, as the number of recommendations grows, the precision of the system tends to decline.

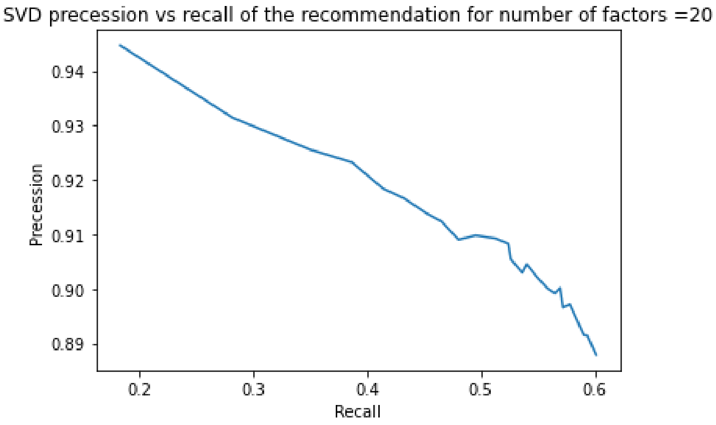
2) To put it differently, recall measures the accuracy of recommended items among the user's liked items. As we recommend more items, the chances of getting more items from the user's preference list also increase. This is why recall increases as t increases. Mathematically, recall's denominator, |G|, is a constant that represents the number of items liked by the user. As t increases, the intersection between S(t) and G increases, resulting in a monotonic rise in recall. Moreover, recall shows a significant improvement at the beginning stages because if the |G| is 5, then t < 5 will have low recall since we are not recommending enough items. Therefore, we see a sharp rise in recall for small t. Once t is greater than |G|, the increase in recall starts to decrease.

3) Precision and recall are negatively correlated, meaning that as one metric increases, the other decreases. The relationship between precision and t is inversely proportional, whereas the relationship between recall and t is directly proportional. Thus, t serves as a bridging factor between precision and recall, and is represented as a latent variable in the final subplot. This implies that there is a trade-off between precision and recall, and we cannot optimize both metrics simultaneously.

SVD



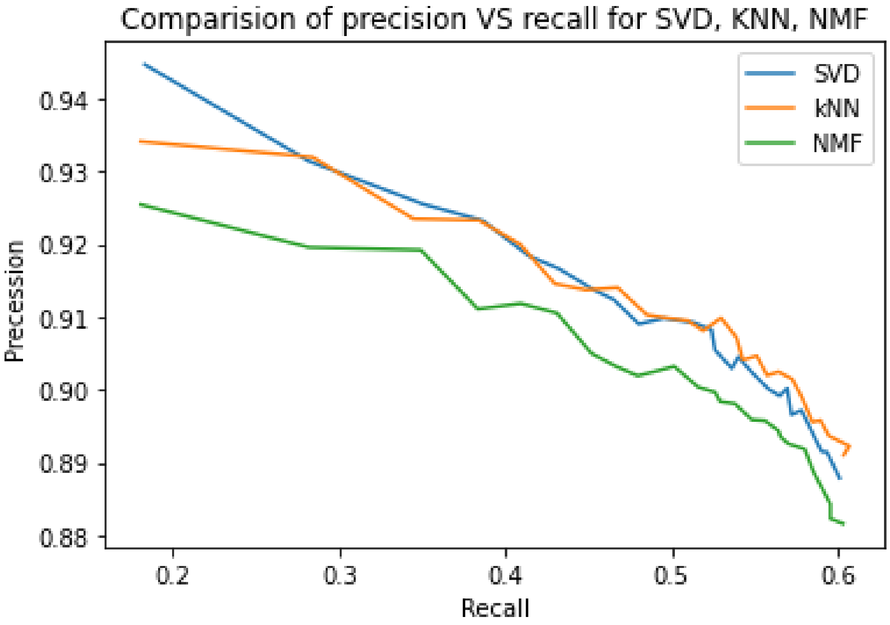




1) Precision is a measure of the accuracy of the recommended items that a user actually likes among all the recommendations made by the system. As the number of recommended items increases, precision tends to decrease. This can be explained by the fact that when a recommendation system is asked to suggest only one movie to a user, it can easily select the one with the highest predicted rating, which is likely to be appreciated by the user. However, as the number of recommended movies increases, for example, to 15, it becomes increasingly difficult for all 15 movies to be appreciated by the user, even if they have predicted ratings close to the threshold of 3. Thus, with a larger number of recommended items, it becomes more challenging to maintain high precision.

2) Recall is a measure of the percentage of correct recommendations among items that the user has liked. Increasing the number of recommended items, or "t", increases the likelihood of recommending more items that the user likes, leading to an increase in recall. This can be explained mathematically by the fact that the denominator in the recall formula represents the number of items the user likes, which is constant. As t increases, the intersection between the recommended items and the user's liked items also grows, leading to a monotonic increase in recall. However, there is a significant improvement in recall at the early stages of t, because if the number of items the user likes is, say, 5, recall will be low for t < 5, since fewer than 5 items are recommended. Therefore, there is a sharp increase in recall for small values of t. After t exceeds the number of items the user likes, the increase in recall slows down.

3) Precision and recall are negatively correlated, meaning that an increase in one usually results in a decrease in the other. This is because as the number of recommended items increases, precision tends to decrease while recall tends to increase. Therefore, t acts as a mediator between precision and recall, represented as a curve in the final subplot. Due to this trade-off, it is not possible to maximize both precision and recall at the same time.



The precision-recall curves of all three models indicate an inverse relationship between average precision and recall. However, MF with bias provides the best performance, followed by kNN and NMF. This suggests that average recall can be increased with only a small decrease in average precision for MF with bias. To evaluate the relevance of recommendation lists generated by different collaborative filters, we can set precision (or recall) and compare the recall (or precision) of different models under the same precision value (or recall). The precision-recall curve's area under the curve, like AUC for the ROC curve, can be used as a scalar metric to compare the performance of different recommendation systems. The figure above shows the precision-recall curves, indicating that MF with bias is the best recommendation system, while NMF is the worst, similar to the results obtained from ROC curves. The same reasoning can be applied here

MF with bias (SVD) vs NMF:

MF with bias is better suited to represent higher-dimensional feature matrices compared to NMF as it allows for deeper factorization with minimal loss of information. NMF, on the other hand, requires positive values for its matrices U and V and has fewer ideal components in U and V compared to MF with bias. MF with bias generates a hierarchical and geometric basis that orders the embeddings by relevance, resulting in embeddings with the most relevant traits in the ratings matrix being further up in the hierarchy. This feature ordering makes the embeddings produced by MF with bias resistant to outliers and noise in the ratings. On the other hand, NMF does not take into account the geometry of the ratings matrix.The embeddings produced by MF with bias are unique and deterministic, whereas those produced by NMF are non-unique and stochastic, with no guarantee of convergence to the optimal U and V each time the function is applied.

MF with bias (SVD) vs k-NN:

The k-NN model does not model bias information separately for each user or item, making it more prone to being influenced by outliers and items that are rarely rated. Additionally, because k-NN directly infers on the sparse ratings matrix, it has lower prediction accuracy in high-dimensional space and is less scalable. This is due to the fact that high-dimensional inference requires a large amount of training data to perform well. Compared to latent-factor models, k-NN is less able to identify semantic information and relationships within the user-item ratings matrix, and it is also more sensitive to items that are rarely rated, making it less generalizable.