ECE219 Large-Scale Data Mining: Models and Algorithms (Due: Sun, 02/25/24)

Project 3: Recommender Systems

Instructor: Vwani Roychowdhury

Aryaman Rajesh Gokarn 506303588

1 Question 1

1.a Computing the sparsity of the movie rating dataset

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

Answer:-

```
Sparsity: 0.016999683055613623
```

1.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.

Answer:-

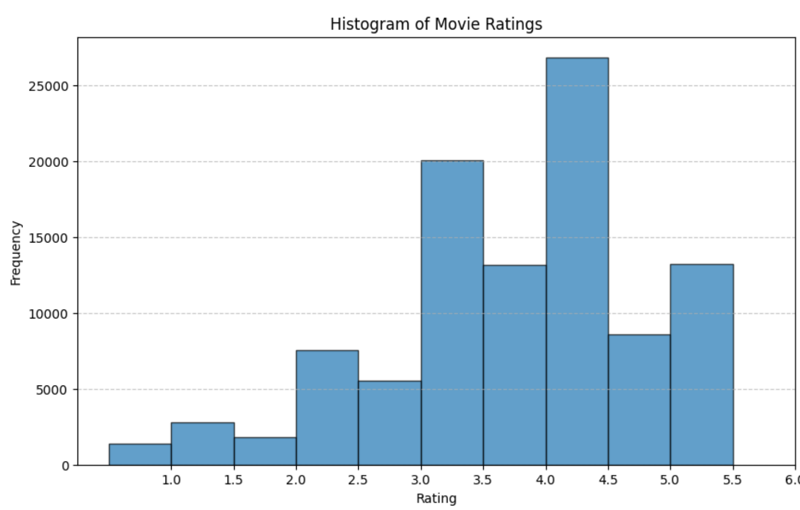


Figure 1: Histogram showing the frequency of the rating values

1.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 broken in any way. A monotonically decreasing trend is expected.

Answer:-

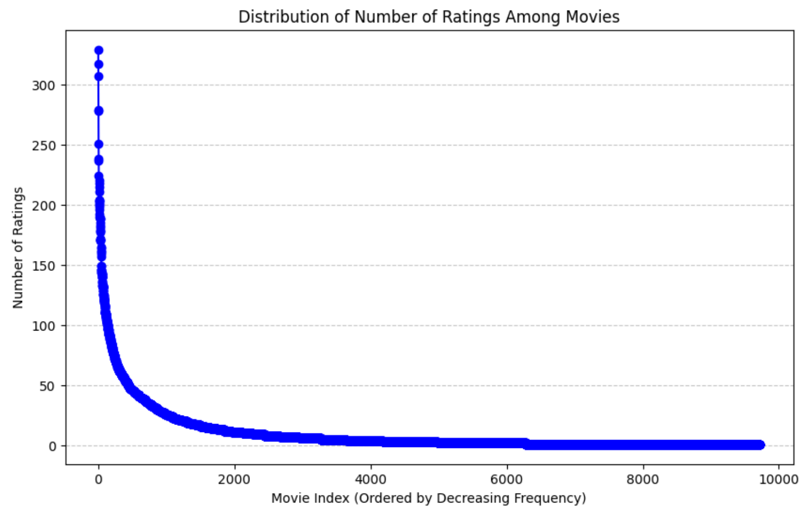


Figure 2: Distribution of the number of ratings received among movies

1.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.

Answer:-

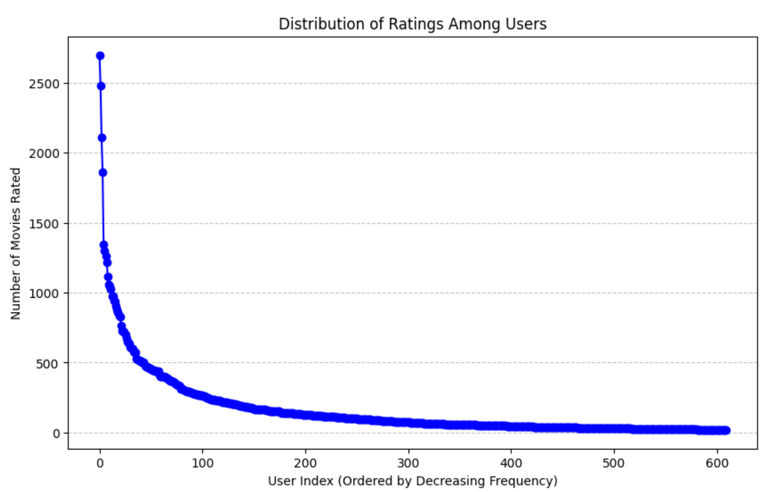


Figure 3: Distribution of ratings among users

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

Answer:-

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

1.f 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.

Answer:-

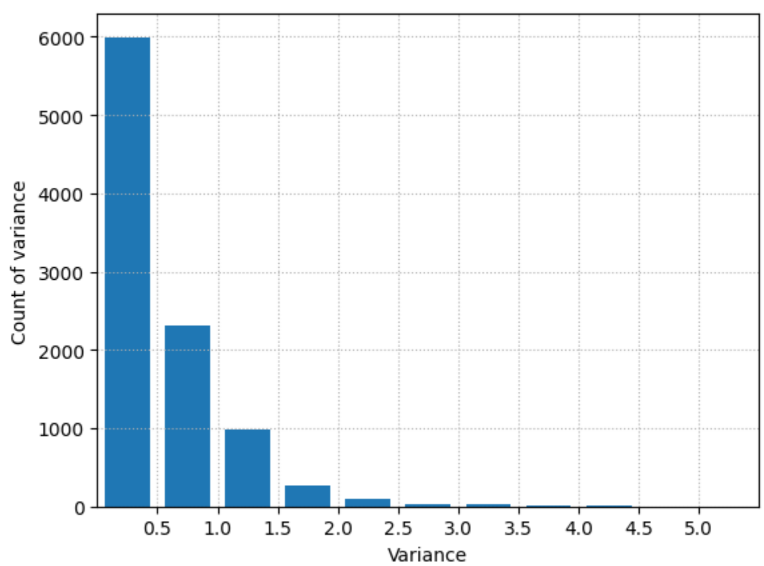


Figure 4: Variance of the rating values

2 Question 2

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

Answer:-

The mean rating *μu* for user *u* is calculated using her specified ratings (*ruk*) for the items in the set *Iu*. The formula for *μu* can be written as:

$$\mu_u = \frac{1}{|I_u|} \sum_{k \in I_u} r_{uk}$$

In words, it is the sum of all ratings (*ruk*) provided by user *u* for items in *Iu*, divided by the total number of items in *Iu* (denoted as |*Iu*|).

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

Answer:-

- In plain words, *Iu*∩*Iv* represents the set of items for which both user u and user u have provided ratings. In other words, it is the intersection of the sets of items that each user has rated.

- If *Iu*∩*Iv* = ∅ (empty set), it means that there are no common items for which both users u and u have provided ratings. In the context of the rating matrix R, this situation occurs when there are no movies that both users u and u have rated.

- This scenario is quite plausible in real-world recommender systems, especially when the rating matrix is sparse. The sparsity of the matrix indicates that users have provided ratings for only a small subset of the total items available. As a result, finding a common set of items that two users have rated can be challenging, leading to an empty intersection (*Iu*∩*Iv* = ∅)

3 Question 3

Understanding the Prediction function: Can you explain the reason behind mean-centering the raw ratings ($r_{vj} - \mu_v$) in the prediction function? (Hint: Consider users who either rate all items highly or rate all items poorly and the impact of these users on the prediction function.)

Answer:-

The reason behind mean-centering the raw ratings $((r_{vj} - \mu_v))$ in the prediction function is to account for user-specific biases in rating behavior. Mean-centering helps to mitigate the impact of users who consistently rate all items either highly or poorly.

Let's break down the reasoning:

1. **User-specific biases:** Users may have individual tendencies to rate items higher or lower on average compared to others. Some users might consistently give high ratings, while others might consistently give low ratings. Mean-centering helps in capturing these biases.
2. **Impact of users rating consistently high:** If a user consistently rates all items highly, their ratings are likely to be biased upward. Mean-centering adjusts for this bias by subtracting the user's mean rating ((\mu_v)), making the prediction relative to the user's own rating behavior.
3. **Impact of users rating consistently poorly:** Similarly, if a user consistently rates all items poorly, their ratings are likely biased downward. Mean-centering ensures that the predicted rating takes into account the user's personal bias by subtracting their mean rating.
4. **Normalization of predictions:** By mean-centering, the prediction function provides a more normalized view of how a user is likely to rate an item compared to their own average rating. This makes the predictions more comparable across users with different rating scales.

In summary, mean-centering is a crucial step to account for user-specific biases and ensure that the predicted ratings are more representative of a user's relative preferences rather than their absolute rating tendencies. This normalization enhances the accuracy and fairness of the collaborative filtering predictions.

4 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).

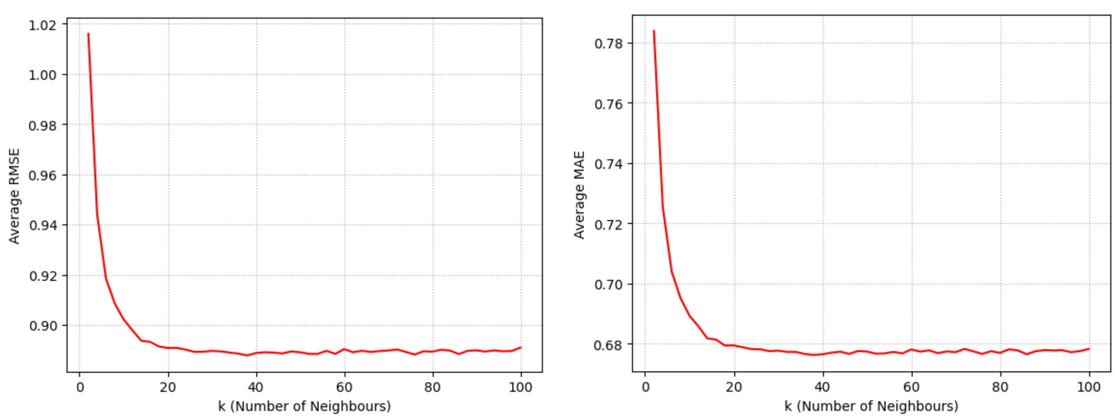


Figure 5: Average RMSE and MAE Plots

The charts depicted above illustrates the kNN Collaborative filter. It is evident that the values exhibit a continuous decrease until a certain point, beyond which there is no notable further reduction in the average values. This pattern is consistent across both graphs.

5 Question 5

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

Answer:-

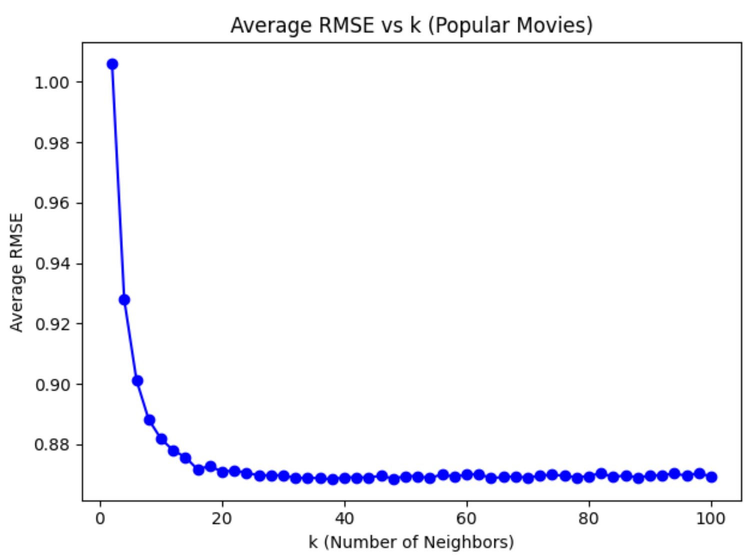
Min value of RMSE: 0.8882835963107757
Min index of RMSE: 48
Min value of MAE: 0.67657722394923
Min index of MAE: 22

6 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.

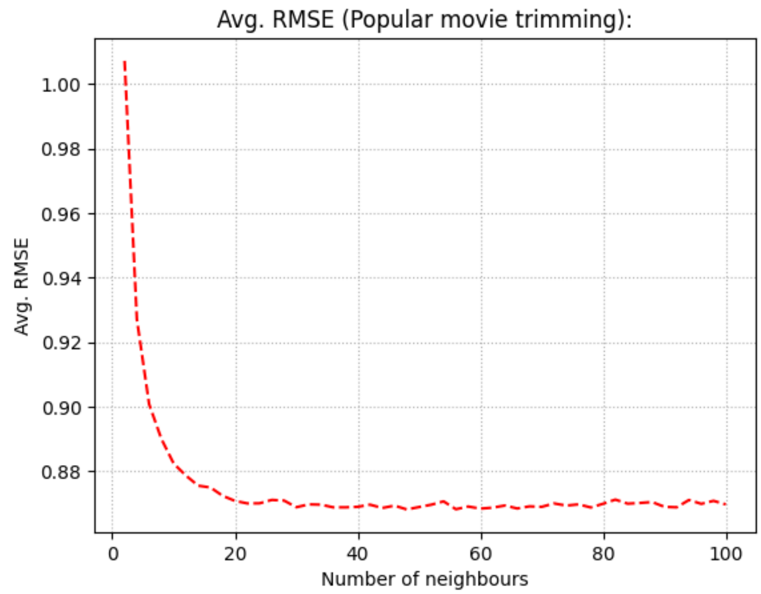
Answer:-

Popular Data
Using cross validate:



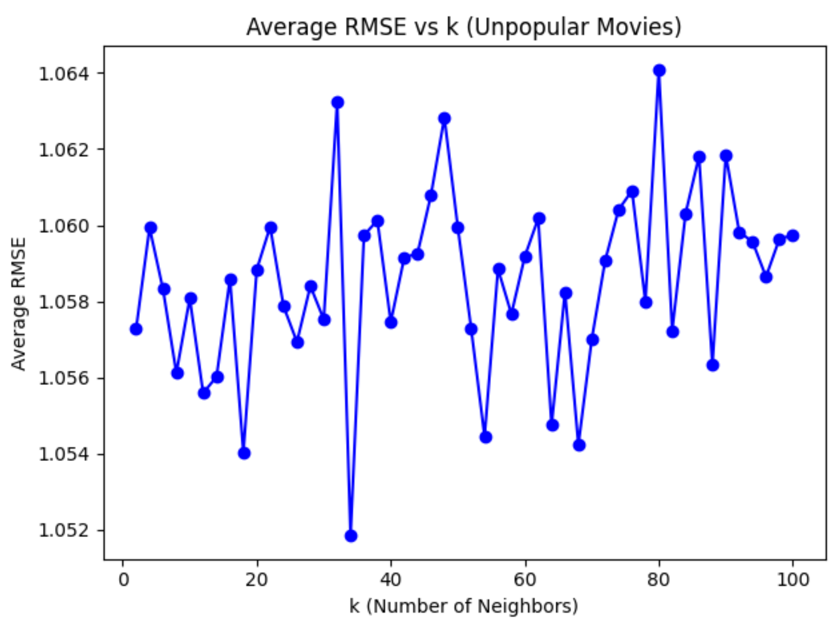
Minimum Average RMSE for Popular Movies: 0.8680020955226008 for k = 52

Using KFold:



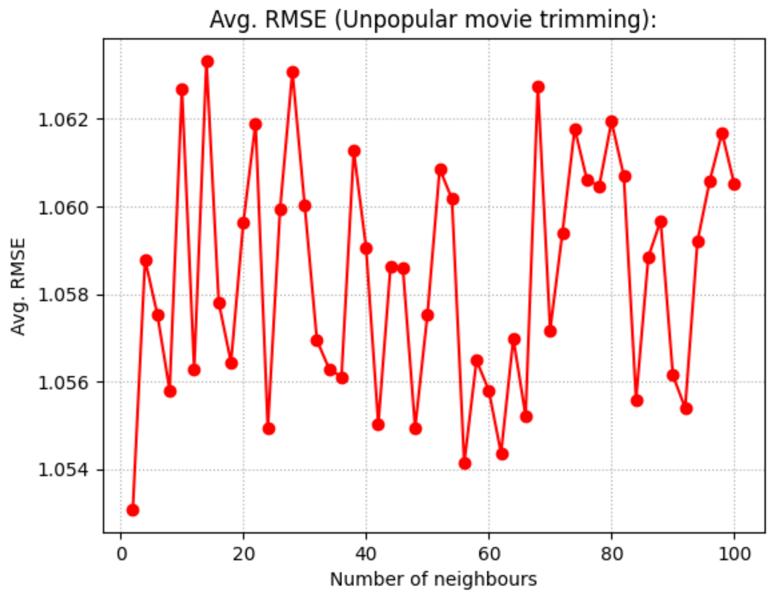
Minimum Average RMSE for Popular Movies: 0.86857405163718 73 for k =19

Unpopular Data:
Using cross validate:



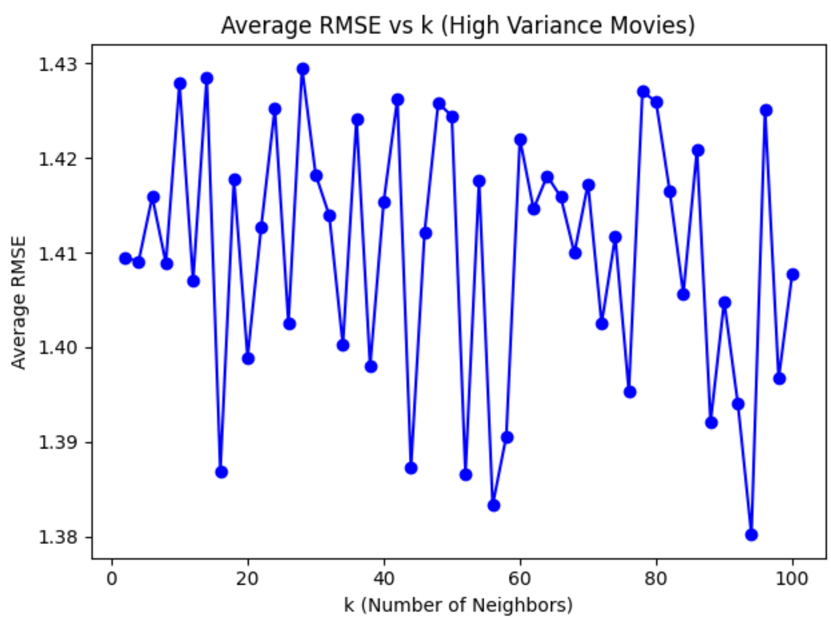
Minimum Average RMSE for Unpopular Movies: 1.051843178296814 for k = 34

Using KFold:



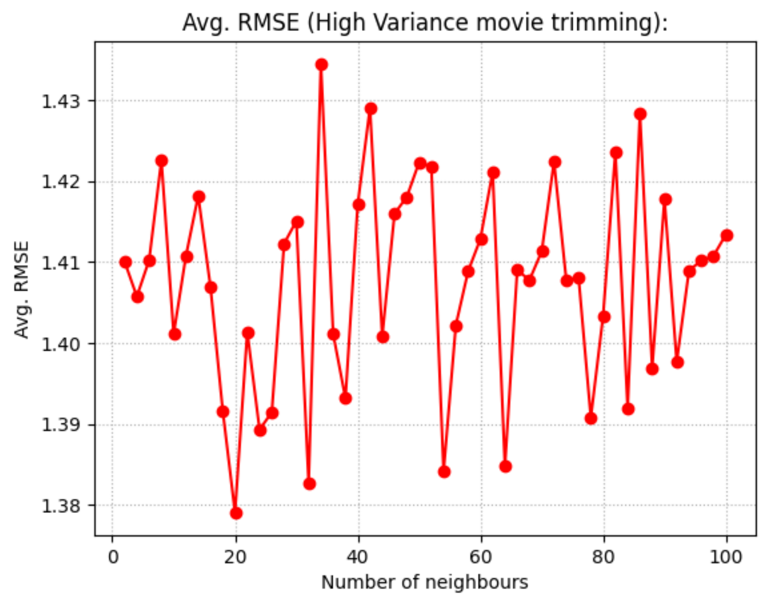
Minimum Average RMSE for Unpopular Movies: 1.0530755893575845 for k = 2

High Variance Data:
Using cross validate:



Minimum Average RMSE for High Variance Movies: 1.380187653759863 for k = 94

Using KFold:

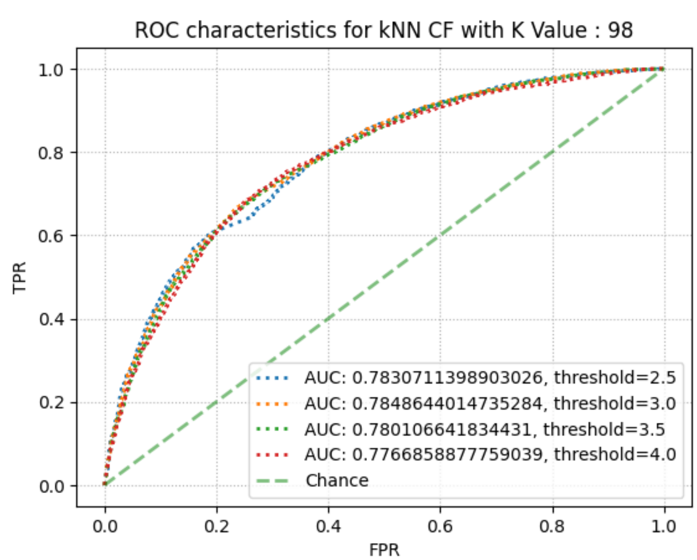


Minimum Average RMSE for High Variance Movies: 1.3790916542148088 for k = 20

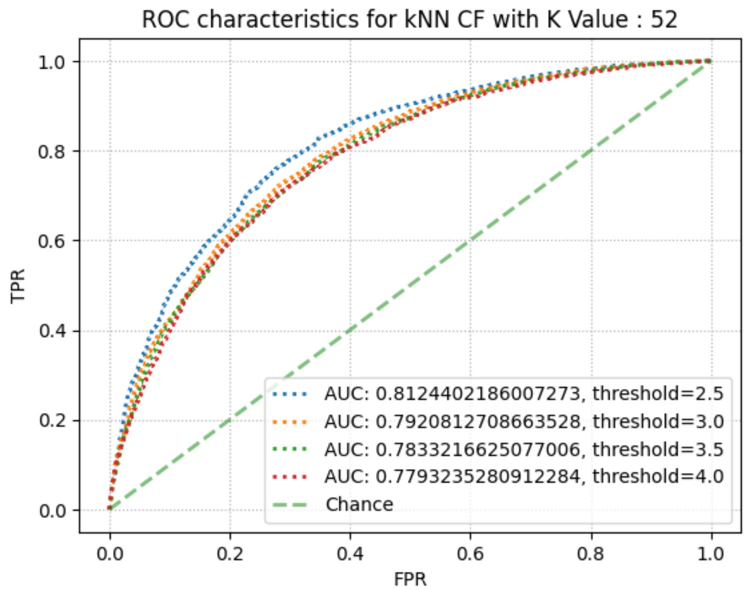
- Plot the ROC curves for the k-NN collaborative filters for threshold values [2.5, 3, 3.5, 4]. These thresholds are applied only on the ground truth labels in held-out validation set. For each of the plots, also report the area under the curve (AUC) value. You should have 4 × 4 plots in this section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

Answer:-

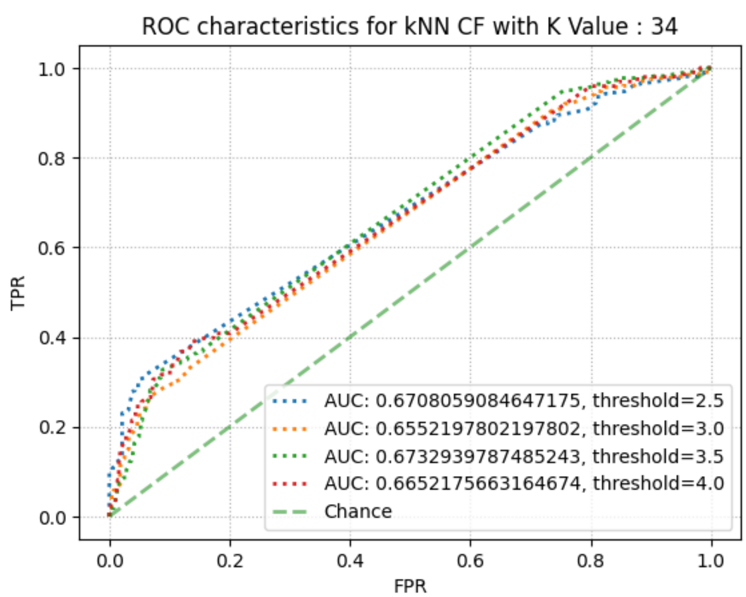
Untrimmed Data:



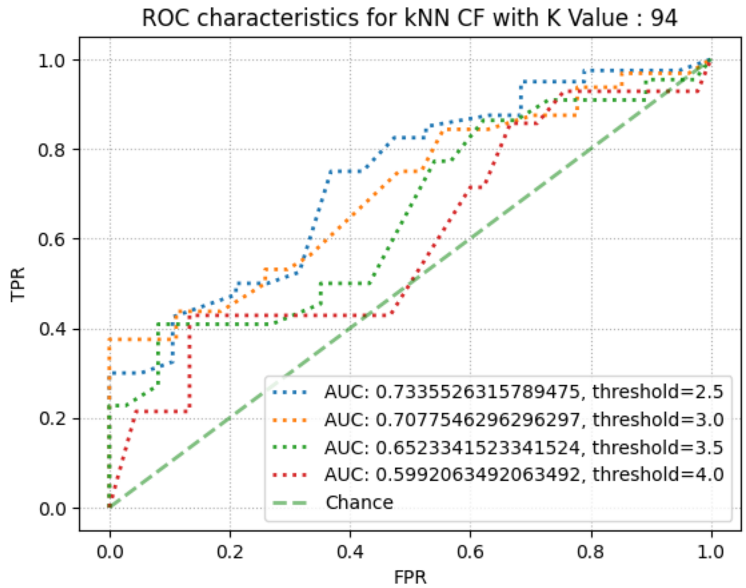
Popular Data:



Unpopular Data:



High Variance Data:



7 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.

Answer:-

No, the optimization problem lacks convexity. This is because both matrices U and V are simultaneously unknown variables. If we hold one variable constant and attempt to solve for the other, the optimization problem can be reformulated into a least squares problem, which is convex. However, the joint convexity for both U and V is not maintained, as there are multiple local minima present in the gradient plane of the objective function. Specifically, if we fix the matrix U and solve for the matrix V, we will have the following least-squares problem:

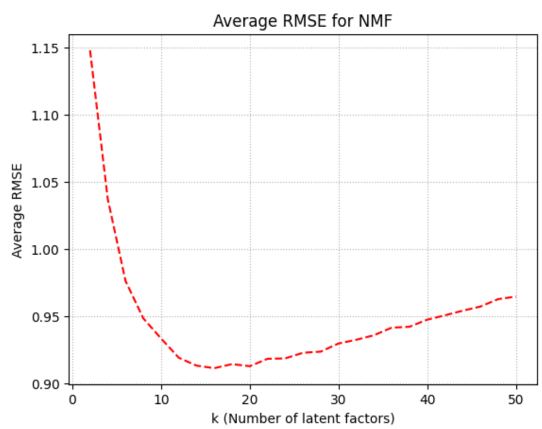
$$\underset{V}{\text{minimize}} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(r_{ij} - (\bar{U}V)_{ij}^{T})^2$$

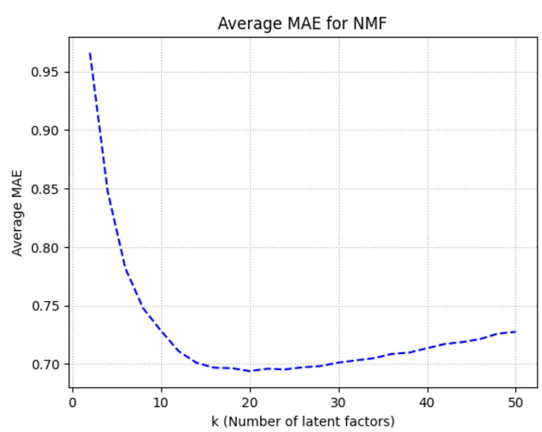
8 Question 8

Designing the NMF Collaborative Filter:

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.

Answer:-





8.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?

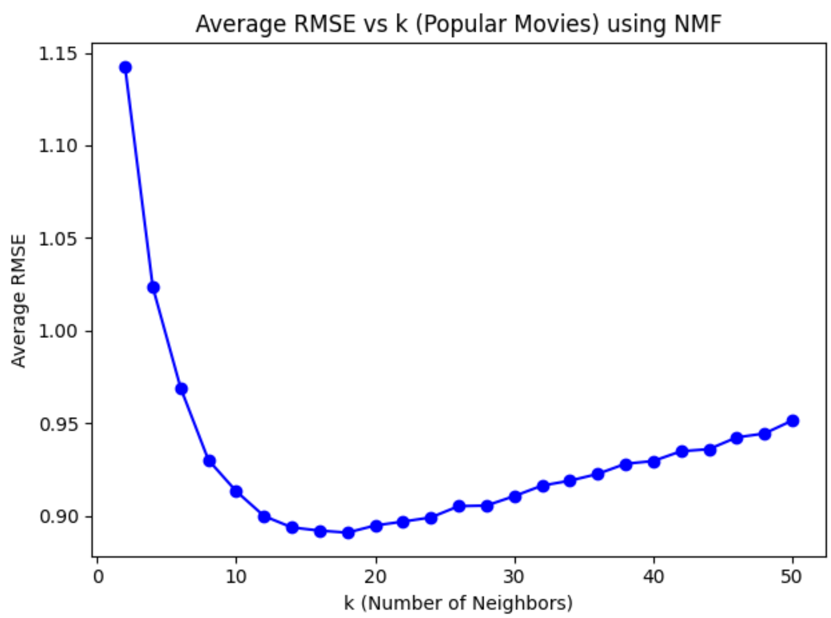
Answer:-

Min value of RMSE: 0.9101903561261692
K Value for NMF corresponding to min value of RMSE: 16
Min value of MAE: 0.6929617378908951
K Value for NMF corresponding to min value of MAE: 22

8.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. 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.

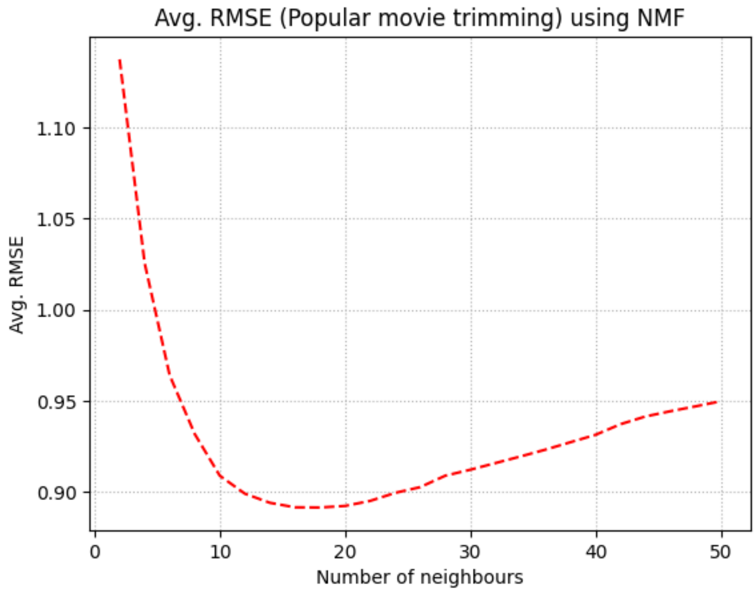
Answer:-

Popular Data
Using cross validate:



Minimum Average RMSE for Popular Movies: 0.8908124737006334 for k = 18

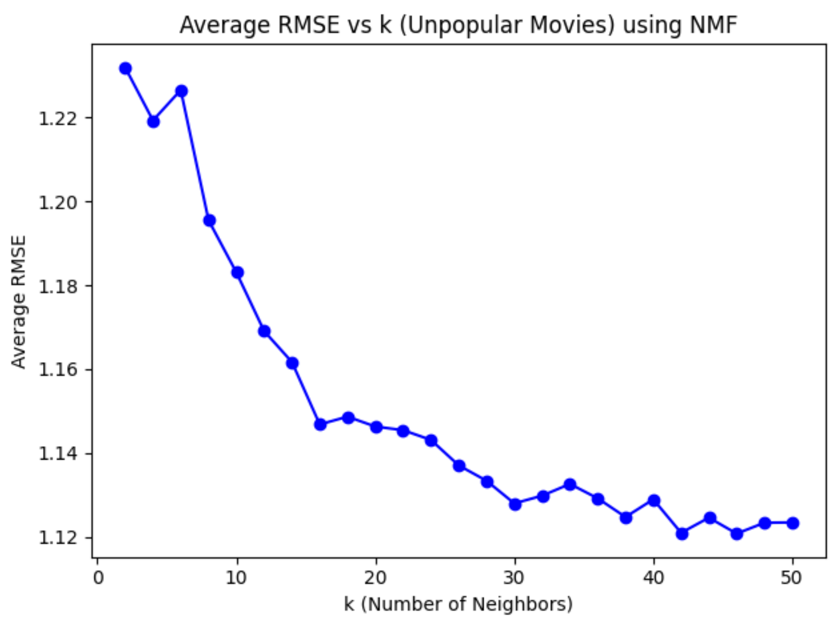
Using KFold:



Minimum Average RMSE for Popular Movies: 0.8915614780941125 for k = 18

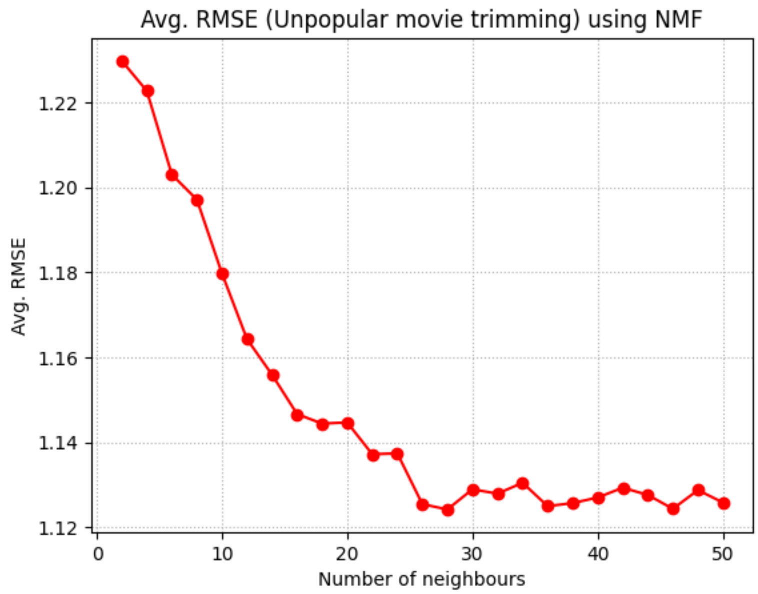
Unpopular Data:

Using cross validate:



Minimum Average RMSE for Unpopular Movies: 1.120828422488641 for k = 46

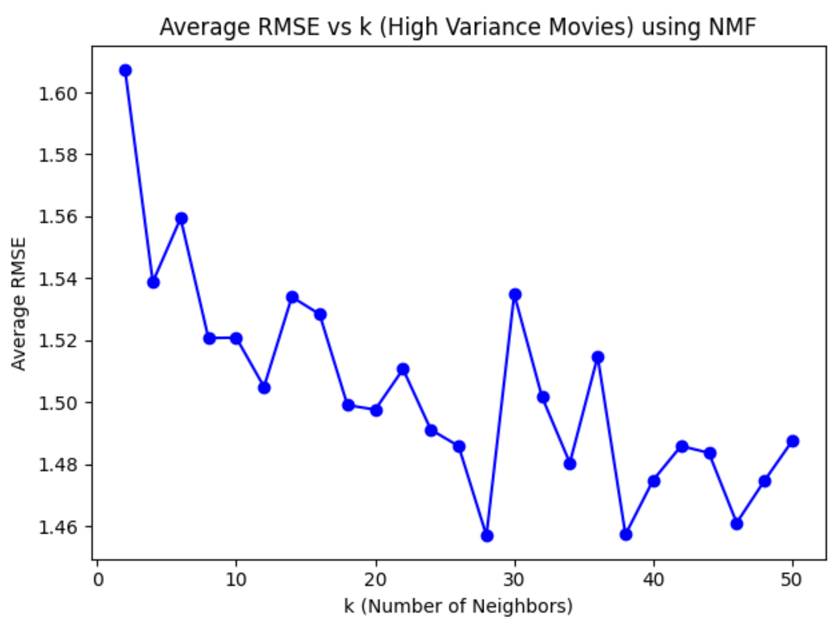
Using KFold:



Minimum Average RMSE for Unpopular Movies: 1.1241982081464803 for k = 28

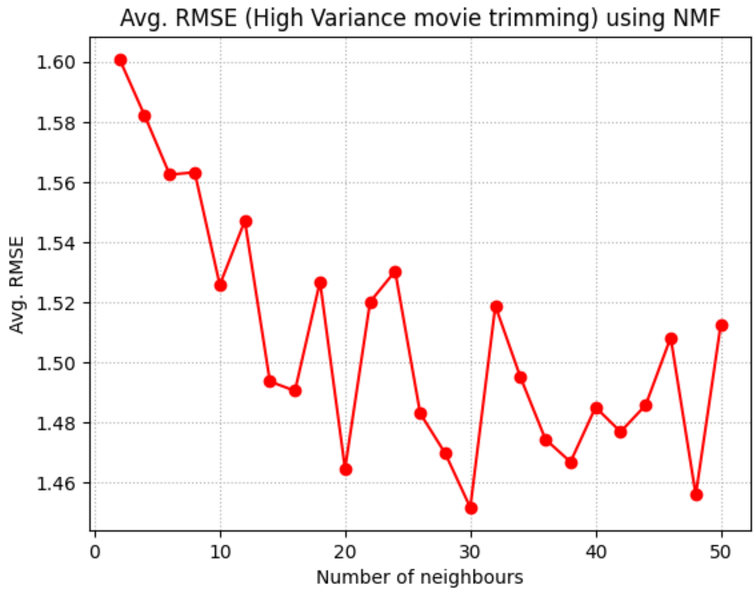
High Variance Data:

Using cross validate:



Minimum Average RMSE for High Variance Movies: 1.456935643091771 for k = 28

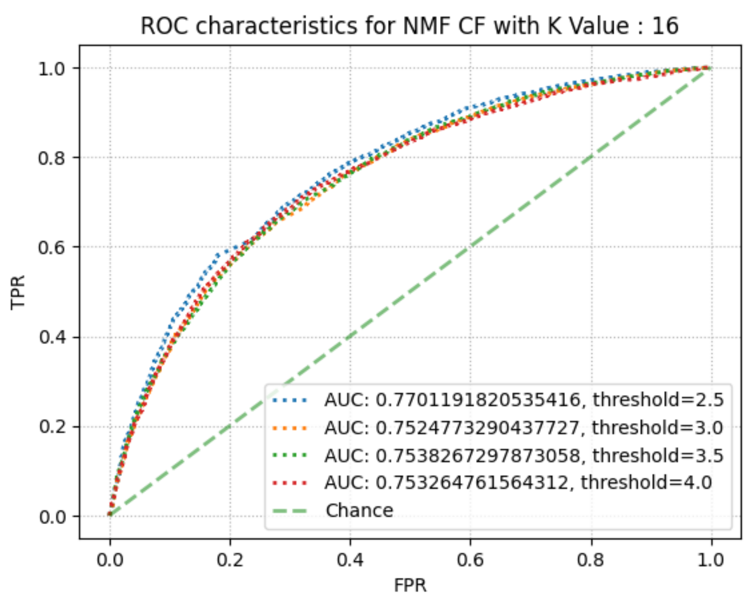
Using KFold:



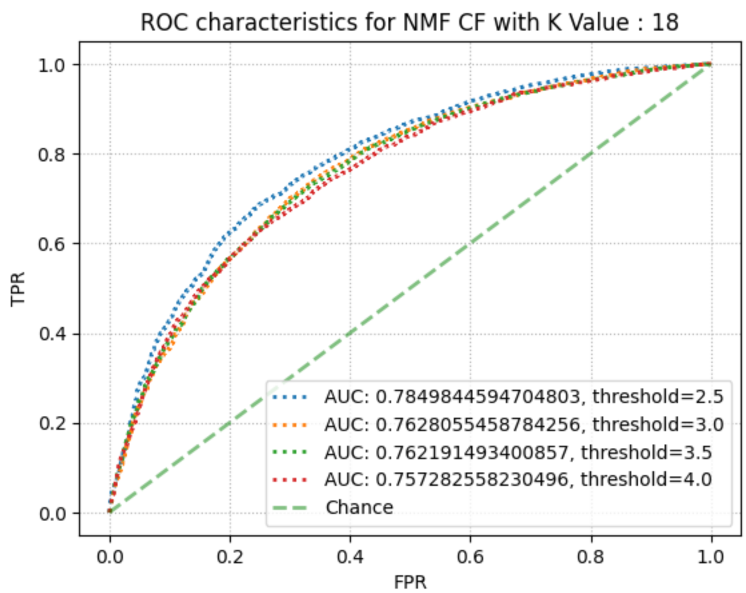
Minimum Average RMSE for High Variance Movies: 1.4516837185866414 for k = 30

- 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.

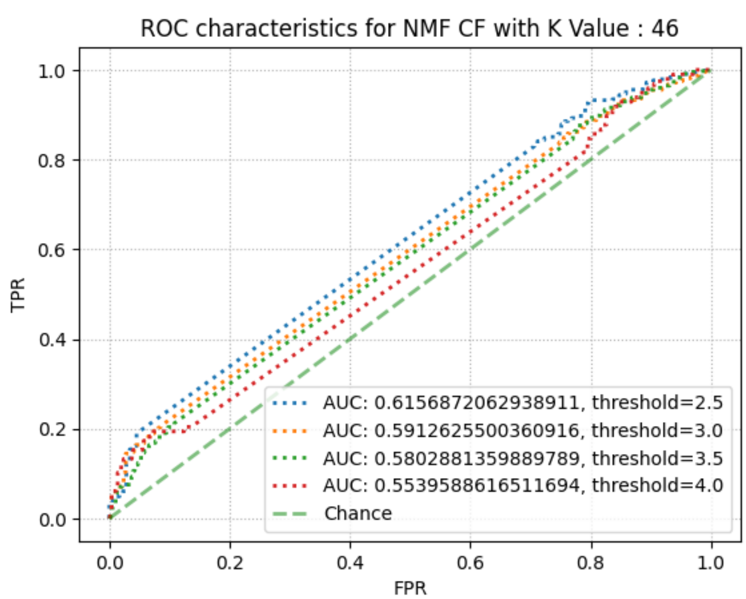
Untrimmed Data:



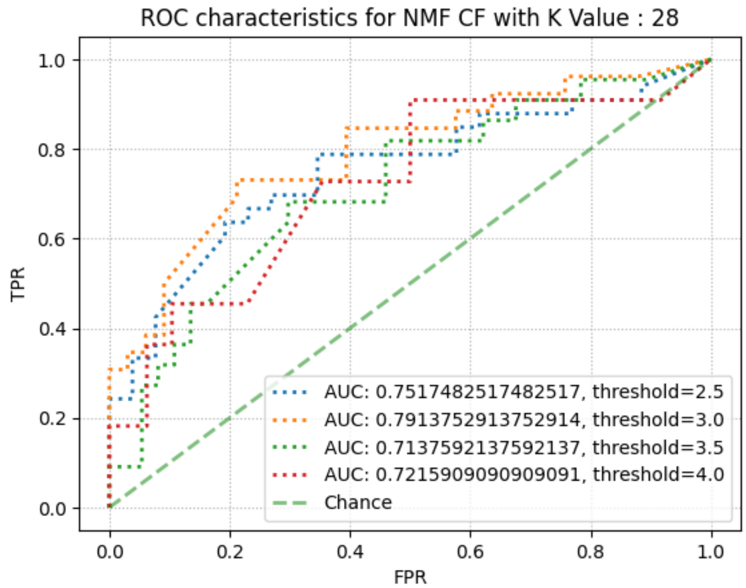
Popular Data:



Unpopular Data:



High Variance Data:



9 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?

Answer:-

```
---------------------- Value of k:  0  ----------------------
Comedy|Romance
Action|Animation|Drama|Sci-Fi
Comedy
Horror
Action
Comedy
Horror
Drama|Romance
Documentary
Drama|Thriller
```

```
----------------------- Value of k:  1  -----------------------
Comedy
Western
Action|War
Drama|Musical|Romance
Comedy
Adventure|Animation|Children|Comedy
Horror
Comedy|Drama
Action|Adventure|Animation|Children|Comedy
Comedy|Drama|Romance
----------------------- Value of k:  2  -----------------------
Drama
Comedy|Romance
Action|Crime|Drama|Thriller
Drama|Romance
Comedy
Adventure|Comedy|Sci-Fi
Horror
Drama|Horror
Comedy
Comedy|Musical
----------------------- Value of k:  3  -----------------------
Comedy|Romance
Action|Drama|Horror|Thriller
Action|Comedy
Action|Crime|Drama|Thriller
Comedy
Action|Comedy|Crime|Drama|Sci-Fi
Comedy
Adventure|Comedy|Fantasy|Sci-Fi
Drama
Action|Drama
----------------------- Value of k:  4  -----------------------
Comedy|Romance
Comedy
Crime|Drama|Thriller
Crime|Mystery|Thriller
Animation|Children|Comedy|Drama
Action|Crime|Drama|Thriller
Comedy|Drama|Romance
Children|Musical
Drama|Romance
Comedy
----------------------- Value of k:  5  -----------------------
Drama|Romance
Comedy|Drama
Comedy
Sci-Fi
Adventure|Children
Action|Sci-Fi|Thriller
Children|Musical
Comedy
Horror|Mystery|Thriller
Adventure|Comedy|Sci-Fi
```

```
----------------------- Value of k:  6  -----------------------
Action|Sci-Fi|Thriller
Animation|Comedy
Action|Crime|Horror|Thriller
Crime|Mystery|Thriller
Comedy|Horror|Sci-Fi
Comedy
Comedy|Drama
Comedy
Adventure|Animation|Children|Comedy
Drama
----------------------- Value of k:  7  -----------------------
Action|Animation|Drama|Sci-Fi
Comedy
Animation|Children|Comedy|Musical
Romance
Sci-Fi|Thriller
Comedy|Drama
Action|Sci-Fi|War
Sci-Fi
Action|Comedy
Drama
----------------------- Value of k:  8  -----------------------
Thriller
Drama|Film-Noir|Mystery|Romance
Drama|War
Drama|Mystery|Romance|War
Children|Comedy
Comedy
Drama
Drama|Musical|Romance
Drama
Comedy|Drama|Romance
----------------------- Value of k:  9  -----------------------
Comedy
Animation|Children|Comedy
Drama|Musical
Drama
Drama
Drama|Romance
Drama|Romance
Action
Action|Adventure|Sci-Fi
Drama
----------------------- Value of k:  10  -----------------------
Comedy|Fantasy|Romance
Drama
Comedy|Romance
Drama|Mystery|Thriller
Comedy|Drama|Fantasy|Romance
Children|Comedy|Fantasy|Musical
Crime
Drama
Comedy|War
Romance
```

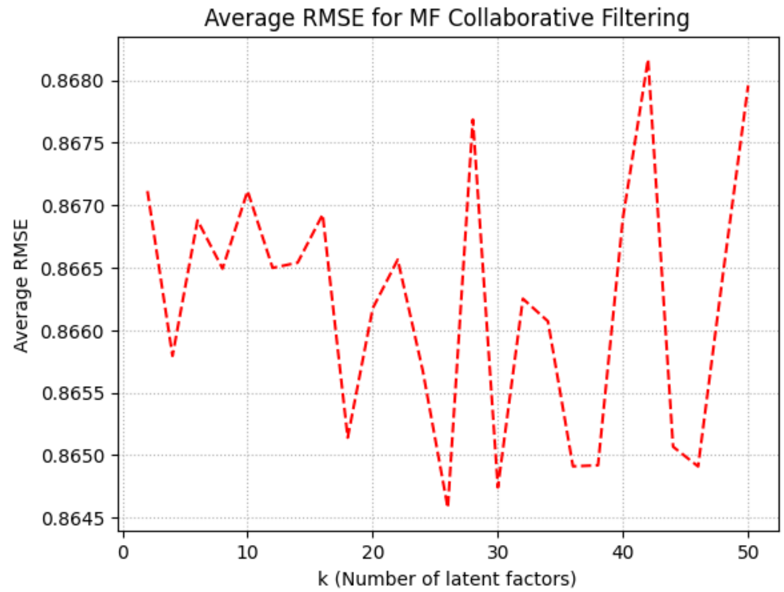
```
----------------------- Value of k:  11  -----------------------
Comedy|Crime
Adventure|Comedy
Sci-Fi
Action
Drama|Film-Noir|Mystery|Romance
Action|Crime|Thriller
Comedy|Drama
Comedy|Horror
Drama|Romance
Comedy|Musical|Romance|Western
----------------------- Value of k:  12  -----------------------
Comedy
Action|Sci-Fi|Thriller
Comedy
Comedy|Drama|Romance
Horror|Thriller
Action|Comedy
Drama|Fantasy|Musical|Romance
Comedy|Drama|Fantasy|Romance
Drama|Romance
Comedy|Drama
----------------------- Value of k:  13  -----------------------
Animation
Action|Drama|War
Action|Drama|Thriller|War
Comedy
Action|Comedy|Crime|Drama
Crime|Drama|Romance
Action|Adventure|Sci-Fi
Horror|Sci-Fi|Thriller
Adventure|Drama|Mystery|Thriller
Thriller
----------------------- Value of k:  14  -----------------------
Comedy
Drama|Musical
Animation|Children|Comedy
Comedy|Drama
Drama
Drama
Comedy|Fantasy
Horror
Crime|Drama|Film-Noir|Romance|Thriller
Comedy|Crime|Drama|Mystery|Thriller
----------------------- Value of k:  15  -----------------------
Animation|Comedy
Comedy|Sci-Fi
Drama|Romance
Drama|War
Action
Adventure|Drama
Fantasy|Horror
Comedy
Drama
Action|Comedy
```

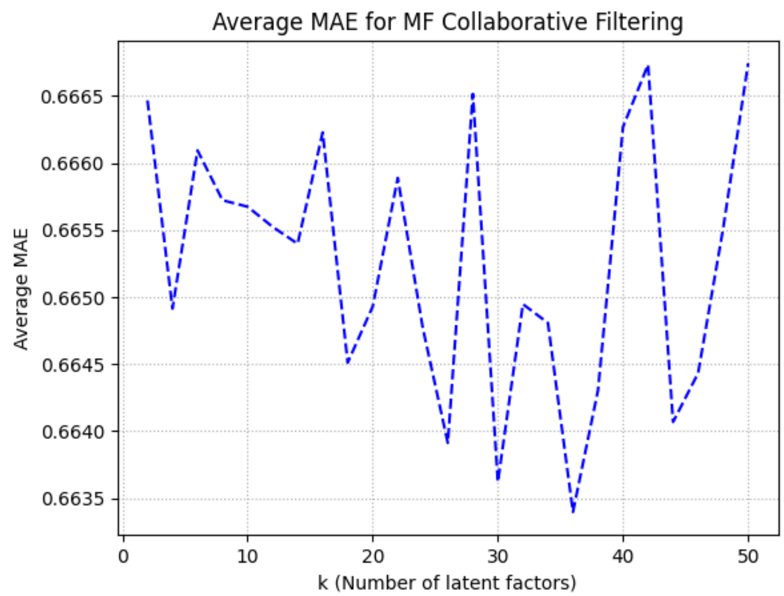
```
----------------------- Value of k:  16  -----------------------
Comedy|Drama|Romance
Crime|Horror|Mystery|Thriller
Drama|Sci-Fi
Crime|Drama|Romance
Comedy|Drama|Romance
Action|Sci-Fi|Thriller
Horror|Sci-Fi|Thriller
Drama
Drama|Mystery|Thriller
Adventure|Children|Fantasy
----------------------- Value of k:  17  -----------------------
Drama
Horror
Drama|Musical
Comedy
Comedy|Horror
Drama
Comedy|Drama|Sci-Fi|War
Documentary
Drama
Action|Crime|Drama|Thriller
----------------------- Value of k:  18  -----------------------
Western
Crime|Drama|Mystery|Thriller
Thriller
Adventure|Comedy
Drama
Drama|Film-Noir|Mystery
Drama|War
Thriller
Drama
Action|Comedy
----------------------- Value of k:  19  -----------------------
Comedy
Comedy
Comedy|Romance
Drama|Romance
Comedy|Drama
Action|Drama|Mystery|Romance|Thriller
Crime|Drama|Thriller
Drama
Crime|Drama|Mystery|Thriller
Adventure|Drama
```

10 Question 10

10.a Within 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.

Answer:-





10.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?

Answer:-

```
Min value of RMSE:  0.8645774992885356
K Value for MF corresponding to min value of RMSE:  26
```

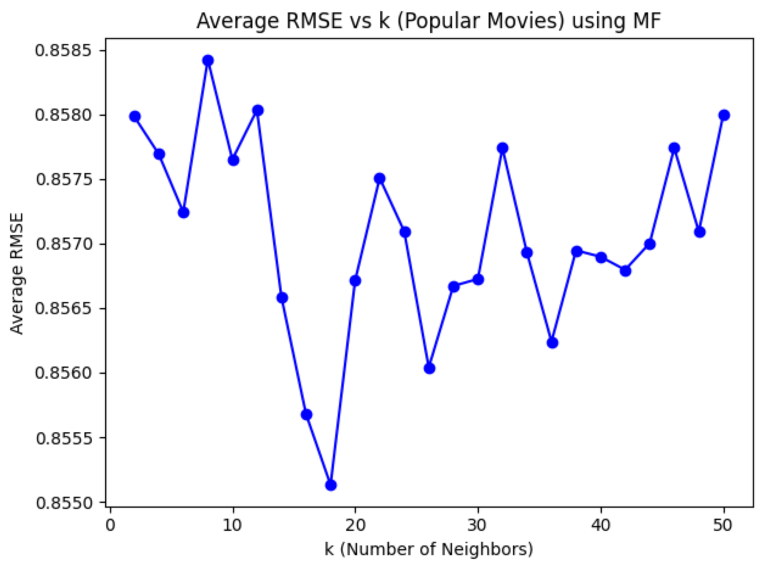
```
Min value of MAE:  0.6633983049607411
K Value for MF corresponding to min value of MAE:  36
```

10.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.

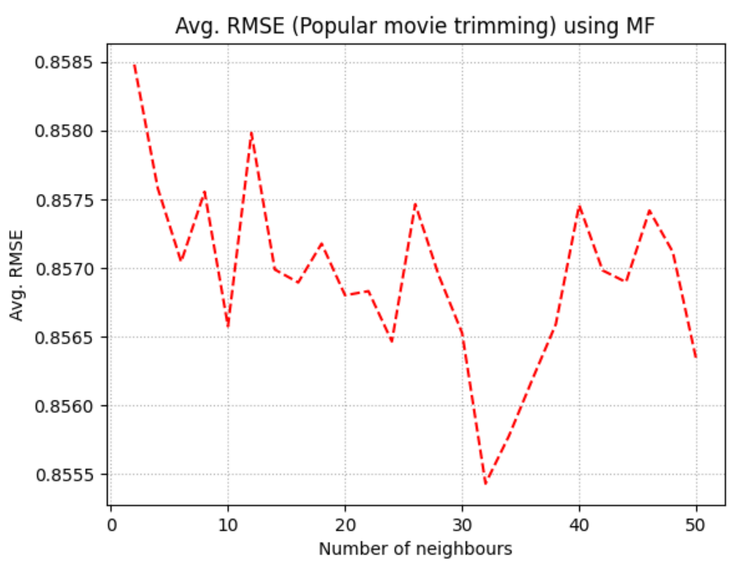
Answer:-

Popular Data:
Using cross validate:



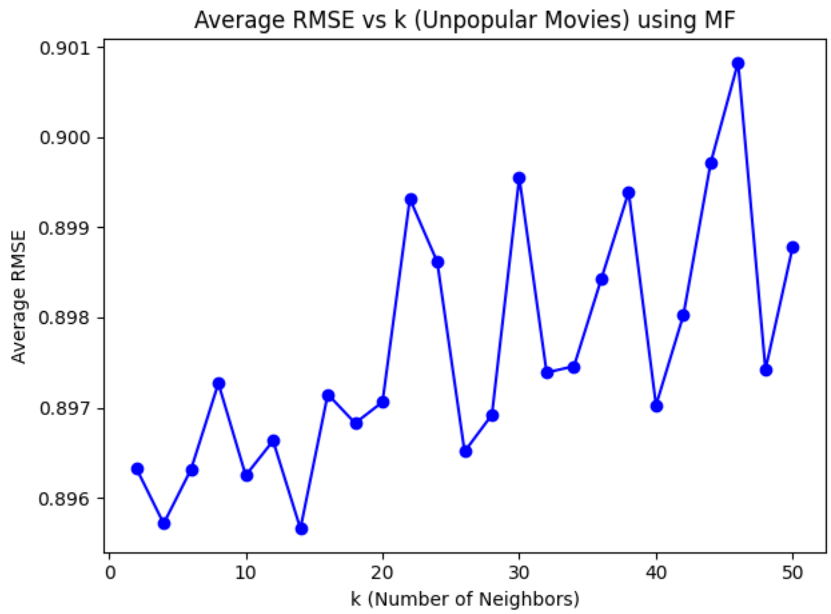
Minimum Average RMSE for Popular Movies using MF: 0.8551311139788768 for k = 18

Using KFold:



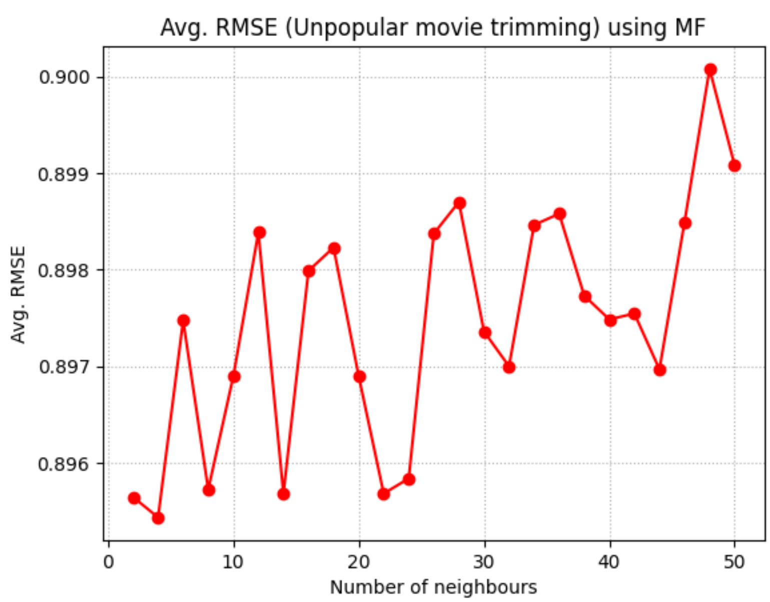
Minimum Average RMSE for Popular Movies using MF: 0.8554279056803425 for k = 32

Unpopular Data:
Using cross validate:



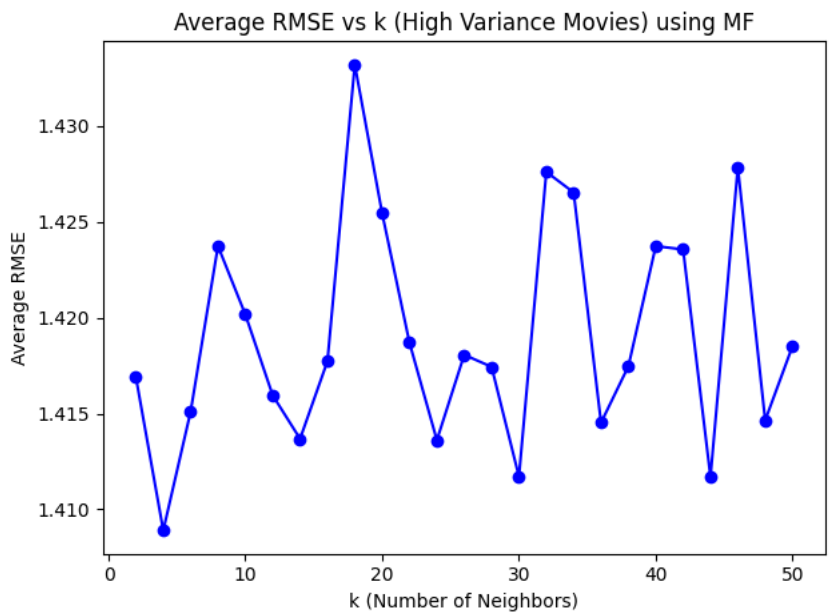
Minimum Average RMSE for Unpopular Movies using MF: 0.8956594599243235 for k = 14

Using KFold:



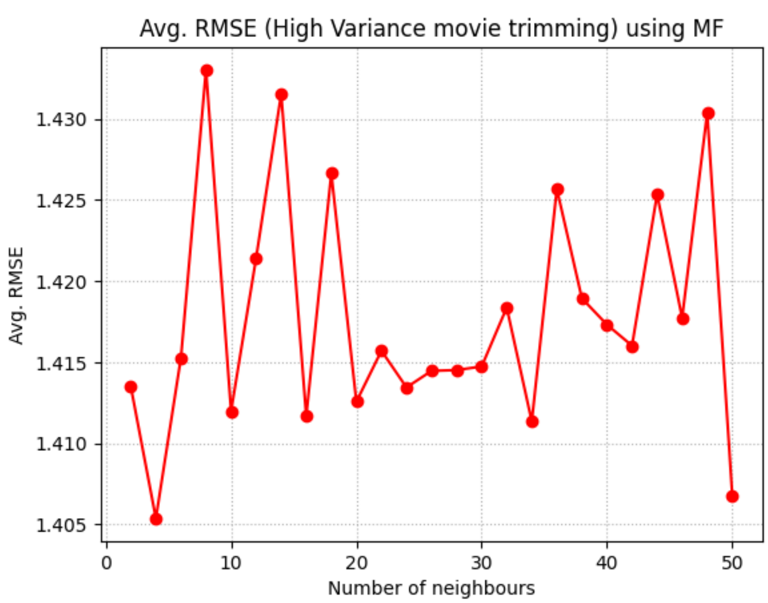
Minimum Average RMSE for Unpopular Movies using MF: 0.8954357823976574 for k = 4

High Variance Data:
Using cross validate:



Minimum Average RMSE for High Variance Movies using MF: 1.408903986652478 for k = 4

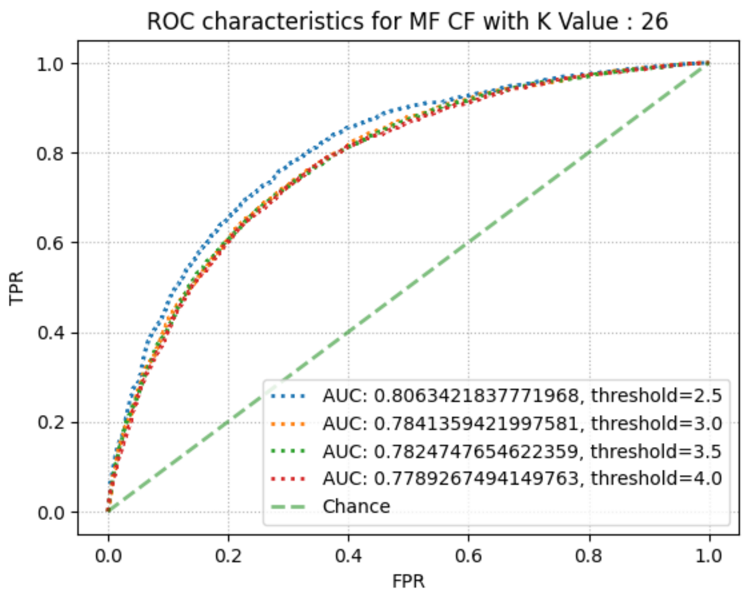
Using KFold:



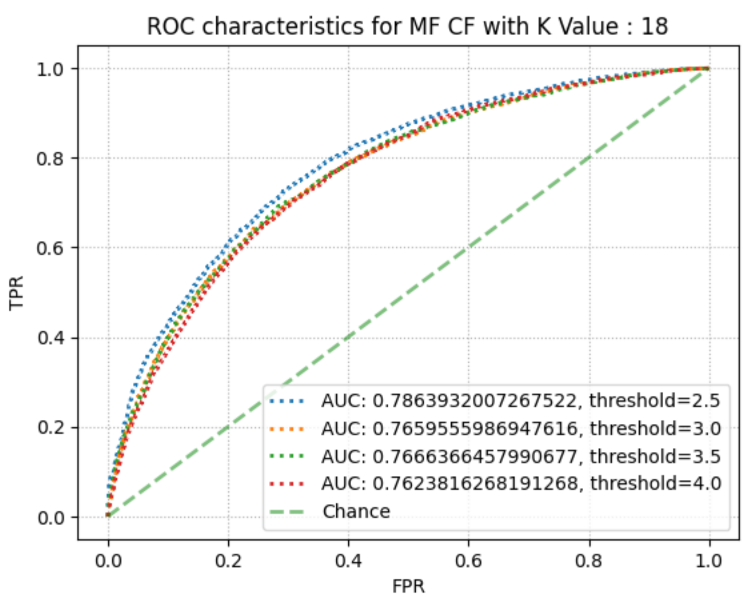
Minimum Average RMSE for High Variance Movies using MF: 1.405366090924081 for k = 4

- 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.

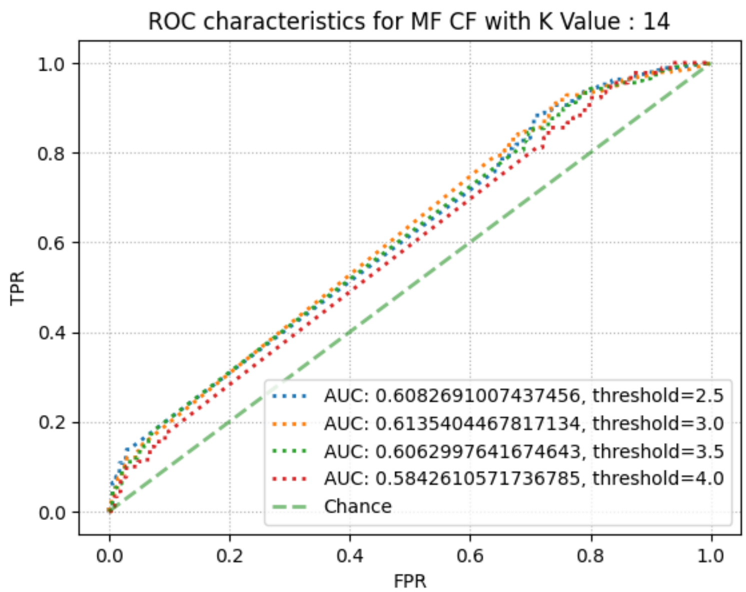
Untrimmed Data:



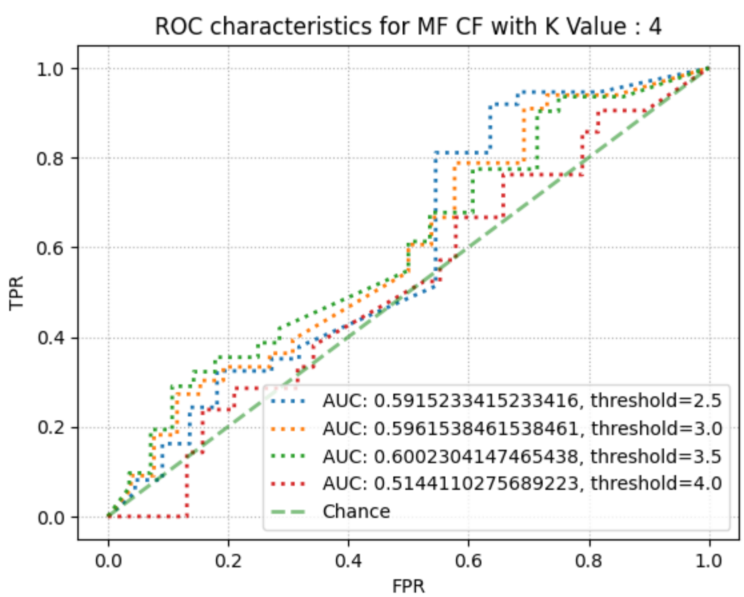
Popular Data:



Unpopular Data:



High Variance Data:



11 Question 11

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.

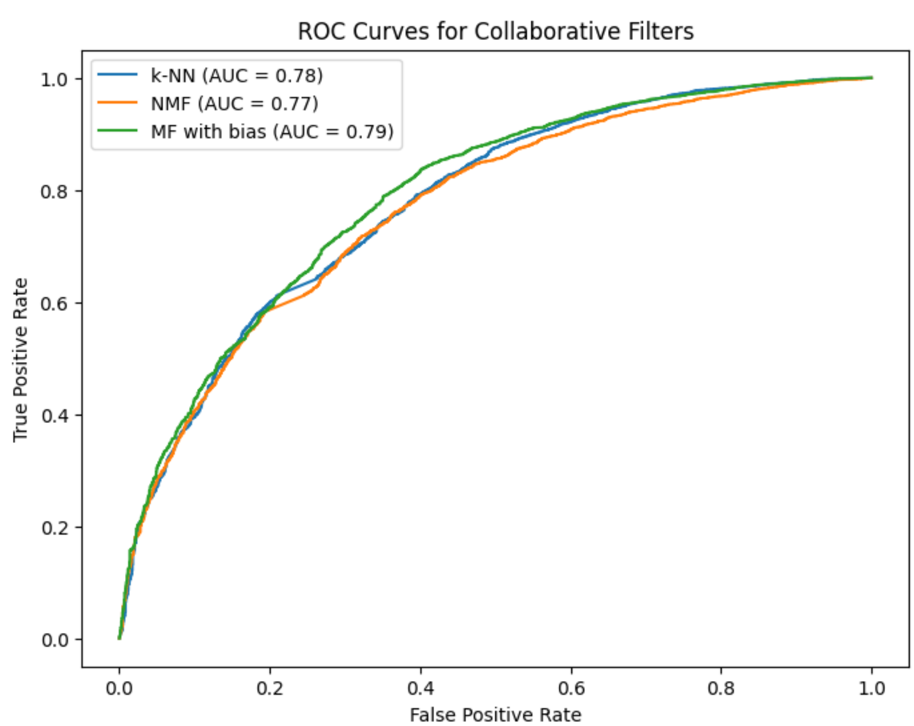
Answer:-

- 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

12 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.

Answer:-



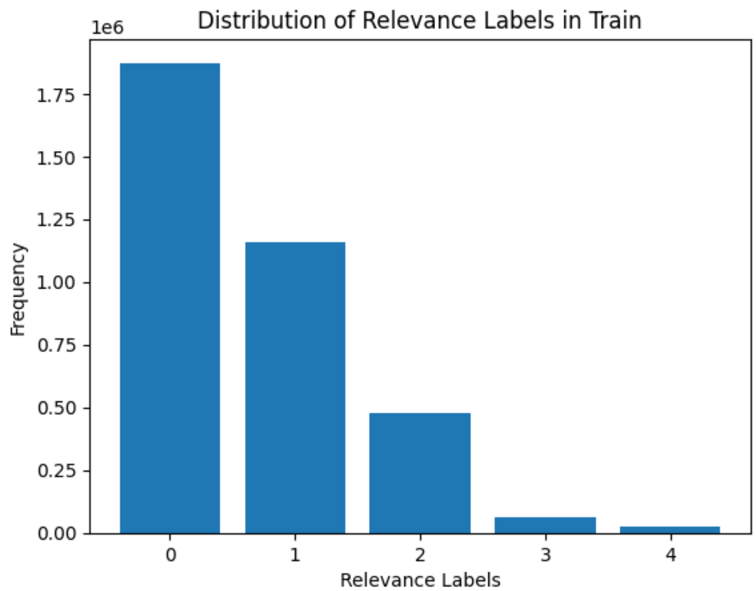
13 Question 13

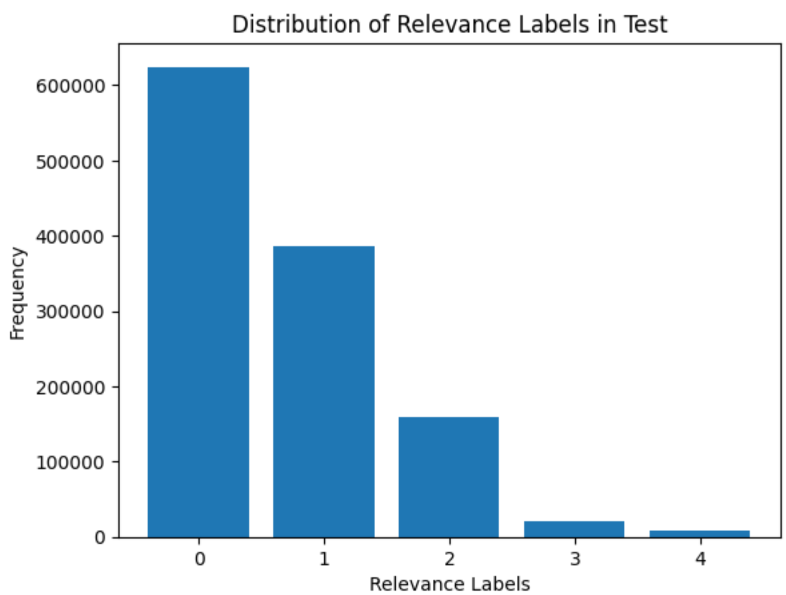
Data Understanding and Preprocessing:

- 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.

Answer:-

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





14 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.

Answer:-

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

15 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.boosterv.feature importance(importance type='gain') as demonstrated here for retrieving importance score per feature. You can also find helper code in the provided notebook.

Answer:-

```
--------------------------------------------------------------------------
--------------------------------------------------------------------------

Top 5 most important features for Fold 1:
Column_133: 23856.7030
Column_7: 4248.5464
Column_107: 4135.2444
Column_54: 4078.4632
Column_129: 3635.0370

--------------------------------------------------------------------------
--------------------------------------------------------------------------

Top 5 most important features for Fold 2:
Column_133: 23578.9083
Column_7: 5157.9649
Column_54: 4386.6698
Column_107: 4094.0122
Column_129: 4035.0707

--------------------------------------------------------------------------
--------------------------------------------------------------------------

Top 5 most important features for Fold 3:
Column_133: 23218.0754
Column_54: 4991.3034
Column_107: 4226.8074
Column_129: 4059.7525
Column_7: 3691.7923

--------------------------------------------------------------------------
--------------------------------------------------------------------------

Top 5 most important features for Fold 4:
Column_133: 23796.8997
Column_7: 4622.6230
Column_54: 3883.4817
Column_129: 3356.8470
Column_128: 3207.5755

--------------------------------------------------------------------------
--------------------------------------------------------------------------
Top 5 most important features for Fold 5:
Column_133: 23540.9424
Column_7: 4794.9452
Column_54: 4079.6086
Column_107: 3514.8358
Column_129: 3209.0584

--------------------------------------------------------------------------
--------------------------------------------------------------------------
```

16 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.

Answer:-

```
--------------------------------------------------------------------------
--------------------------------------------------------------------------
For Fold 1:
nDCG@3: 0.37967488460229254
nDCG@5: 0.3850299691938894
nDCG@10: 0.4083636029390886
--------------------------------------------------------------------------
--------------------------------------------------------------------------
For Fold 2:
nDCG@3: 0.3739449461043477
nDCG@5: 0.3819536013454118
nDCG@10: 0.4045026694861529
--------------------------------------------------------------------------
--------------------------------------------------------------------------
For Fold 3:
nDCG@3: 0.3823833692306899
nDCG@5: 0.3899961152757789
nDCG@10: 0.4116363812695088
--------------------------------------------------------------------------
--------------------------------------------------------------------------
For Fold 4:
nDCG@3: 0.381976845689231
nDCG@5: 0.39281004672399866
nDCG@10: 0.4121071637228934
--------------------------------------------------------------------------
--------------------------------------------------------------------------
For 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.

Answer:-

```
----------------------------------------------------------------------------
----------------------------------------------------------------------------
For Fold 1:
nDCG@3: 0.4542530326427776
nDCG@5: 0.46265744453383695
nDCG@10: 0.4819713060930259
----------------------------------------------------------------------------
----------------------------------------------------------------------------
For Fold 2:
nDCG@3: 0.457290225801309
nDCG@5: 0.4602669061430629
nDCG@10: 0.4772534003341443
----------------------------------------------------------------------------
----------------------------------------------------------------------------
For Fold 3:
nDCG@3: 0.4497901754692966
nDCG@5: 0.4586395637756899
nDCG@10: 0.4774361560299901
----------------------------------------------------------------------------
----------------------------------------------------------------------------
For Fold 4:
nDCG@3: 0.46063528567047524
nDCG@5: 0.46734032124732483
nDCG@10: 0.48888147783549574
----------------------------------------------------------------------------
----------------------------------------------------------------------------
For 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.