Project 3: Recommender Systems

Tania Rajabally
UID: 806153219

QUESTION 1:

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

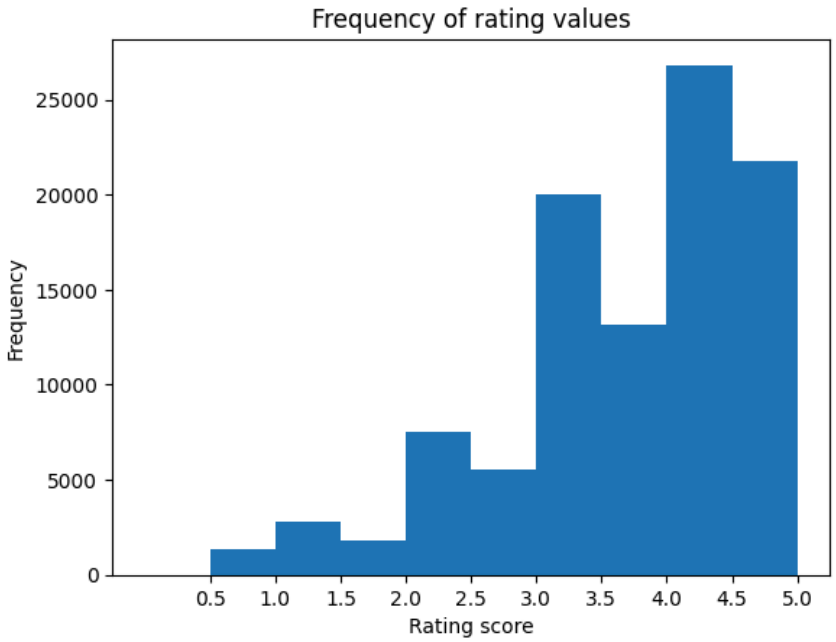
A. Compute the sparsity of the movie rating dataset:

Sparsity = Total number of available ratings / Total number of possible ratings

Sparsity: 0.016999683055613623

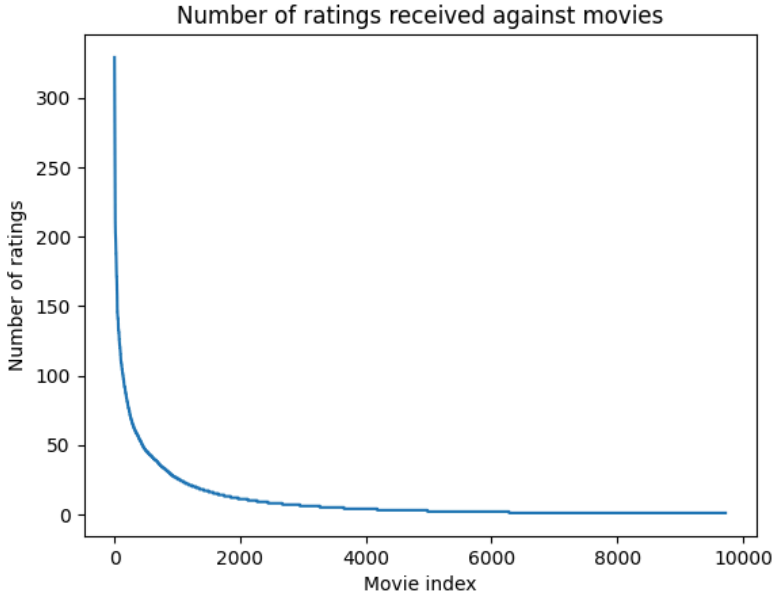
We can see that the sparsity is low. This means that many movie ratings are missing. From the dataset, we can see that there are many movies compared to the number of users and not all the users have watched the majority of the movies. This implies that the Rating matrix R is sparse in nature.

B. Plot a histogram showing the frequency of the rating values: Bin the raw rating values into intervals of width 0.5 and use the binned rating values as the horizontal axis. Count the number of entries in the ratings matrix R that fall within each bin and use this count as the height of the vertical axis for that particular bin. Comment on the shape of the histogram.



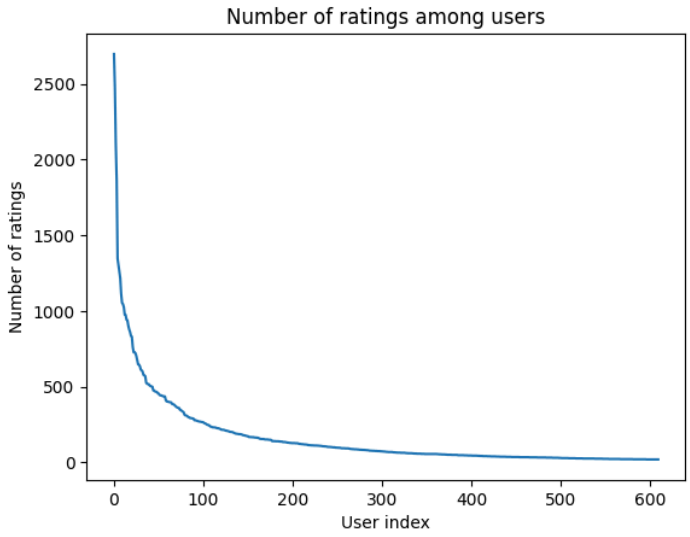
From the above histogram, we can see that most of the ratings lie between 3 and 5. Very few ratings are below 3. 4 has received the most ratings. The distribution is more concentrated towards the right. We can also see that the integer values of the ratings have received more ratings as compared to the decimal values.

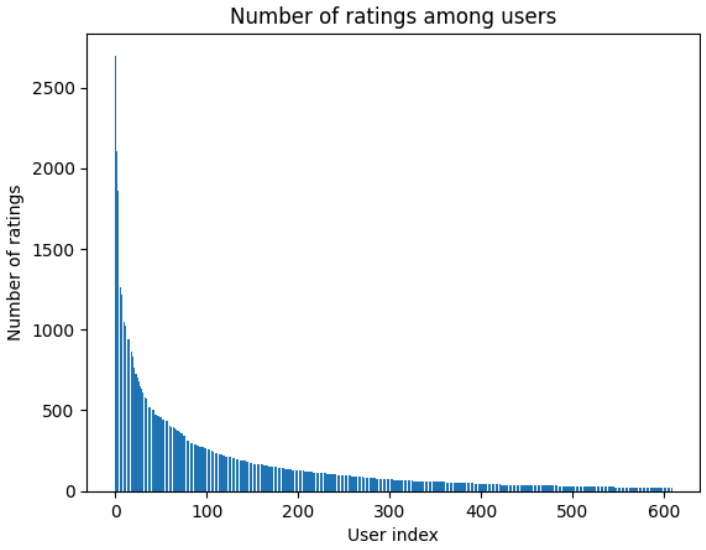
C. Plot the distribution of the number of ratings received among movies: The X-axis should be the movie index ordered by decreasing frequency and the Y -axis should be the number of ratings the movie has received; ties can be broken in any way. A monotonically decreasing trend is expected.



The curve is monotonically decreasing as approximately 500 movies received less than 50 ratings. This confirms that the matrix R is sparse.

D. Plot the distribution of ratings among users: The X-axis should be the user index ordered by decreasing frequency and the Y -axis should be the number of movies the user has rated. The requirement of the plot is similar to that in Question C.



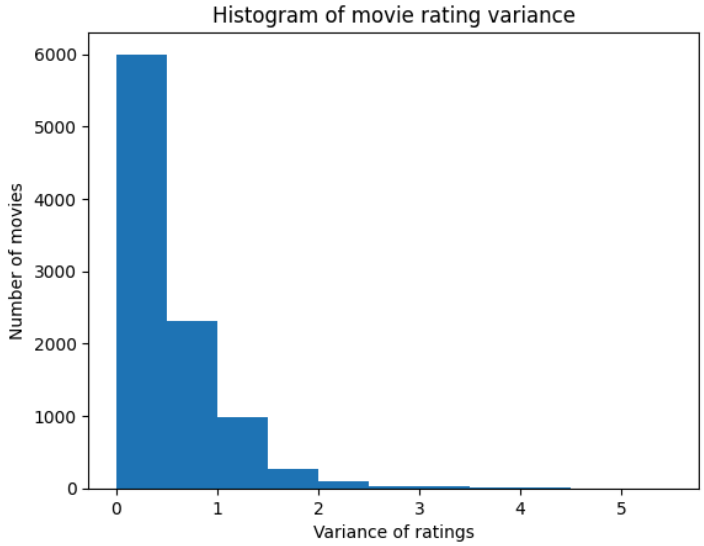


This graph is also monotonically decreasing. Only approximately 50 users have given ratings to 500 movies or more. The remaining have given ratings to less than 500 movies.

E. Discuss the salient features of the distributions from Questions C,D and their implications for the recommendation process.

We can see that both the graphs are monotonically decreasing in nature. For the number of ratings against movies, we notice that the curve is monotonically decreasing as approximately 500 movies received less than 50 ratings. For the number of ratings against users, we notice that only approximately 50 users have given ratings to 500 movies or more. The remaining have given ratings to less than 500 movies. This confirms that the matrix R is sparse. This may imply that the users generally choose movies with high ratings and watch that. This distribution is not very good since the data is very sparse. With most of the elements in the representation being 0, they do not contribute to any information during the model training. This may lead to a poor performing model. The parameters may overfit with just a few movies that have ratings.

F. Compute the variance of the rating values received by each movie: Bin the variance values into intervals of width 0.5 and use the binned variance values as the horizontal axis. Count the number of movies with variance values in the binned intervals and use this count as the vertical axis. Briefly comment on the shape of the resulting histogram.



We can see that most movies have very low variance in their ratings. This makes sense as most users watch movies with good ratings. Very few movies have a variance of 2.5 or higher. The ratings of the movie do not vary significantly.

<u>QUESTION 2:</u>

Understanding the Pearson Correlation Coefficient:

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

The formula of u_u in terms of I_u and r_{uk} is:

$$\mu_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{ui}$$

Where u_u = Mean rating for user u computed using her specified ratings;
I_u = Set of item indices for which ratings have been specified by user u;
r_{uk} = Rating of user u for item k.

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

$I_u \cap I_v$ represents the intersection of the sets I_u and I_v which means the intersection of the movies rated by user u and user v. This intersection represents the set of movies rated by both the users u and v.
Yes, this set i.e the intersection can be 0 if both the users, u and v, have never rated the same movie i.e there are no common movies rated by user u and user v.

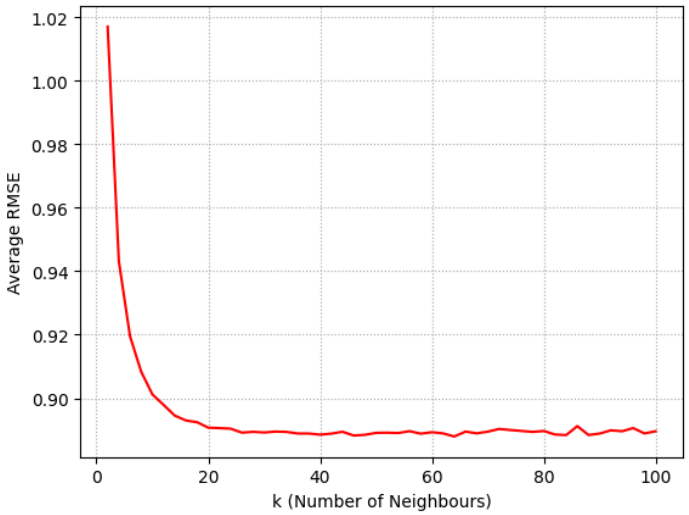
QUESTION 3:

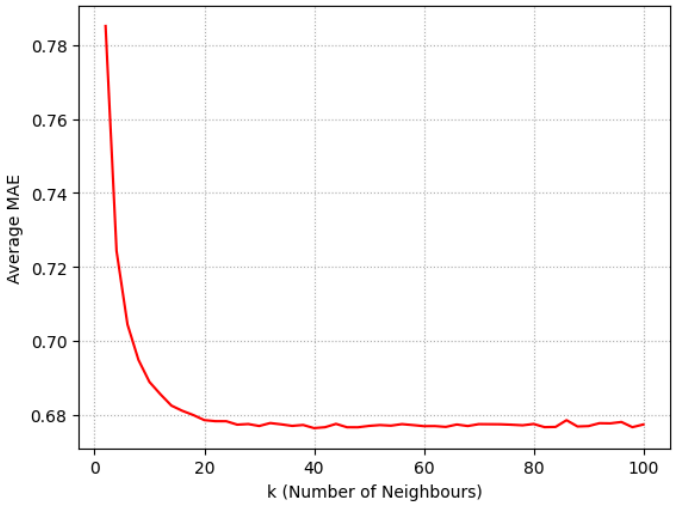
Understanding the Prediction function: Can you explain the reason behind mean-centering the raw ratings (rvj − μv) in the prediction function? (Hint: Consider users who either rate all items highly or rate all items poorly and the impact of these users on the prediction function.)

Mean centering the raw ratings is important to normalize the ratings relative to the user baseline. This normalization helps in comparing ratings across different users who may have different rating scales or tendencies. It ensures that the prediction is not biased by users who generally rate items higher or lower than others. It also takes into account the individual bias of the users. Mean-centering allows the collaborative filtering algorithm to focus on the relative differences in ratings rather than absolute values. In sparse datasets where users have rated only a small subset of items, mean-centering helps in making predictions even when there is limited overlap between users or items. Overall, mean-centering the raw ratings in the prediction function improves the robustness, accuracy, and generalization ability of collaborative filtering algorithms by accounting for user biases, normalizing ratings, and reducing the impact of sparsity in the dataset.

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).





The two graphs above show the kNN Collaborative filter. We can see that the value is monotonically decreasing until a point after which we do not see any significant decrease in the average values. This trend is observed for both the graphs.

QUESTION 5:

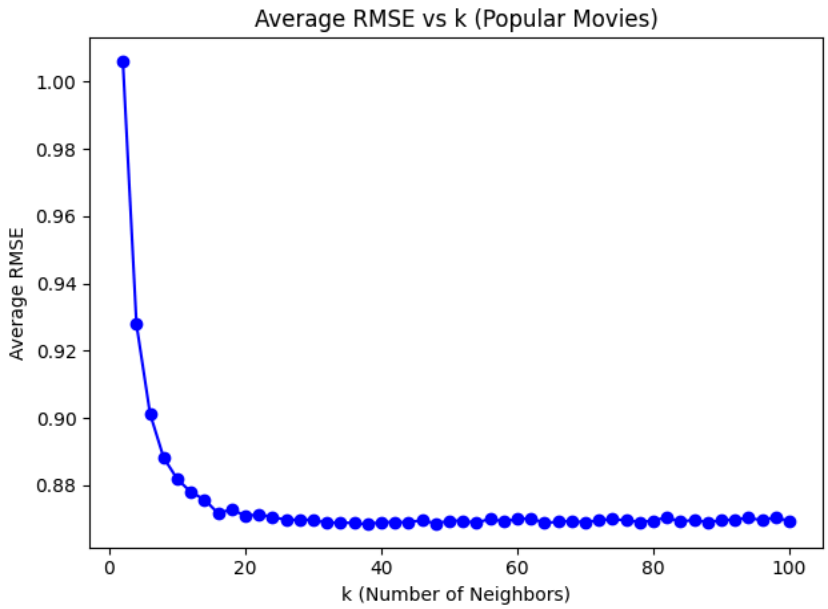
Use the plot from question 4, to find a 'minimum k'. Note: The term 'minimum k' in this context means that increasing k above the minimum value would not result in a significant decrease in average RMSE or average MAE. If you get the plot correct, then 'minimum k' would correspond to the k value for which average RMSE and average MAE converge to a steady-state value. Please report the steady state values of average RMSE and average MAE.

We find the values as below:
Min value of RMSE: 0.8883963716107726
K Value corresponding to min value of RMSE: 42
Min value of MAE: 0.6761575337414852
K Value corresponding to min value of MAE: 42

QUESTION 6:

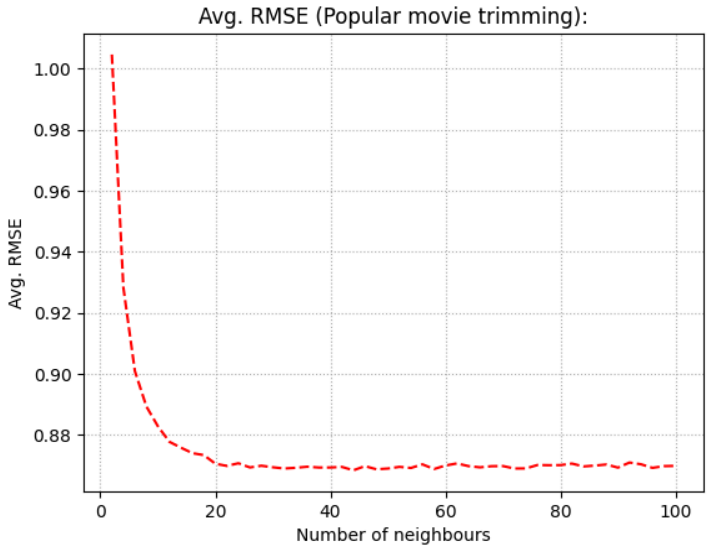
Within EACH of the 3 trimmed subsets in the dataset, design (train and validate):
A k-NN collaborative filter on the ratings of the movies (i.e Popular, Unpopular or High-Variance) and evaluate each of the three models' performance using 10-fold cross validation:

- **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 Data
  Using cross validate:



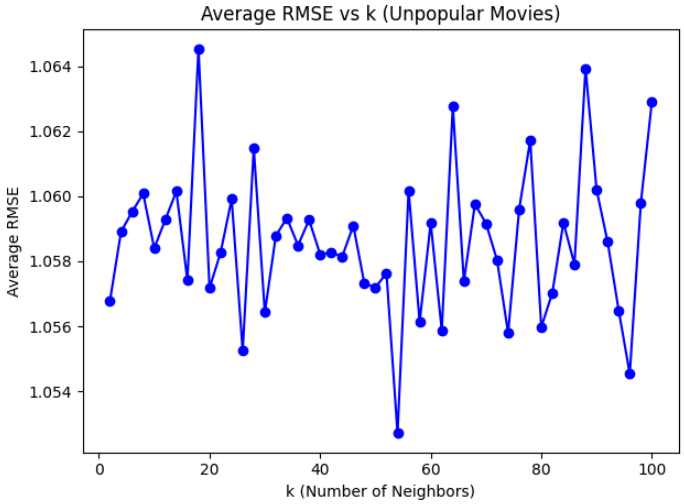
Minimum Average RMSE for Popular Movies: 0.8684798204535076 for k = 38

Using KFold:



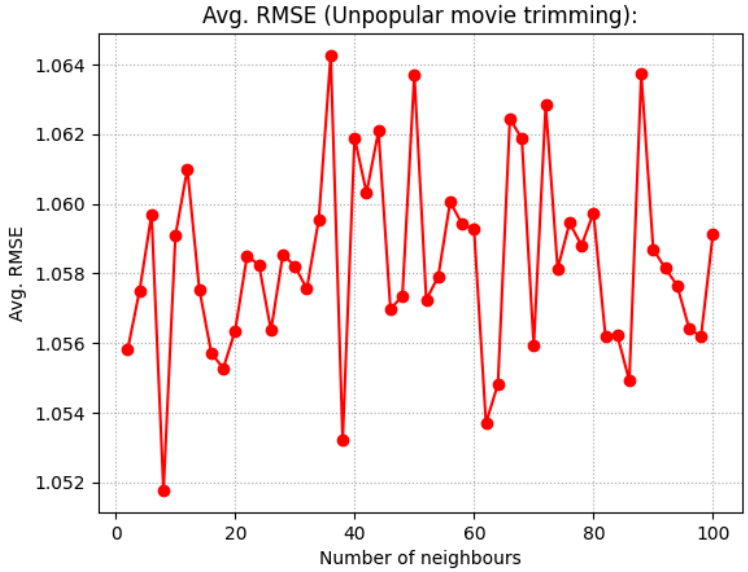
Minimum Average RMSE for Popular Movies: 0.8684169363140496 for k = 44

Unpopular Data:
Using cross validate:



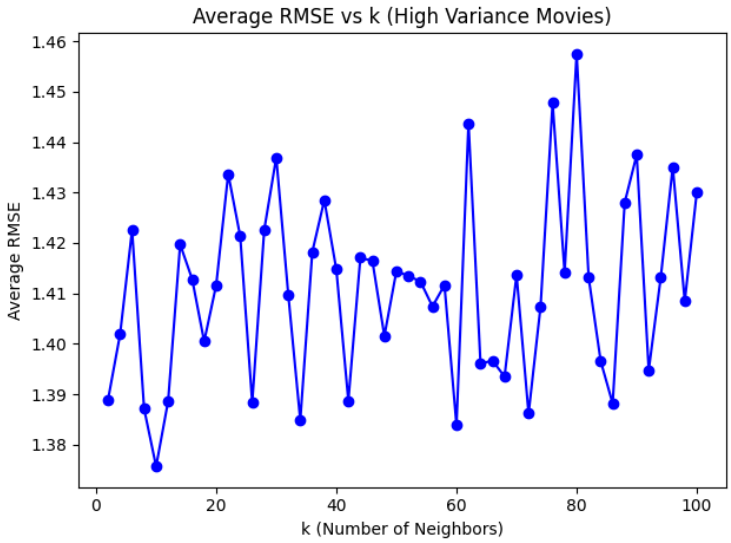
Minimum Average RMSE for Unpopular Movies: 1.0527101848059413 for k = 54

Using KFold:



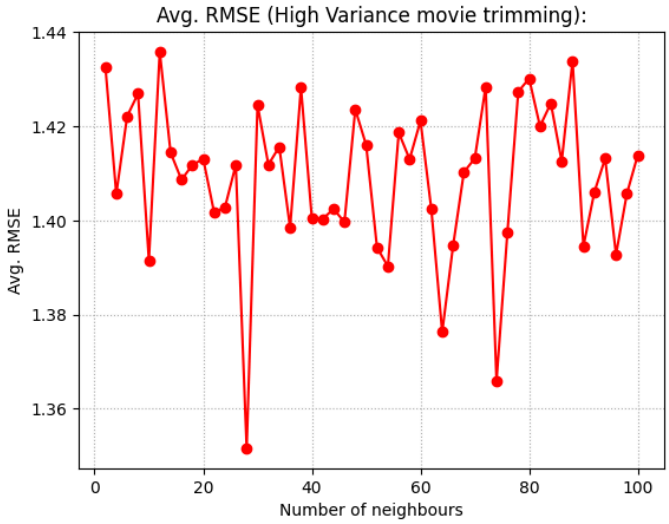
Minimum Average RMSE for Unpopular Movies: 1.051767399467695 for k = 8

High Variance Data:
Using cross validate:



Minimum Average RMSE for High Variance Movies: 1.37570326107562 for k = 10

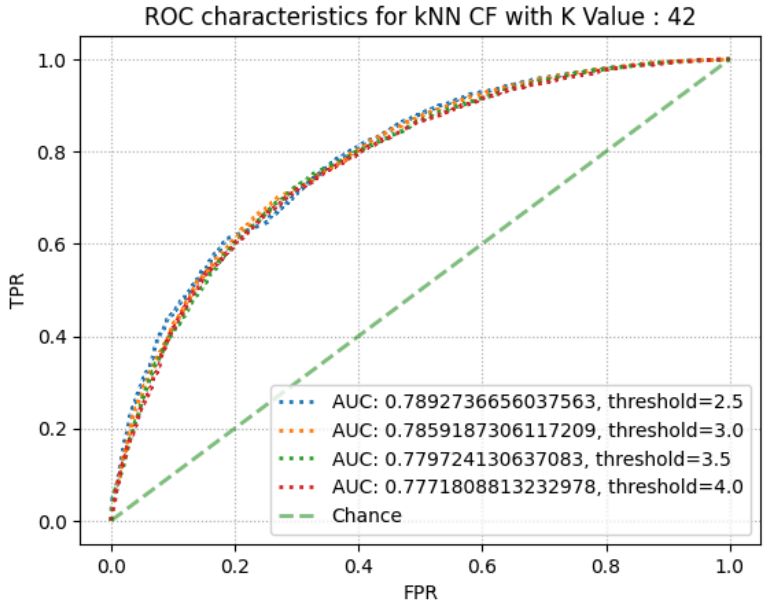
Using KFold:



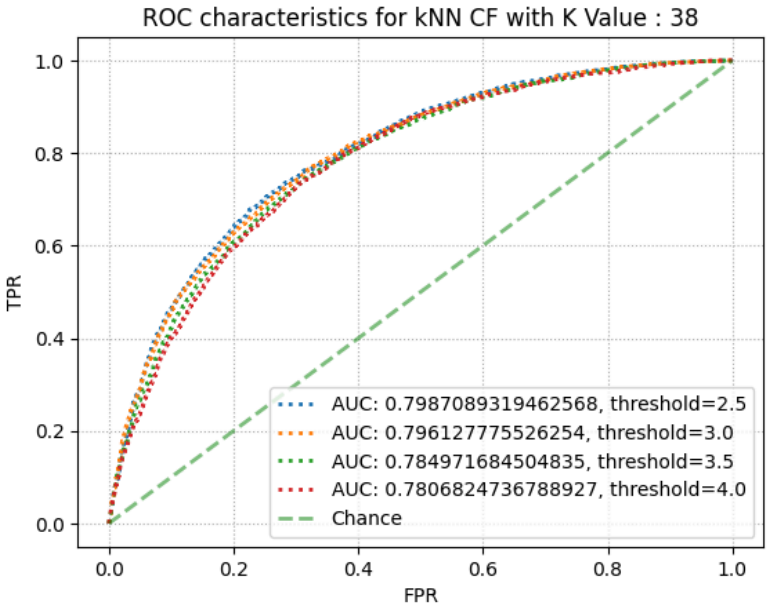
Minimum Average RMSE for High Variance Movies: 1.3515786593368913 for k = 28

- **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 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.**

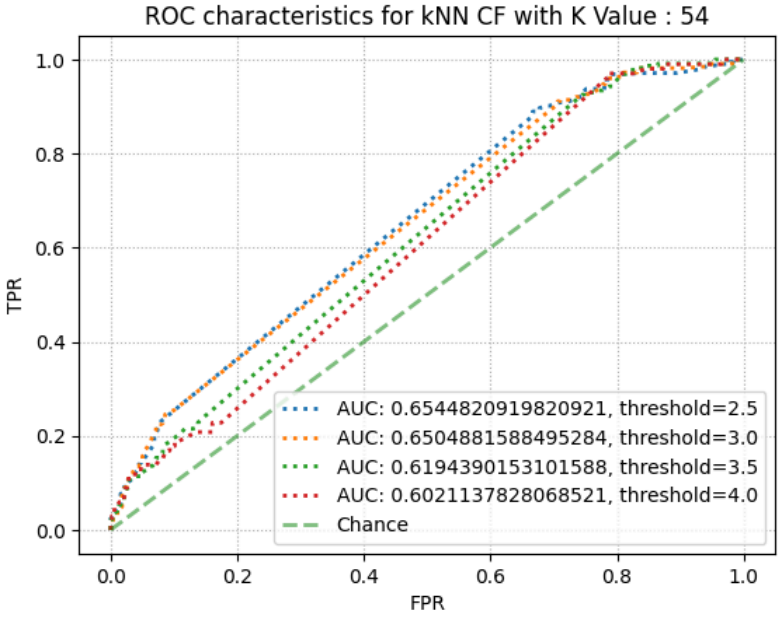
Untrimmed Data:



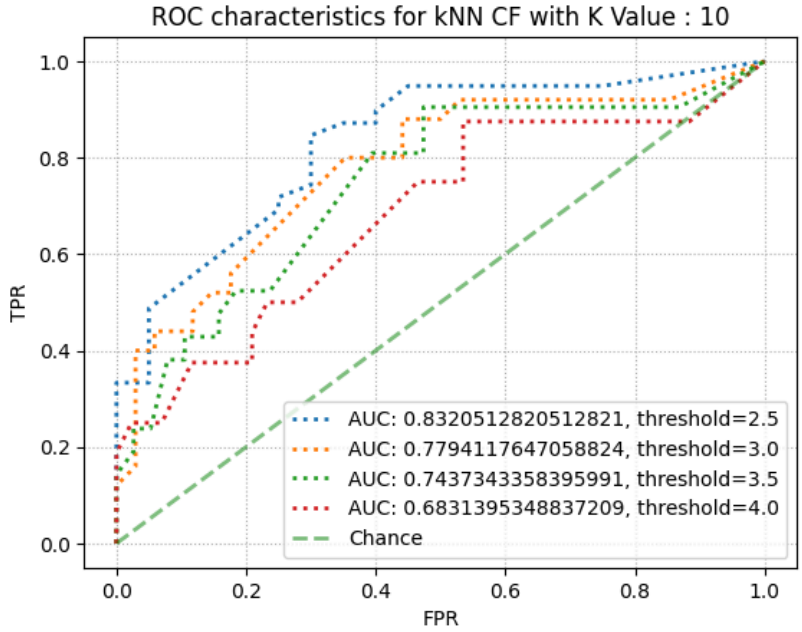
Popular Data:



Unpopular Data:



High Variance Data:



QUESTION 7:

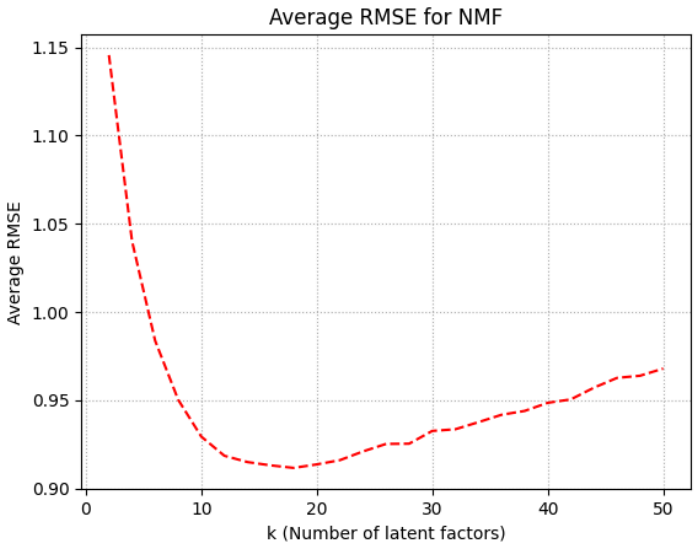
Understanding the NMF cost function: Is the optimization problem given by equation 5 convex? Consider the optimization problem given by equation 5. For U fixed, formulate it as a least-squares problem.

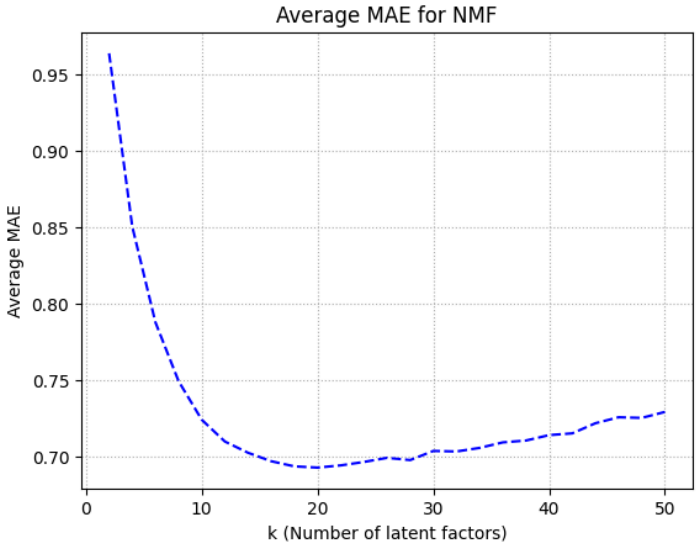
No, the optimization problem is not convex. Both matrices U and V are unknown variables at the same time. If we fix one of the two variables and then try and solve for the other variable, then the given optimization problem can be formulated to a least squares problem which is a convex problem. Therefore, 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. If we fix U and solve for matrix V, the least squares problem will be:

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

QUESTION 8:

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 (Y- axis) against k (X-axis). For solving this question, use the default value for the regularization parameter.**





We can see that the error first decreases with an increase in k (Number of latent factors). But, after a point, it starts increasing linearly. This increasing value of k does not help significantly with the performance. This is because increasing k leads to an increase in the dimension and the rating matrix becomes noisy.

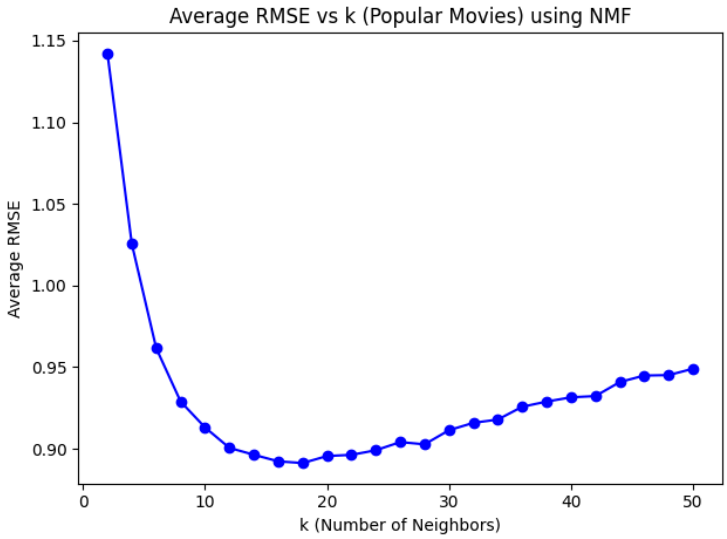
B. **Use the plot from the previous part to find the optimal number of latent factors. Optimal number of latent factors is the value of k that gives the minimum average RMSE or the minimum average MAE. Please report the minimum average RMSE and MAE. Is the optimal number of latent factors same as the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

Min value of RMSE: 0.9143005748405425
K Value for NMF corresponding to min value of RMSE: 16
Min value of MAE: 0.6942119087641884
K Value for NMF corresponding to min value of MAE: 24
There are 19 genres of movies in the dataset. The optimal value of k we have obtained is very close to this number.

C. **Performance on trimmed dataset subsets: For each of Popular, Unpopular and High- Variance 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.**

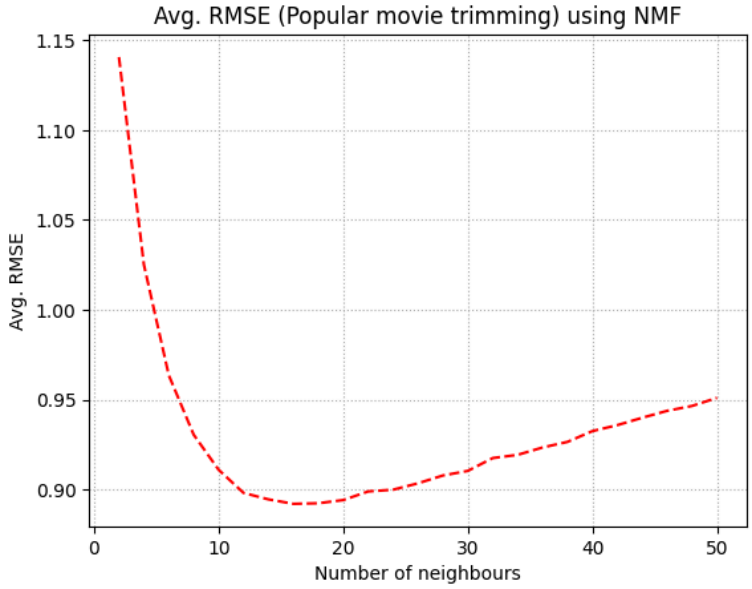
Popular Data:

Using cross validate:



Minimum Average RMSE for Popular Movies using NMF: 0.8912869916751903 for k = 18

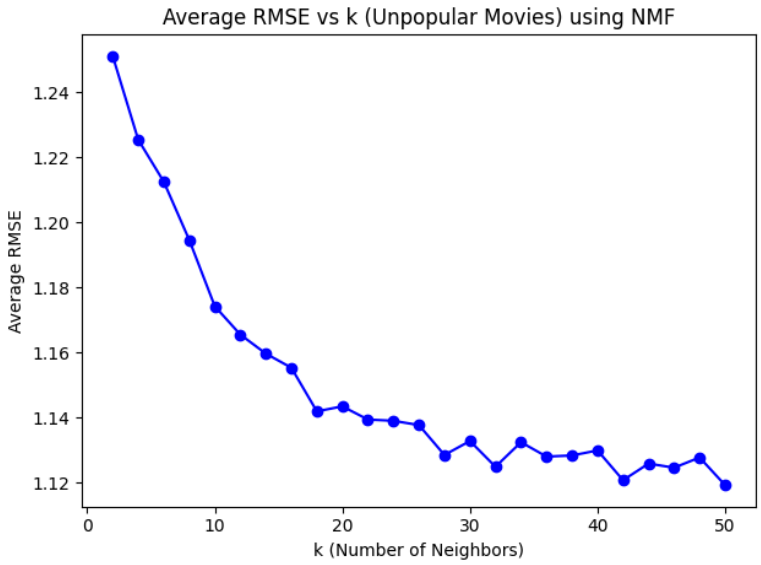
Using KFold:



Minimum Average RMSE for Popular Movies using NMF: 0.892189371583061 for k = 16

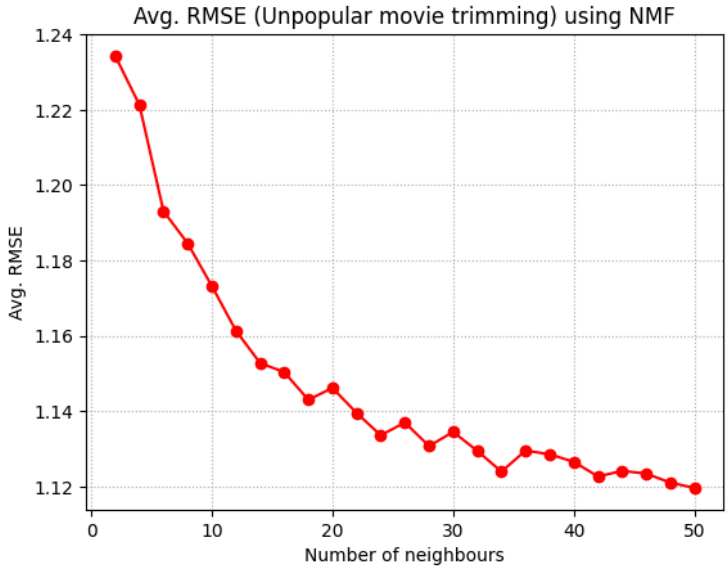
Unpopular Data:

Using cross validate:



Minimum Average RMSE for Unpopular Movies using NMF: 1.1191761795398452 for k = 50

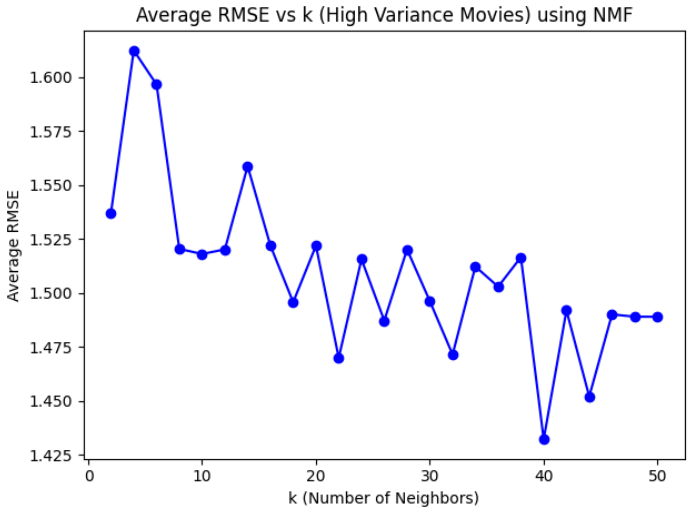
Using KFold:



Minimum Average RMSE for Unpopular Movies using NMF: 1.1196456867107254 for k = 50

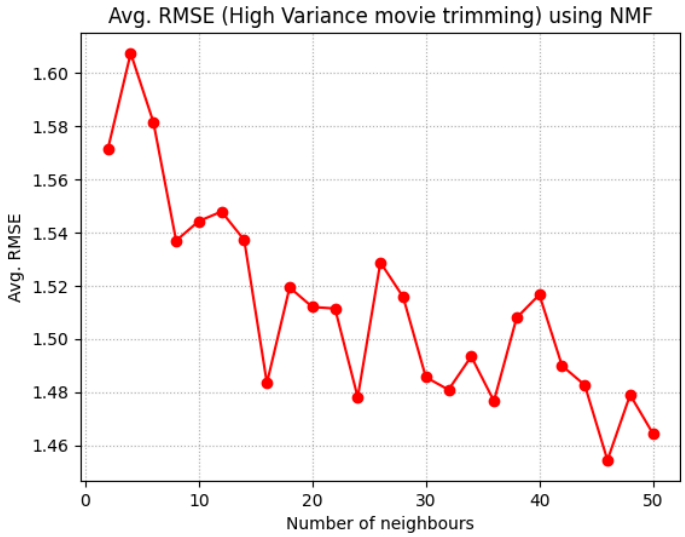
High Variance Data:

Using cross validate:



Minimum Average RMSE for High Variance Movies using NMF: 1.4322523568216146 for k = 40

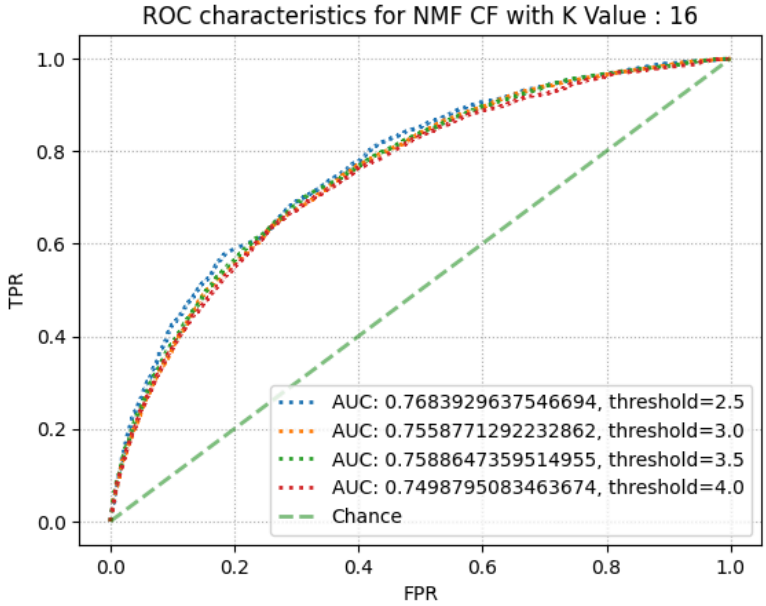
Using KFold:



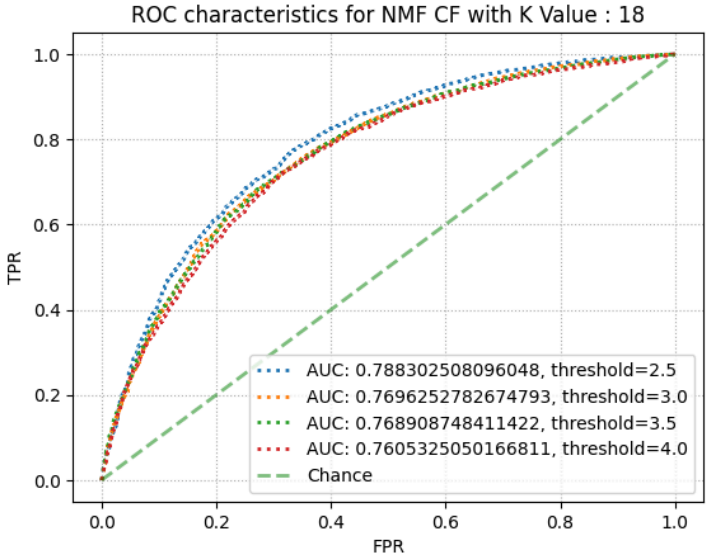
Minimum Average RMSE for High Variance Movies using NMF: 1.454408430825764 for k = 46

- **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.**

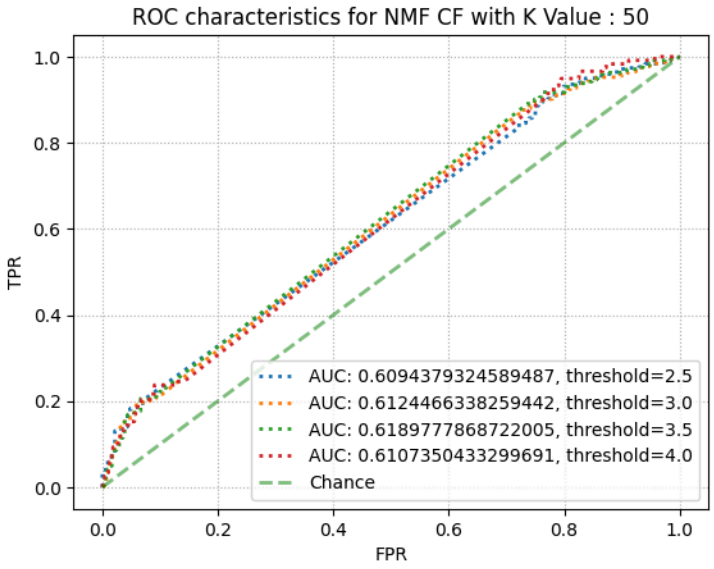
Untrimmed Data:



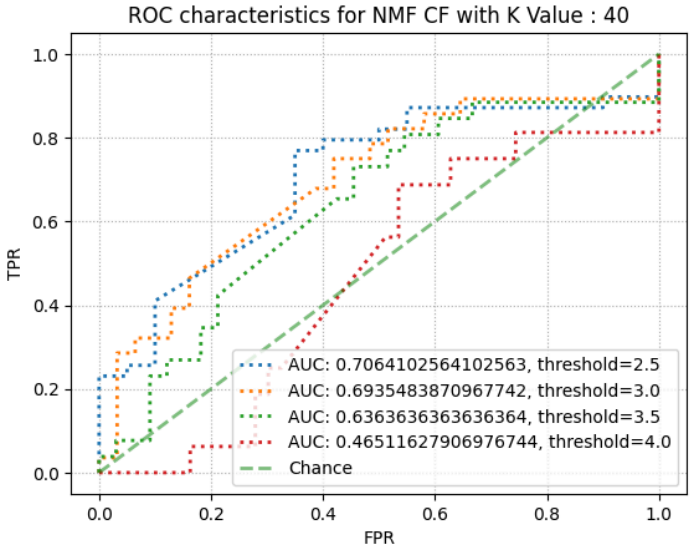
Popular Data:



Unpopular Data:



High Variance Data:



<u>QUESTION 9:</u>

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?

For k=20, we pick up the value of V, sort the movies in descending order and report the genres of the top 10 movies as below:

---------- Value of k: 0 ----------

Action|Adventure|Drama

Comedy|Drama|Romance
Sci-Fi|Thriller
Adventure|Drama|Romance
Action|Crime|Thriller
Comedy|Drama
Comedy|Drama
Comedy|Drama
Drama|Film-Noir|Thriller
Comedy|Drama|Fantasy
---------- Value of k: 1 ----------
Action|Drama|Sci-Fi|Thriller
Fantasy|Musical|Romance
Drama|Musical
Drama
Comedy|Drama|Fantasy|Mystery|Romance
Comedy|Romance
Comedy
Animation|Children
Drama
Comedy|Crime|Horror
---------- Value of k: 2 ----------
Comedy
Comedy|Drama
Comedy|Drama|Romance
Comedy|Crime
Action|Crime|Thriller
Crime|Drama
Drama
Comedy|Drama|Romance
Comedy|Drama|Romance|Western
Comedy|Romance
---------- Value of k: 3 ----------
Children|Fantasy|Musical
Comedy|War
Romance
Sci-Fi|Thriller
Comedy|Sci-Fi
Mystery|Thriller
Comedy
Drama|Fantasy
Drama|Musical|Romance
Adventure|Comedy|Romance
---------- Value of k: 4 ----------
Comedy|Romance
Comedy|Documentary
Drama|Fantasy
Comedy|Crime

Drama|Thriller
Drama
Crime|Drama
Drama
Comedy|Drama|Romance
Comedy|Drama|Romance
---------- Value of k: 5 ----------
Comedy|Crime
Comedy|Drama|Fantasy|Romance
Action|Sci-Fi|Thriller|Western
Sci-Fi|Thriller
Crime|Drama|Thriller
Drama|Horror|Thriller
Comedy|Drama
Comedy|Fantasy|Romance
Documentary
Action|Drama|Thriller
---------- Value of k: 6 ----------
Action|Adventure|Animation|Drama|Fantasy
Drama|Romance
Action|Adventure|Sci-Fi
Crime|Drama|Mystery|Thriller
Action|Crime|Drama|Thriller
Drama
Comedy|Crime|Mystery|Thriller
Comedy
Crime|Drama|Western
Adventure|Children
---------- Value of k: 7 ----------
Action|Adventure|Drama|War
Action|Adventure|Comedy|Crime|Thriller
Crime|Drama
Documentary
Animation|Comedy|War
Horror
Documentary
Drama
Comedy|Drama|Romance
Comedy|Drama|Romance
---------- Value of k: 8 ----------
Crime|Mystery
Drama|Horror
Adventure|Children|Fantasy|Sci-Fi
Action|Sci-Fi|Thriller
Action|Adventure|Drama|Thriller
Drama
Comedy|Romance

Drama|Romance|War
Comedy|Sci-Fi
Comedy
---------- Value of k: 9 ----------
Drama|Romance
Comedy|Documentary|Drama|Romance
Thriller
Horror
Crime|Drama
Comedy|Drama
Comedy|Horror
Crime|Drama|Thriller
Comedy
Action|Comedy
---------- Value of k: 10 ----------
Drama
Action
Drama
Sci-Fi
Action|Crime|Drama
Action|Adventure|Crime|Thriller
Drama|Film-Noir|Thriller
Action|Comedy|Crime|Drama
Comedy|Horror
Horror|Thriller
---------- Value of k: 11 ----------
Drama
Action|Adventure|Animation|Drama|Fantasy
Action|Drama|Thriller
Animation|Drama|Romance|Sci-Fi
Crime|Drama|Thriller
Crime|Drama
Drama
Drama|Romance
Drama
Drama|Thriller
---------- Value of k: 12 ----------
Crime|Thriller
Children|Comedy|Western
Drama
Comedy|Fantasy
Crime|Thriller
Crime|Drama|Thriller
Comedy
Action|Children|Comedy
Adventure|Comedy|Sci-Fi
Drama|Thriller

---------- Value of k: 13 ----------
Drama
Drama
Action|Sci-Fi|Thriller
Children|Comedy
Comedy|Crime|Mystery
Drama|War
Action|Comedy
Animation|Children|Fantasy
Drama
Documentary
---------- Value of k: 14 ----------
Drama|Thriller
Comedy
Adventure|Children
Drama|Horror|Thriller
Action|Adventure|Fantasy
Comedy|Drama
Documentary
Drama|Western
Drama|Romance|War
Comedy
---------- Value of k: 15 ----------
Action|Adventure|Animation|Fantasy|IMAX
Comedy|Drama
Animation|Comedy|Fantasy|Musical|Romance
Horror
Drama
Drama
Action|Drama|Horror|Thriller
Action|Comedy
Action|Crime|Drama
Crime|Horror|Mystery|Thriller
---------- Value of k: 16 ----------
Action|Drama|Thriller
Drama
Drama
Comedy|Drama|Mystery|Romance
Comedy|Crime
Comedy|Drama
Crime|Drama|Thriller
Action|Sci-Fi
Comedy|Romance
Comedy
---------- Value of k: 17 ----------
Comedy
Comedy|Horror

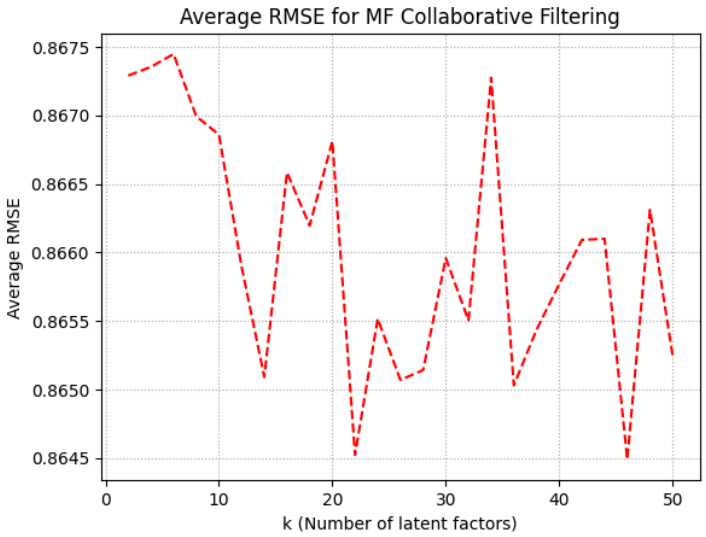
Drama|Thriller
Drama
Comedy|Romance
Action|Children|Comedy|Fantasy|Sci-Fi
Action|Adventure|Sci-Fi
Action|Sci-Fi
Adventure|Drama|War
Comedy|Drama
---------- Value of k: 18 ----------
Drama
Comedy|Mystery
Documentary
Comedy|Romance
Drama|Thriller
Comedy|Drama
Crime|Drama
Comedy|Musical
Drama|Fantasy|Horror|Thriller
Fantasy|Horror|Sci-Fi|Thriller
---------- Value of k: 19 ----------
Crime|Drama
Romance
Animation|Drama|Romance|Sci-Fi
Crime|Romance|Thriller
Documentary
Crime|Drama
Horror|Mystery|Thriller
Comedy
Drama|War
Action|Adventure|Children|Drama

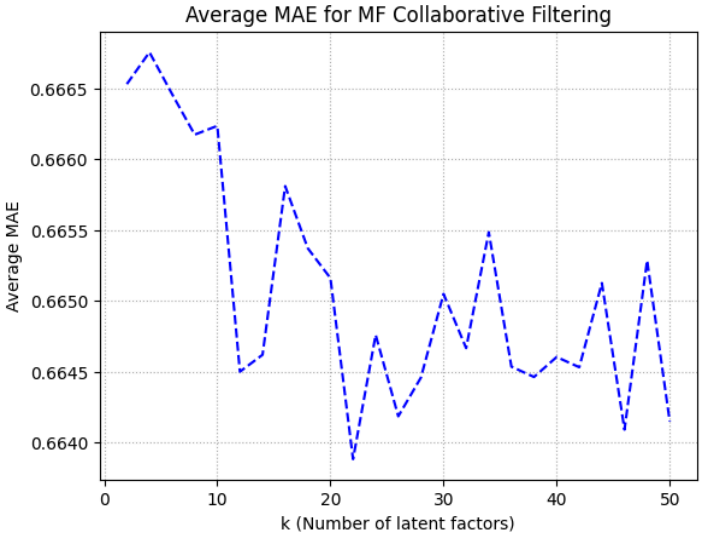
We can see that the latent factors are closely related to the movies. We can see that each group is strongly related to some genres whereas other groups are related to others. Thus we can conclude that the top 10 movies belong to a small set of genres for each group.

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.**





Though the graph is erratic in nature, we can observe that the values are consistent for a tiny range of k.

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?

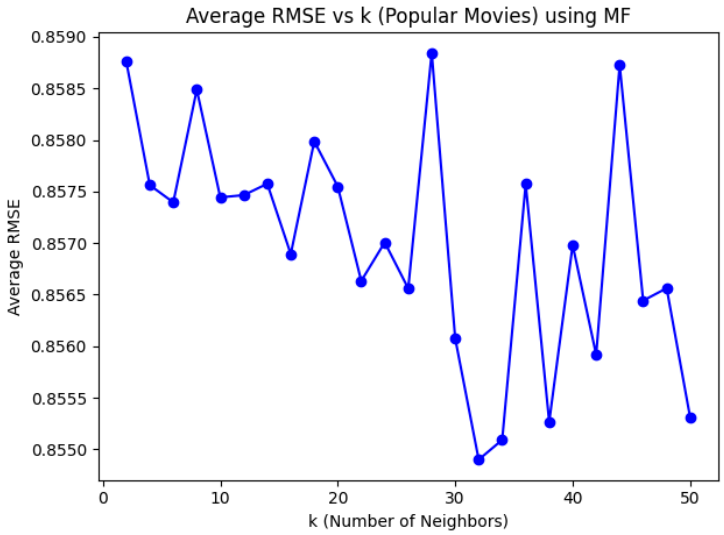
Min value of RMSE: 0.8638590129266989
K Value for MF corresponding to min value of RMSE: 28
Min value of MAE: 0.6629621308629217
K Value for MF corresponding to min value of MAE: 28
No, the optimal number of latents factors isn't the same. The value for MAE is pretty close to that of movie genres. This outperforms both kNN and NMF.

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.**

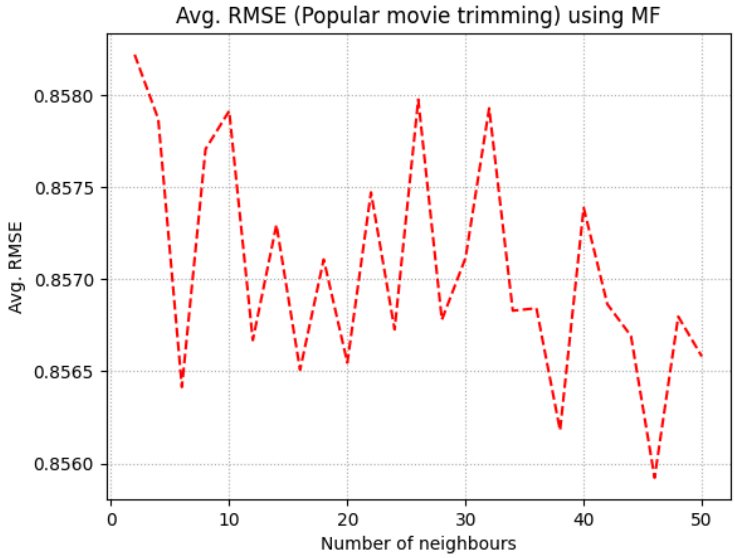
Popular Data:

Using cross validate:



Minimum Average RMSE for Popular Movies using MF: 0.8548996644046705 for k = 32

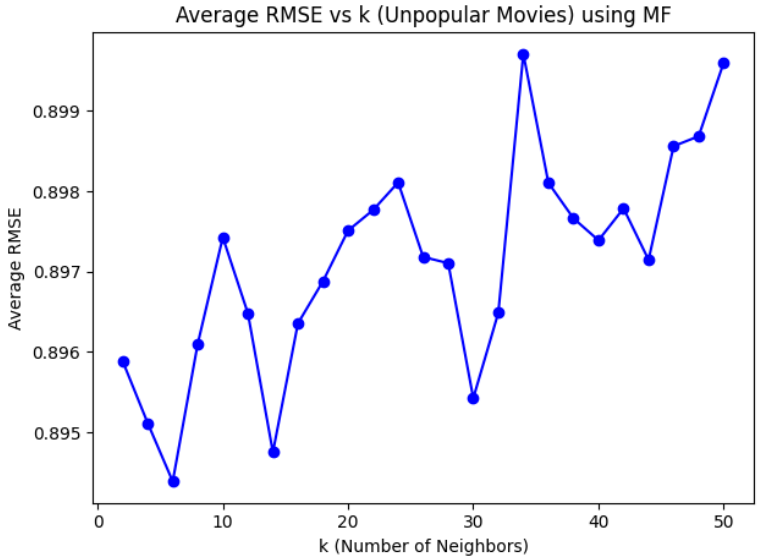
Using KFold:



Minimum Average RMSE for Popular Movies using MF: 0.8559222450175007 for k = 46

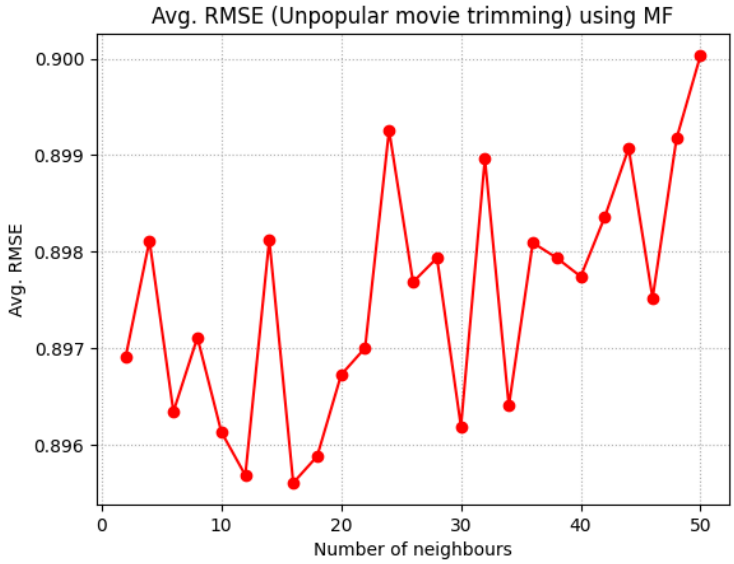
Unpopular Data:

Using cross validate:



Minimum Average RMSE for Unpopular Movies using MF: 0.8943898168957268 for k = 6

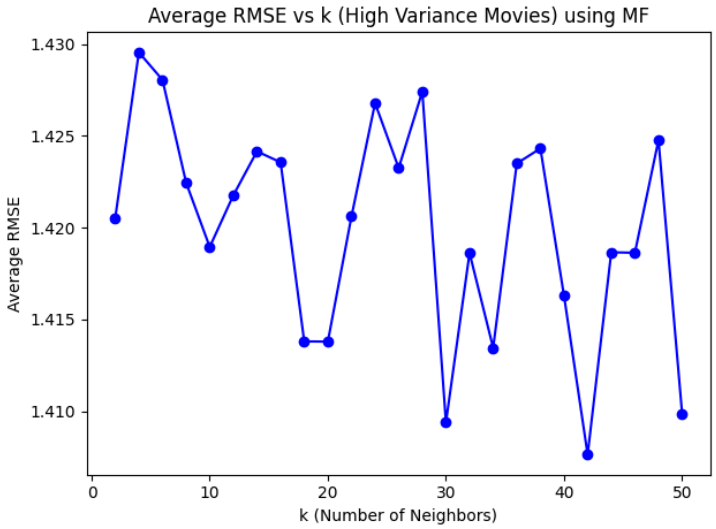
Using KFold:



Minimum Average RMSE for Unpopular Movies using MF: 0.8956050581857941 for k = 16

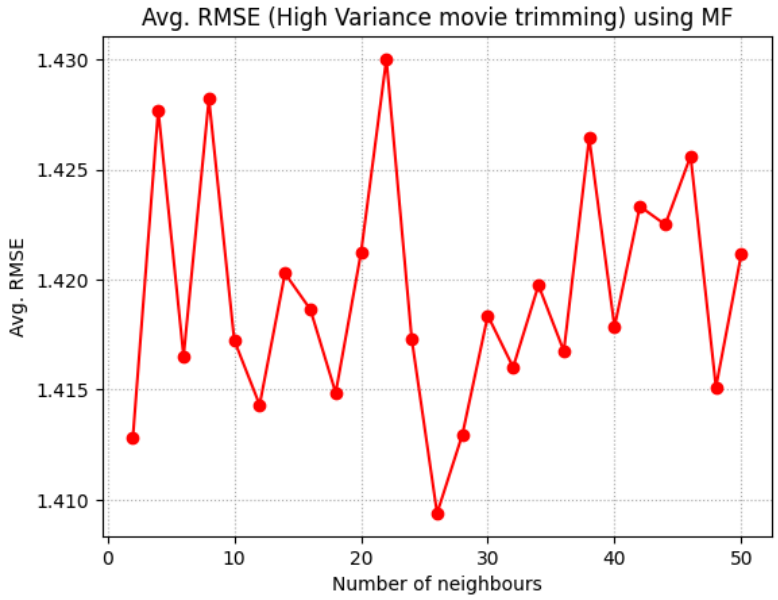
High Variance Data:

Using cross validate:



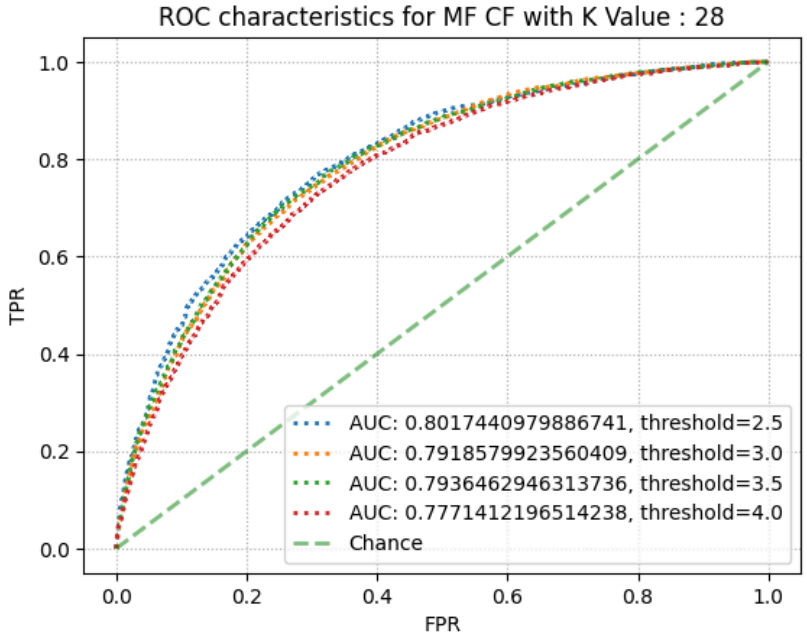
Minimum Average RMSE for High Variance Movies using MF: 1.4076443611129084 for k = 42

Using KFold:

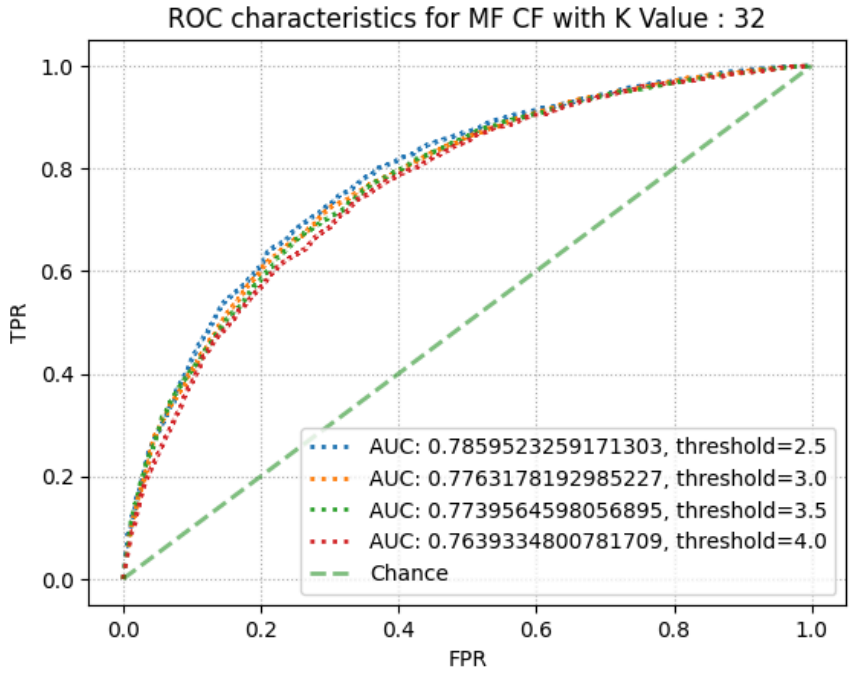


Minimum Average RMSE for High Variance Movies using MF: 1.4093797328685413 for k = 26

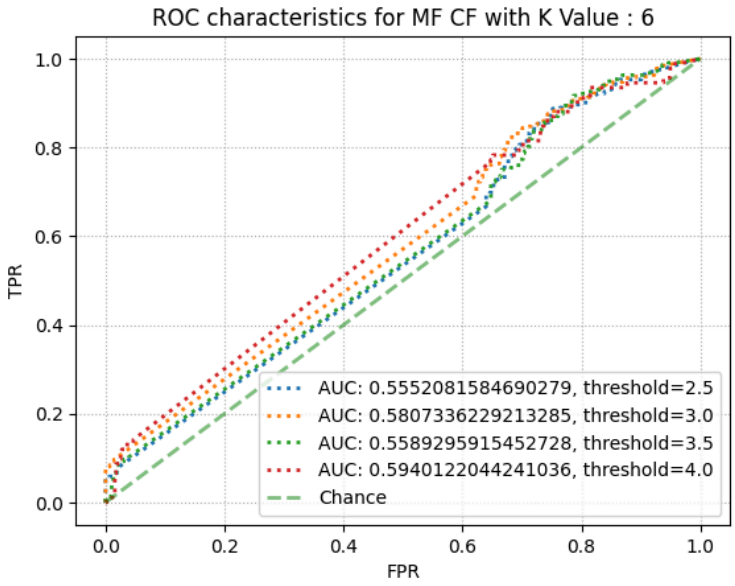
- **Plot the ROC curves for the MF-based collaborative filter and also report the area under the curve (AUC) value as done in Question 6.**
  Untrimmed Data:



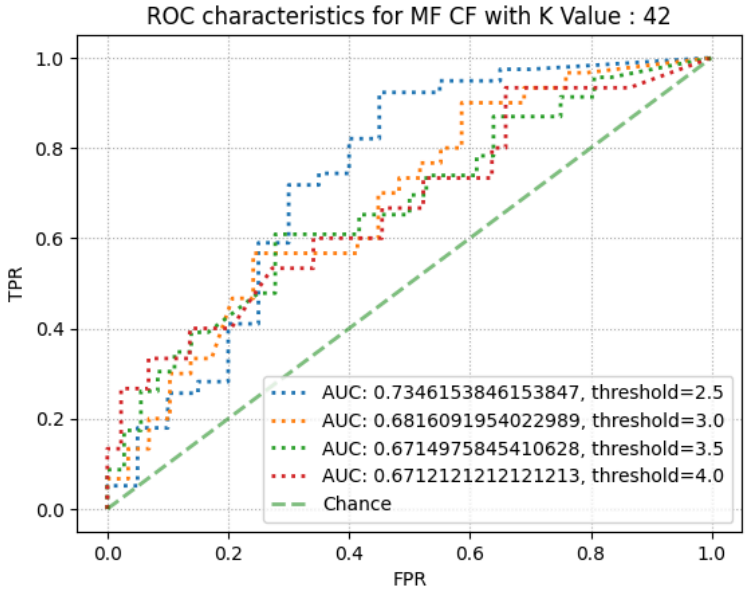
Popular Data:



Unpopular Data:



High Variance Data:



<u>QUESTION 11:</u>

Designing a Na¨ive Collaborative Filter:

- **Design a naive collaborative filter to predict the ratings of the movies in the original dataset and evaluate it's performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.**
  Average RMSE across the 10 folds for untrimmed data is 0.9410498478350723

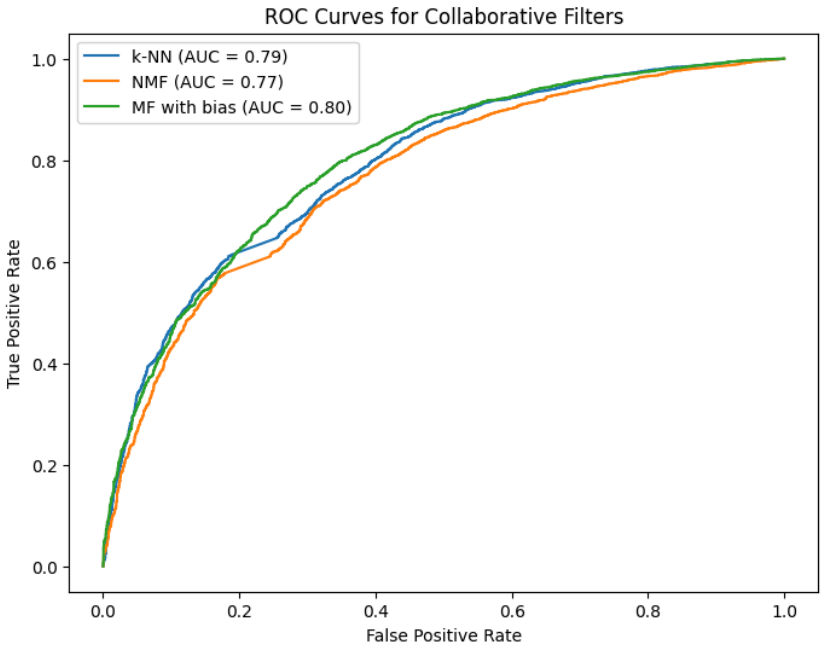
- **Performance on dataset subsets: For each of Popular, Unpopular and High-Variance test subsets -**

- **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 across the 10 folds for popular data is 0.9419809189073478
Average RMSE across the 10 folds for unpopular data is 1.0994365270630237
Average RMSE across the 10 folds for high variance data is 2.218492947925321

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.



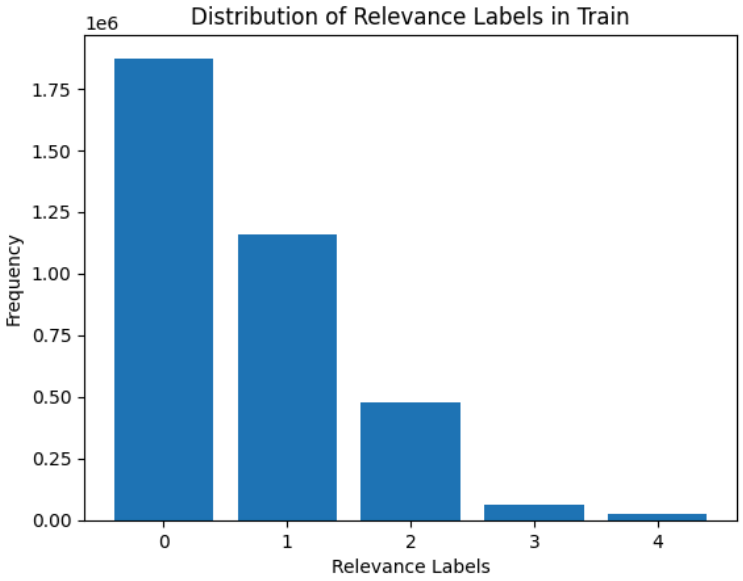
The above figure shows the ROC Curve for k-NN, NMF and MF based collaborative Filtering. We can see that MF performs the best and has the highest AUC followed by k-NN and then NMF. SVD is better able to represent the higher dimensional feature matrix as it does not have any constraints on U and V and thus provides a better factorization with much less information loss. Whereas NMF imposes certain conditions on U and V and hence we have less number of optimal choices. SVD produces embeddings of features with high relevance. This is done in a hierarchical manner. Therefore it is ordered in terms of relevance. Therefore for a high value of k, the embeddings do not hinder the model. Therefore they are robust to outliers and are not that affected by noise as compared to NMF. On the other hand k-NN predicts directly on the sparse rating matrix. This is why it has a poor prediction accuracy on higher dimensions. K-NN models are hence more difficult to scale. kNN is more sensitive to outliers.

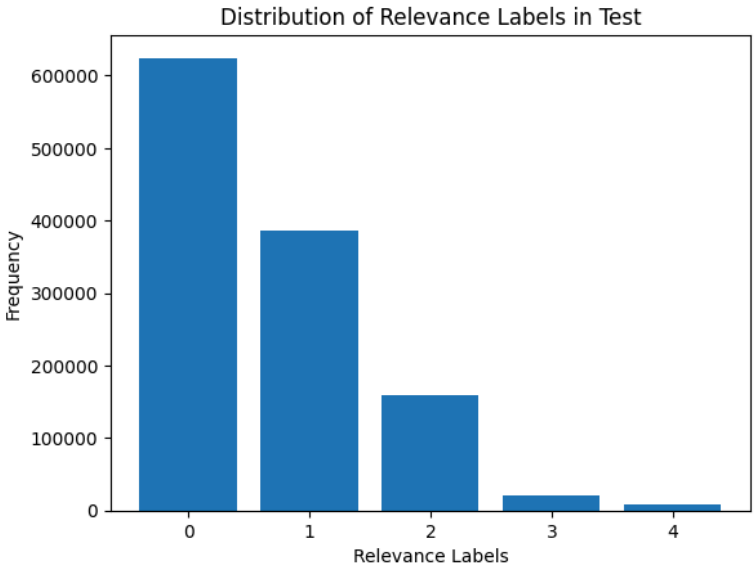
QUESTION 13:

Data Understanding and Preprocessing:

- **Use the provided helper code for loading and pre-processing Web10k data.**
- **Print out the number of unique queries in total and show distribution of relevance labels.**

The number of unique queries in training set is 10000
The number of unique queries in testing set is 10000





QUESTION 14:

LightGBM Model Training:

For each of the five provided folds, train a LightGBM model using the 'lambdarank' objective. After training, evaluate and report the model's performance on the test set using nDCG@3, nDCG@5 and nDCG@10.

Fold 1:
nDCG@3: 0.4564571300800643
nDCG@5: 0.4632890672260867
nDCG@10: 0.48286731451235976

Fold 2:
nDCG@3: 0.4538895365009714
nDCG@5: 0.4573292117374164
nDCG@10: 0.4767546810011047

Fold 3:
nDCG@3: 0.4490681494620125
nDCG@5: 0.4583480538865081
nDCG@10: 0.47589507831078093

Fold 4:
nDCG@3: 0.461178820507814
nDCG@5: 0.4663860127875315
nDCG@10: 0.487724614983737

Fold 5:
nDCG@3: 0.46963442883961365
nDCG@5: 0.4714315145908388
nDCG@10: 0.49035928048966515

QUESTION 15:

Result Analysis and Interpretation:

For each of the five provided folds, list top 5 most important features of the model based on the importance score. Please use model.booster .feature importance(importance type='gain') as demonstrated here for retrieving importance score per feature. You can also find helper code in the provided notebook.

Below are the top 5 features and their respective scores:

Fold 1 - Top 5 features :
[('Column_133', 23856.702950954437),
('Column_7', 4248.546391487122),
('Column_107', 4135.244449853897),
('Column_54', 4078.463216304779),
('Column_129', 3635.03702378273)]

Fold 2 - Top 5 features :
[('Column_133', 23578.90825009346),
('Column_7', 5157.964912414551),
('Column_54', 4386.669756650925),
('Column_107', 4094.0121722221375),
('Column_129', 4035.0706725120544)]

Fold 3 - Top 5 features :
[('Column_133', 23218.075441122055),
('Column_54', 4991.3033719062805),

('Column_107', 4226.807395458221),
('Column_129', 4059.7525141239166),
('Column_7', 3691.792320251465)]

Fold 4 - Top 5 features :
[('Column_133', 23796.899673223495),
('Column_7', 4622.622978448868),
('Column_54', 3883.4817056655884),
('Column_129', 3356.8469800949097),
('Column_128', 3207.5755367279053)]

Fold 5 - Top 5 features :
[('Column_133', 23540.94235444069),
('Column_7', 4794.9451723098755),
('Column_54', 4079.608554124832),
('Column_107', 3514.8357515335083),
('Column_129', 3209.0584440231323)]

QUESTION 16:

Experiments with Subset of Features:

For each of the five provided folds:

- **Remove the top 20 most important features according to the computed importance score in the question 15. Then train a new LightGBM model on the resulted 116 dimensional query- url data. Evaluate the performance of this new model on the test set using nDCG. Does the outcome align with your expectations? If not, please share your hypothesis regarding the potential reasons for this discrepancy.**

  Fold 1:
  nDCG@3: 0.37967488460229254
  nDCG@5: 0.3850299691938894
  nDCG@10: 0.4083636029390886

  Fold 2:
  nDCG@3: 0.3739449461043477
  nDCG@5: 0.3819536013454118
  nDCG@10: 0.4045026694861529

  Fold 3:
  nDCG@3: 0.3823833692306899
  nDCG@5: 0.3899961152757789
  nDCG@10: 0.4116363812695088

  Fold 4:
  nDCG@3: 0.381976845689231
  nDCG@5: 0.39281004672399866
  nDCG@10: 0.4121071637228934

  Fold 5:

nDCG@3: 0.38428336621785103
nDCG@5: 0.39216767580543455
nDCG@10: 0.4166871494621703

Yes the output aligns with the expected output. If we remove the top 20 features, we expect the performance to drop. This is why we observe that for each folder, the performance decreases drastically since we removed the important features.

- **Remove the 60 least important features according to the computed importance score in the question 15. Then train a new LightGBM model on the resulted 76 dimensional query-url data. Evaluate the performance of this new model on the test set using nDCG. Does the outcome align with your expectations? If not, please share your hypothesis regarding the potential reasons for this discrepancy.**

  Fold 1:
  nDCG@3: 0.4542530326427776
  nDCG@5: 0.46265744453383695
  nDCG@10: 0.4819713060930259

  Fold 2:
  nDCG@3: 0.457290225801309
  nDCG@5: 0.4602669061430629
  nDCG@10: 0.4772534003341443

  Fold 3:
  nDCG@3: 0.4497901754692966
  nDCG@5: 0.4586395637756899
  nDCG@10: 0.4774361560299901

  Fold 4:
  nDCG@3: 0.46063528567047524
  nDCG@5: 0.46734032124732483
  nDCG@10: 0.48888147783549574

  Fold 5:
  nDCG@3: 0.470186124814149
  nDCG@5: 0.4733522942533459
  nDCG@10: 0.4908165844880891

  Yes the output aligns with the expected output. If we remove the bottom 60 features, we expect the performance to not drop drastically. This is because these features do not contribute majorly to the performance. This is why we observe that for each folder, the performance does not decrease much since we removed the least important features.