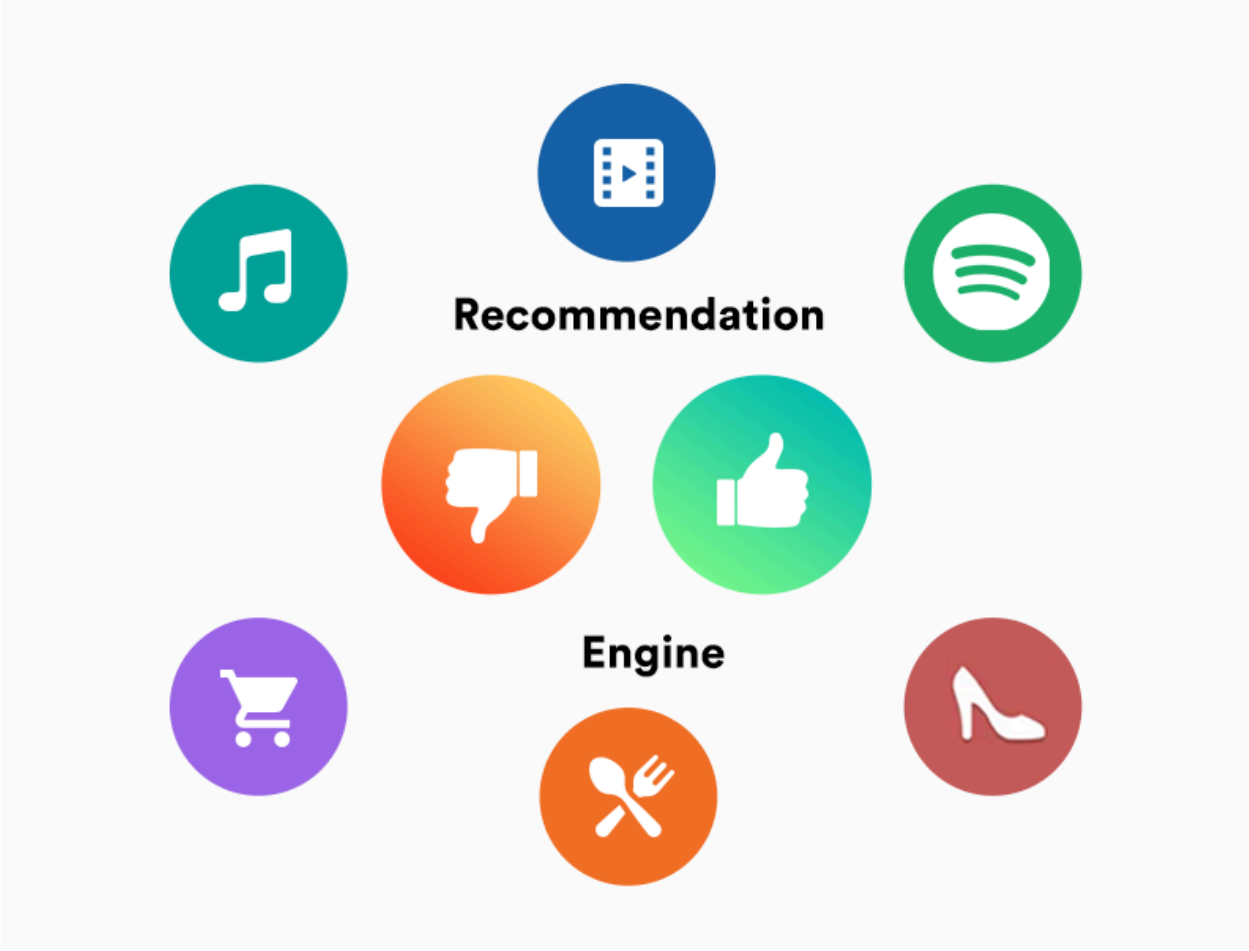
Project 3: Recommendation Systems

Authors: Alex Chen, Jiamin Xu

Electrical and Computer Engineering 219: Winter 2024

Professor Vwani Roychowdhury

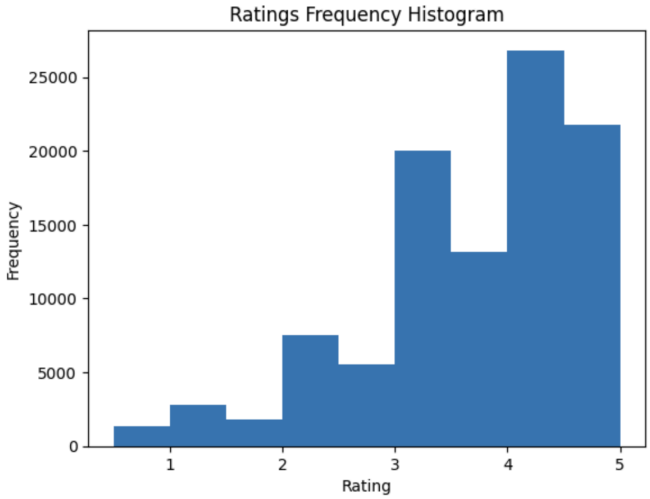


Question 1

Part A

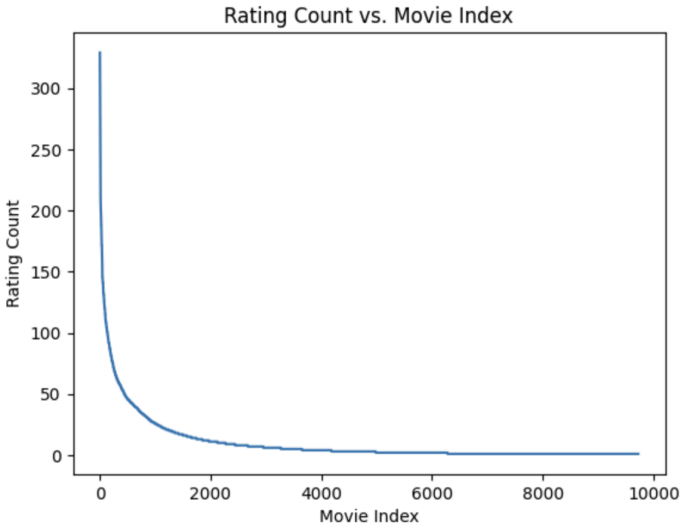
The sparsity of the movie rating dataset is 0.0170. There are 100836 total ratings, 610 different users, and 9724 different movies.

Part B

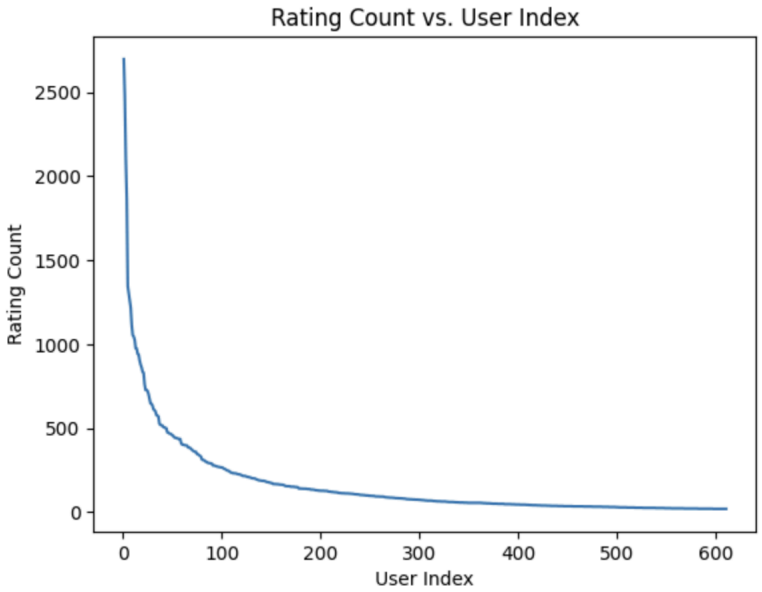


The histogram is skewed to the left – the majority of ratings are higher, with a median of around 3.5 or 4, but there are many outliers that fall in lower bins, pulling the mean below the median.

Part C



Part D



Part E

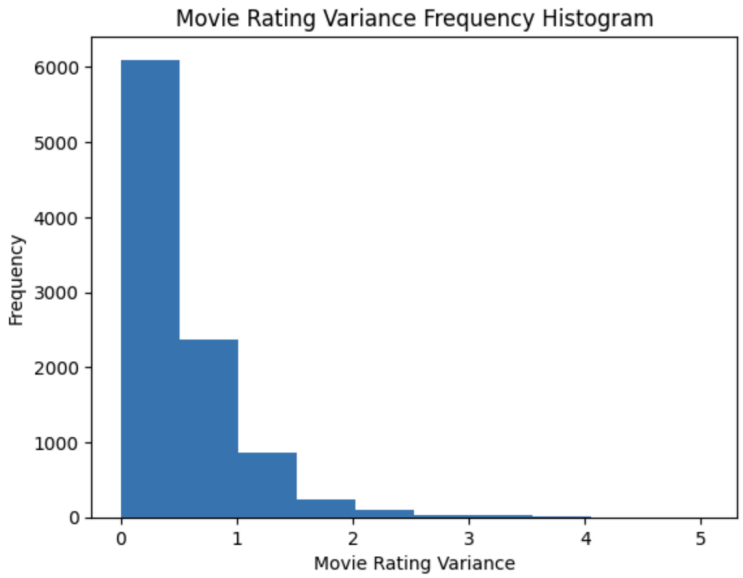
As we can see, more popular movies have larger numbers of ratings, while less popular movies have much less ratings. The popularity quickly drops, seemingly exponentially.

With rating count versus user, the shape of the curve is similar and indicates a similar observation – some users make many more ratings than others, which is expected as in the industry, some people are more invested and will watch more movies and perform more ratings than others.

These shapes / relationship in the data indicate that the data is extremely sparse. Firstly, in terms of the R matrix, or the user versus movie matrix, the rows are very sparse. This can be seen because only few users have performed many ratings. There are 9724 different ratings, and only an extremely small portion of users are even close to rating a quarter of them. In terms of the columns, they are sparse as well. There are only an extremely small proportion of movies, for which the number of different users rating them exceed a quarter of total users.

Overall, this shows how sparse the data is, overall, which is a downside to machine learning since it makes it harder for a model to extract meaningful relationships, and it may lead to overfitting as well. The recommendation process requires more intricate navigation in this case, and collaborative filtering and/or content-based filtering will have great value.

Part F



The histogram is skewed to the right – most movies have a lower variance, with a median variance of around 0.5, while a good amount of outliers exist, pulling the mean variance to the right. However, the skew is not too large given the large number of movies with low variance, as over 2/3s of movies fall in the 0 - 0.5 variance bin.

Question 2

Part A

$\mu_u = (\Sigma r_{uk}) / \text{size}(I_u)$ where the summation runs for k over the values in I_u. This is essentially summing up all ratings that the user has made (in I_u) and dividing by total number of ratings.

Part B

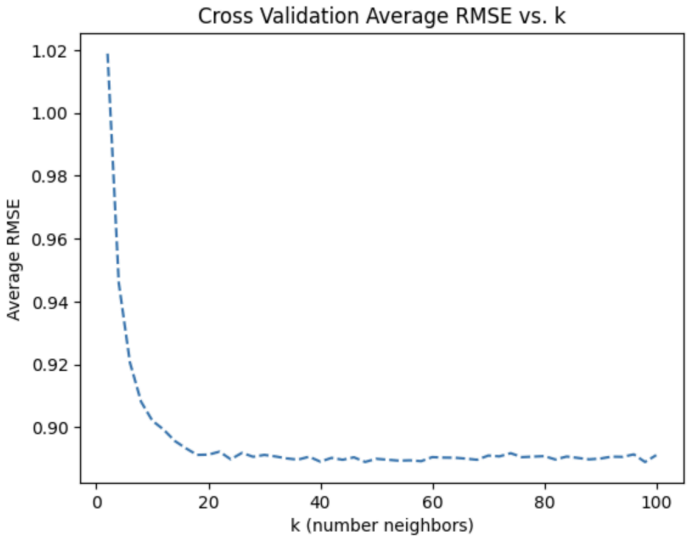
Yes, it is possible for the intersection of I_u and I_v to be the empty set. This is saying that it is possible for two users to not both have rated the same movie.

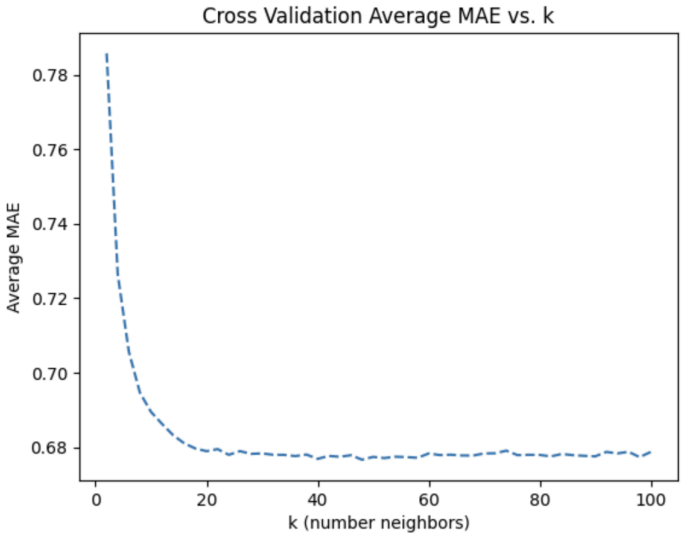
Question 3

Mean-centering plays the role of ensuring that the tendency of a user to rate higher or lower than another user, on average, does not affect the predicted rating of the user in question. This is because if a user tends to rate movies higher than the user in question and we did not center this rating, then the predicted rating for the user in question would be higher than it should be. Instead, it is more accurate to think about how a user rates relative to their average rating. Thus, when performing mean-centering, we get more of this relative relationship – if a user rates a movie higher than their average score, then we take this into account with respect to the user in

question. Likewise, if a user rates a movie lower than their average score, the user in question will have a predicted score lower than their average.

Question 4



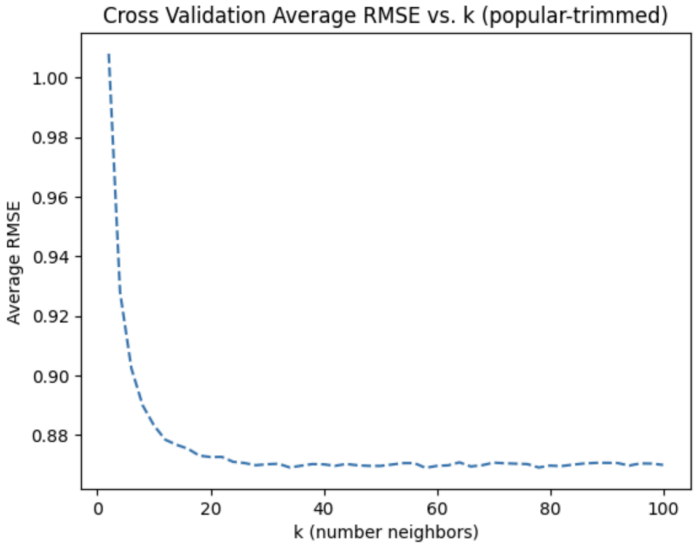


Question 5

For both metrics, the minimum k value is approximately 20. The average RMSE steady state value is approximately 0.88, while the average MAE steady state value is approximately 0.68.

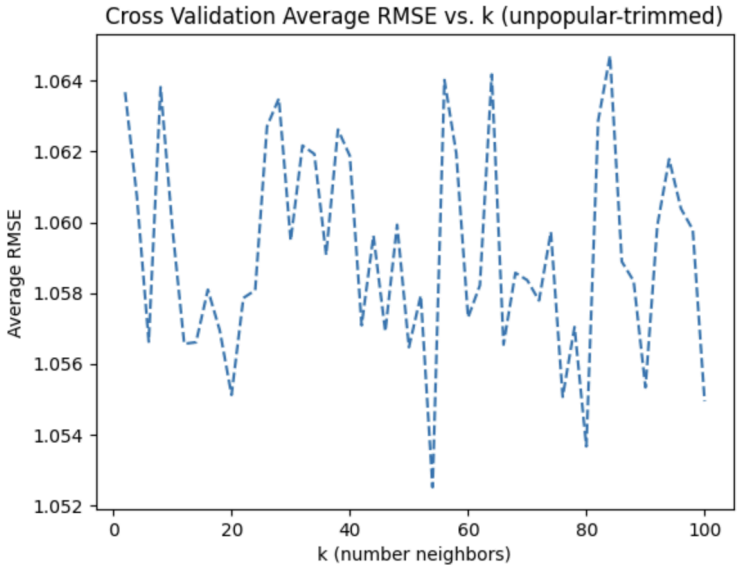
Question 6

Popular-trimmed dataset



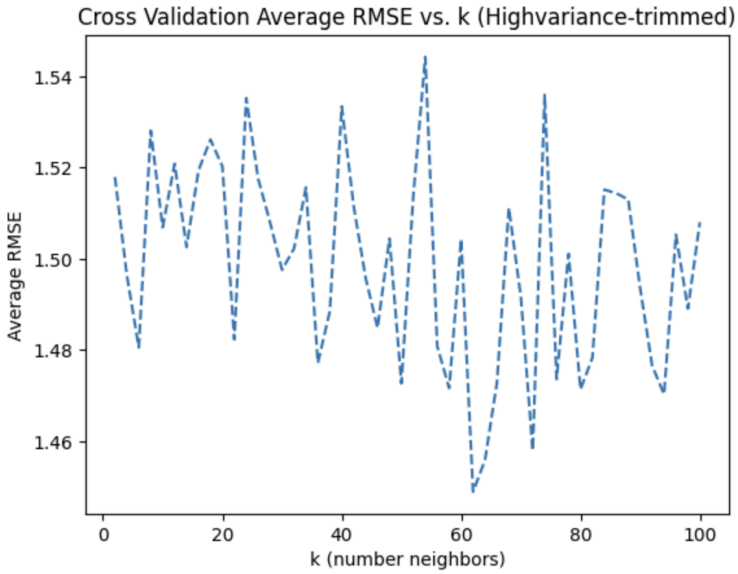
Popular-trimmed: minimum average RMSE -- 0.8689901190667406
Popular-trimmed: minimum average RMSE k -- 58

Unpopular-trimmed dataset



Unpopular-trimmed: minimum average RMSE -- 1.0525069721069475
Unpopular-trimmed: minimum average RMSE k -- 54

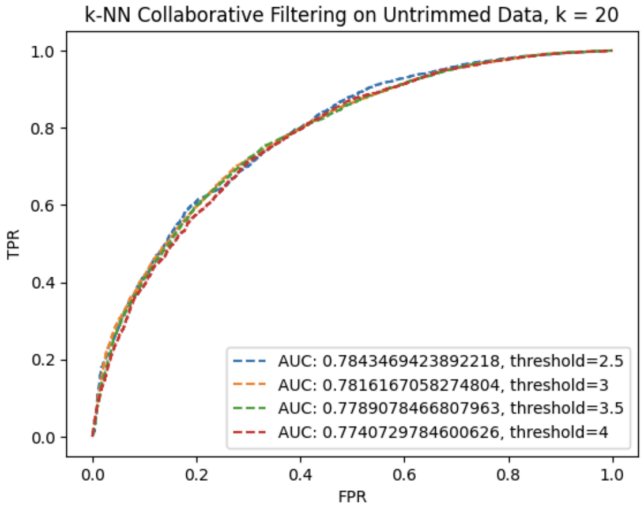
Highvariance-trimmed dataset

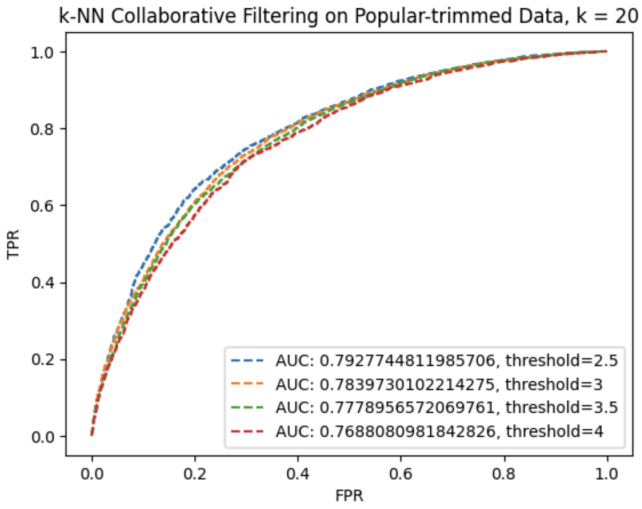


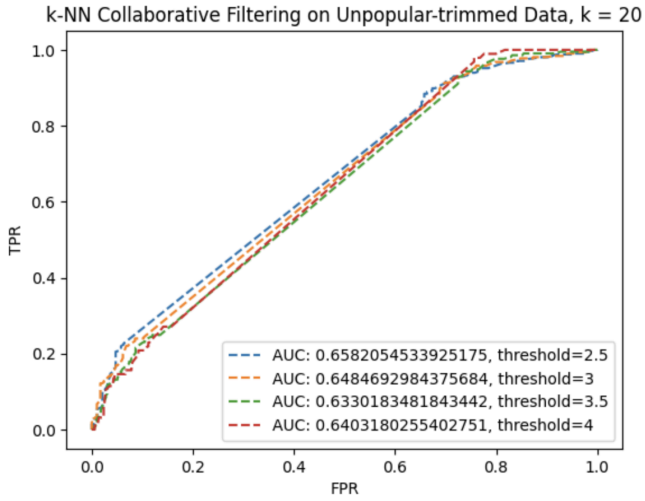
Highvariance-trimmed: minimum average RMSE -- 1.448911959989735
Highvariance-trimmed: minimum average RMSE k -- 62

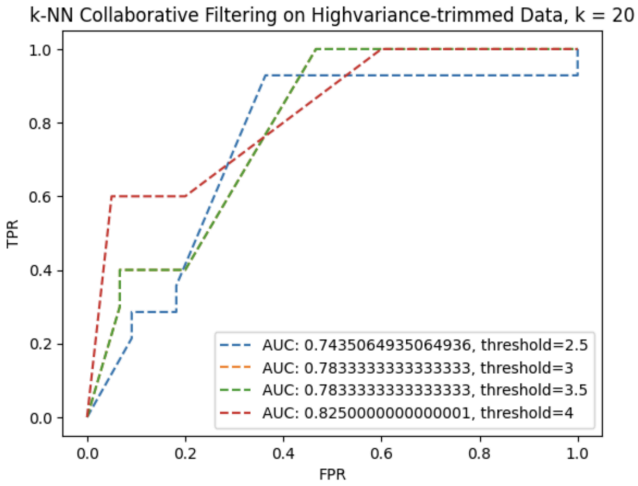
Generally, it can be seen that the order of trimmed-datasets from lowest to highest minimum average RMSE is popular-trimmed, unpopular-trimmed, and then highvariance-trimmed. This makes sense because only using movies that have more ratings is a more consistent dataset. Specifically, not only will the “k-nearest” neighbors of any user only be determined based on overlap of movies that have many ratings, but also, when predicting a movie’s rating based on those neighbors, the probability that those neighbors will also have had watched those movies is much larger. On the contrary, in a situation such that we only use unpopular movies, when looking at the 3-nearest neighbors (assume k = 3) for a certain user, the movie we are trying to predict may not have been watched by any of those 3 neighbors. Especially when we trim the dataset to unpopular movies with less than a few ratings, and when k is much large, almost all of those neighbors will not have watched the movie in question. In these cases, the predicted rating would basically default to the average rating of the user in question. This will likely be a skewed and inaccurate prediction. This explains the commonly occurring peaks / spikes in the second plot corresponding to unpopular-trimmed movies. Using certain neighbor counts will luckily lead to a more accurate rating, but generally, there is no consistent rating prediction. Lastly, for highvariance-trimmed, yet popular movies, although we keep the requirement of popular movies, since those movies have high variance in ratings, this means a neighbors ratings of a movie will not necessarily translate to the rating of the user in question. A movie can be greatly above one neighbor’s average rating, but greatly below another neighbor’s average rating, thus confuscating the prediction for the user in question and creating the varying accuracies seen in the last plot.

ROC curves









The curvatures seem appropriate as well, given the contexts. With untrimmed data and popular-trimmed data, the curve is concave, while the latter two curves are piecewise and linear at many points. The reasoning behind this is same as why the average RMSE graphs are as they are.

As for threshold, it seems as threshold increases above 2.5, for the untrimmed and popular-trimmed data, AUC has monotonically decreasing behavior, while it has generally increasing behavior for the unpopular-trimmed and highvariance-trimmed data. For the first two cases, this is logical because 2.5 is a halfway point that is in the middle and thus has less bias.

Question 7

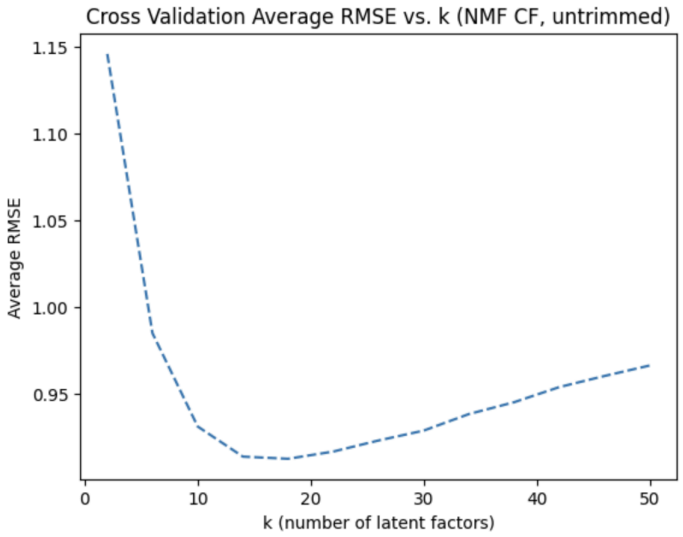
Typically, a sum of squares problem is convex. There is one global minimum, and it is continuous, convex, and unimodal. However, although the optimization function is a sum of squares, it is not convex. The added term of the product of U and V^T introduces non-linearity

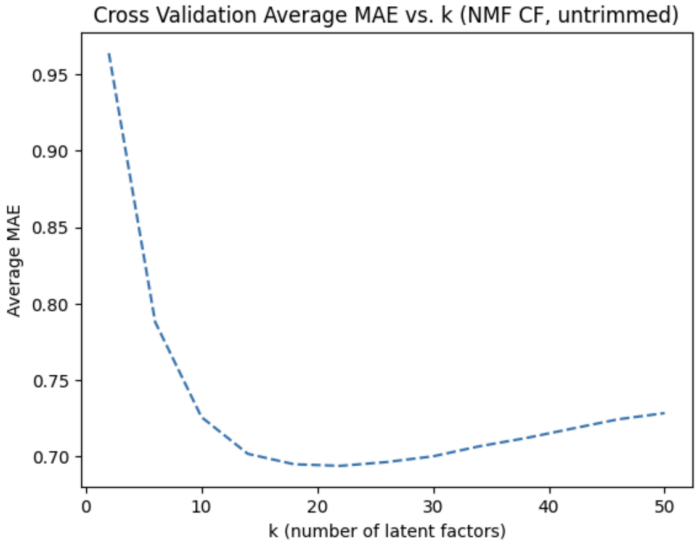
and causes the overall function to not be convex. Furthermore, this is confirmed because both stochastic gradient descent (SGD) and alternating least-squares (ALS) are not guaranteed to converge to an optimal minimum, meaning the function is not convex.

ALS fixes U as a constant, then and attempts to solve for the values of V that minimize the objective function. When this occurs, the problem is reduced to a least squares optimization. Then, once the optimal V is found, ALS goes back to U and attempts to find the new optimal value. Thus, ALS alternates between two least-squares optimizations.

Question 8

Part A





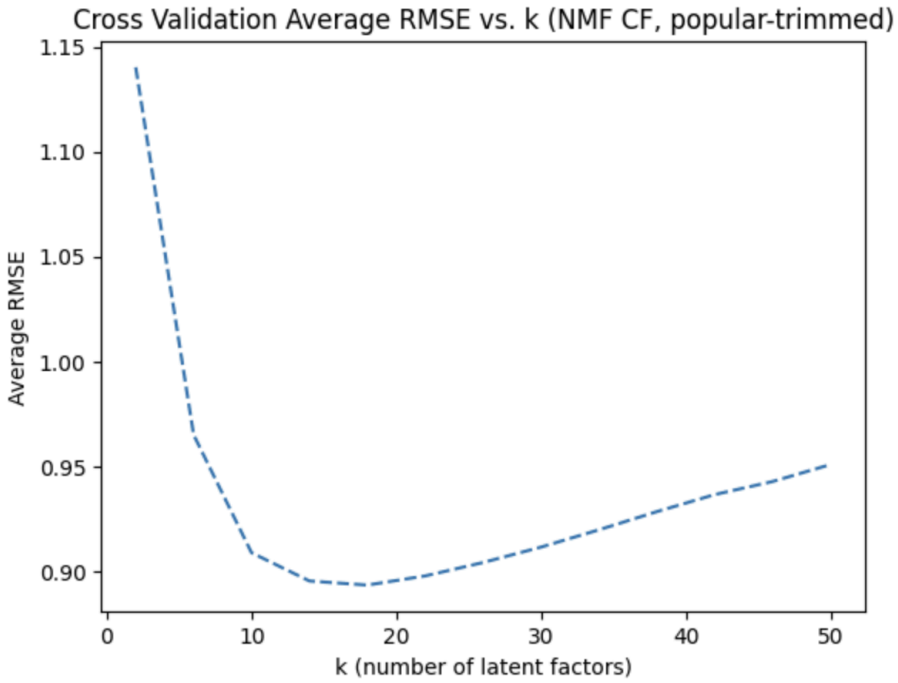
Part B

NMF CF, untrimmed: minimum average RMSE -- 0.914050159180191
NMF CF, untrimmed: minimum average RMSE k -- 18

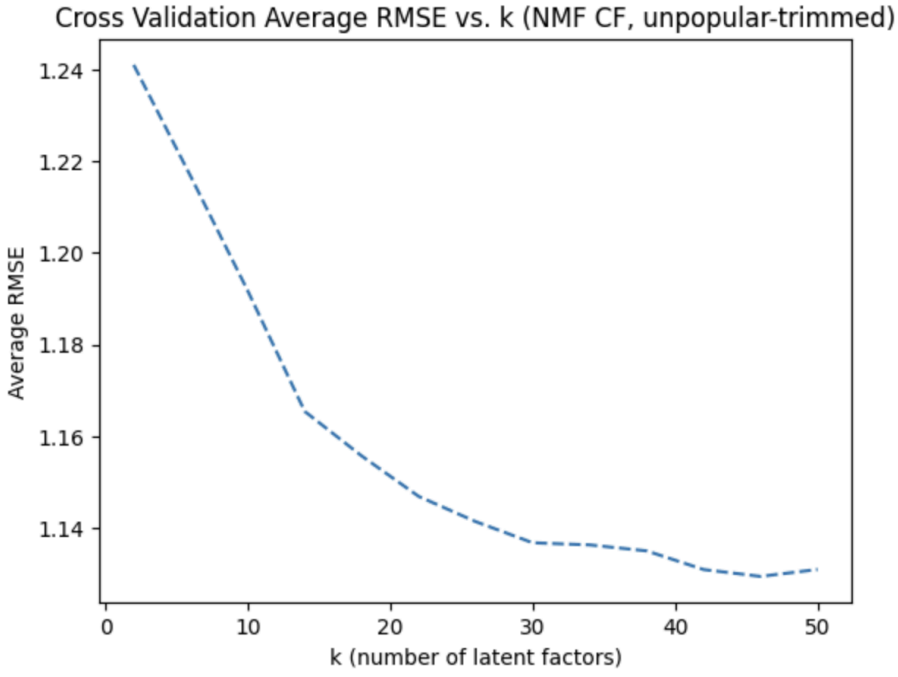
NMF CF, untrimmed: minimum average MAE -- 0.6950673401429037
NMF CF, untrimmed: minimum average MAE k -- 22

The optimal number of latent factors for each case is not exactly equal to the number of different genres, 19, as computed from *movies.csv*; however, they are close.

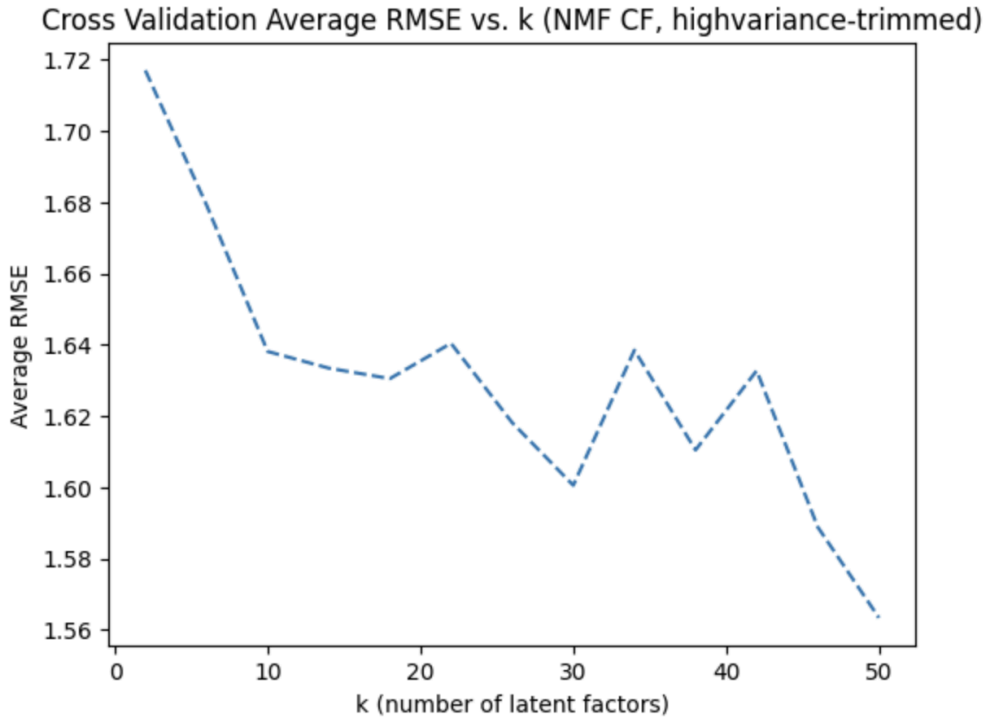
Part C



NMF CF, popular-trimmed: minimum average RMSE -- 0.8940006507697635
NMF CF, popular-trimmed: minimum average RMSE k -- 18



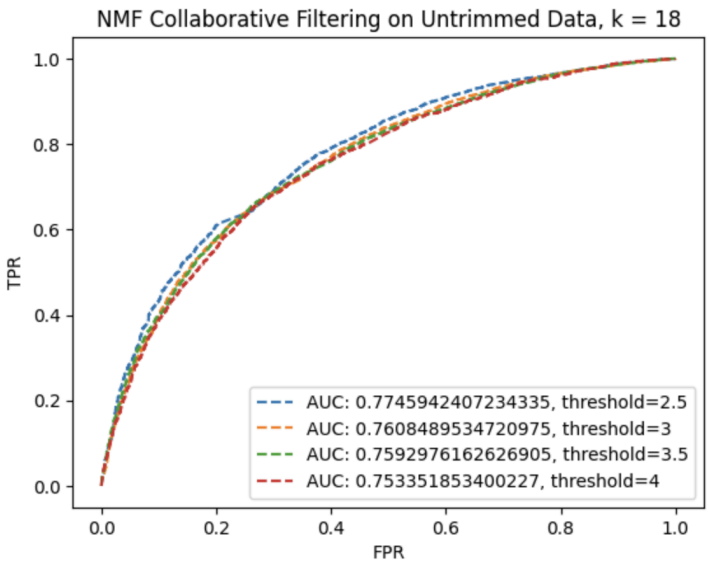
NMF CF, unpopular-trimmed: minimum average RMSE -- 1.1292495676733743
NMF CF, unpopular-trimmed: minimum average RMSE k -- 50

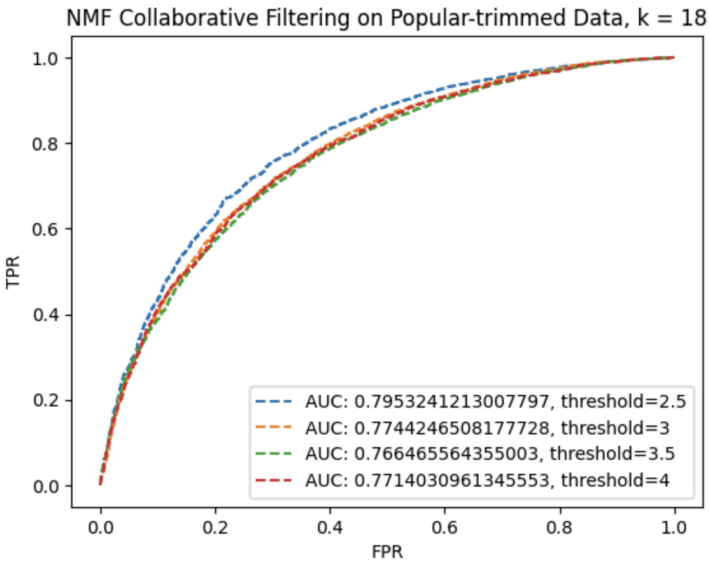


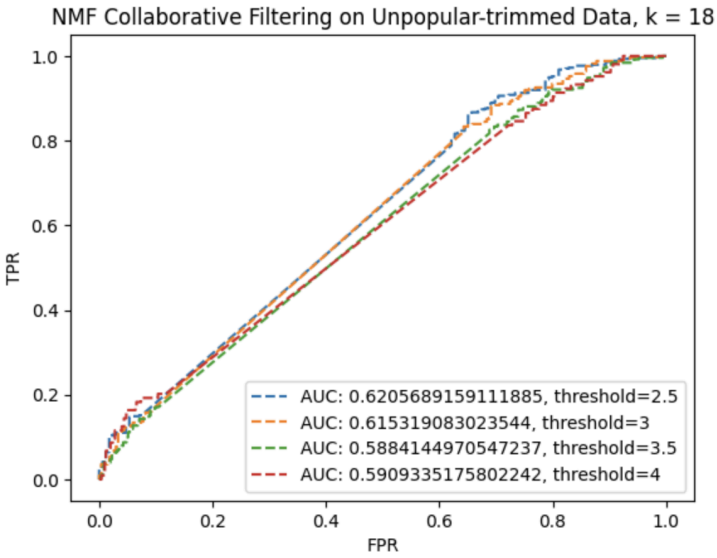
NMF CF, highvariance-trimmed: minimum average RMSE -- 1.5733716982169603
NMF CF, highvariance-trimmed: minimum average RMSE k -- 34

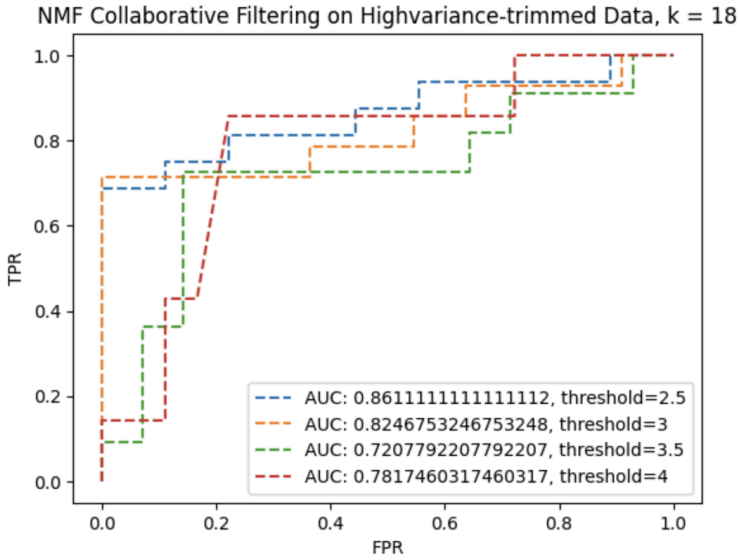
ROC Curves

For the ROC curves, I used the optimal latent factor count, 18, derived from the untrimmed dataset experiments. The following are the curves for each of the four datasets:









Question 9

Once performing NMF on *R*, then observing the genres of the top 10 movies for each movie-latent factor, it appears that each of these top 10 movies falls within a small category of movies. For example, here are the results of the first three columns. The first column is comprised of movies falling in the drama / comedy. The second column seems to be action / adventure. The third column seems to be musical / comedy. Thus, each movie-latent factor does seem to correspond to a movie genre, although not perfectly as seen by how the genres mix. Notably, as the latent factor increases (from first, to second, to third, etc.), it seems as if the homogeneity of movie genres decreases and each factor represents a genre to a weaker extent. However, we did no formal test to confirm this.

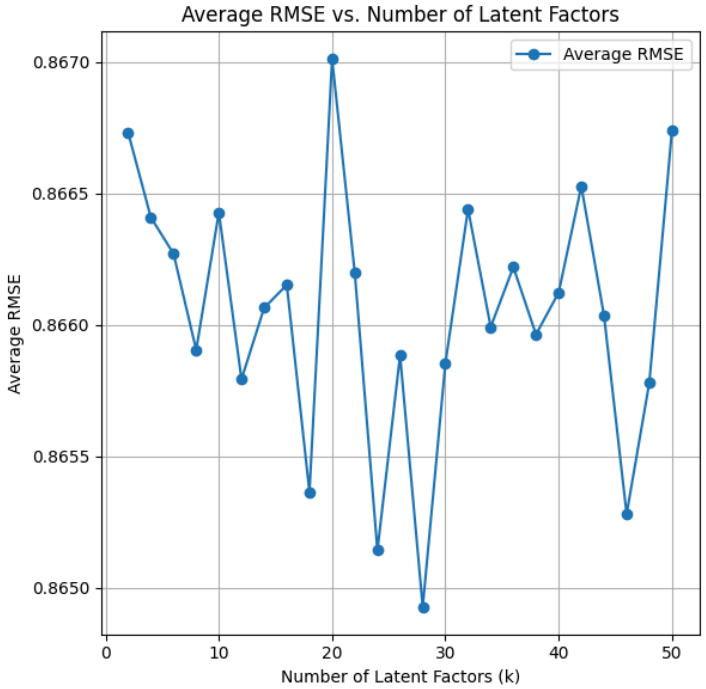
Column 0
Drama
Comedy
Crime|Drama|Thriller
Comedy
Crime|Drama
Comedy|Romance
Drama
Drama
Drama
Comedy|Horror|Sci-Fi

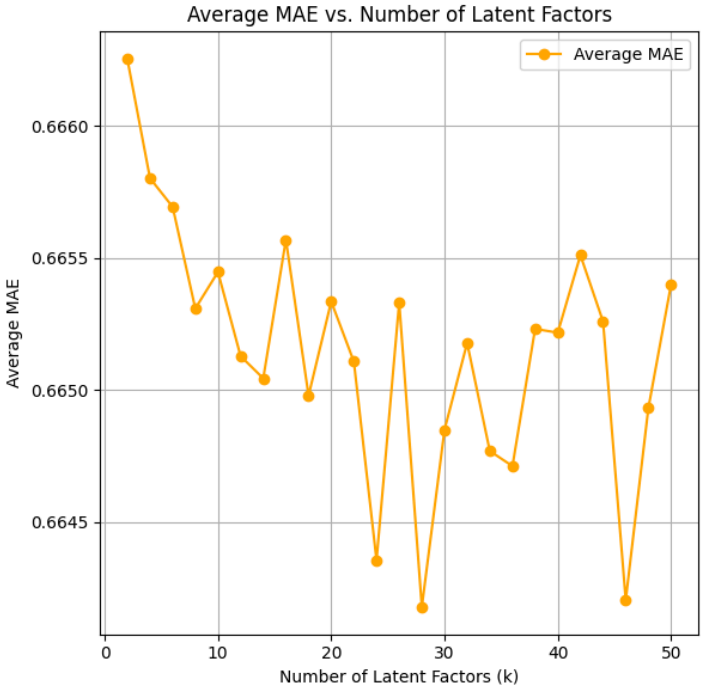
Column 1
Comedy|Drama
Horror|Thriller
Drama|Romance

Action|Adventure|Comedy
Comedy
Action|Adventure|Romance
Action|Drama
Drama
Action|Fantasy|Sci-Fi|Thriller|War
Action|Crime|Drama

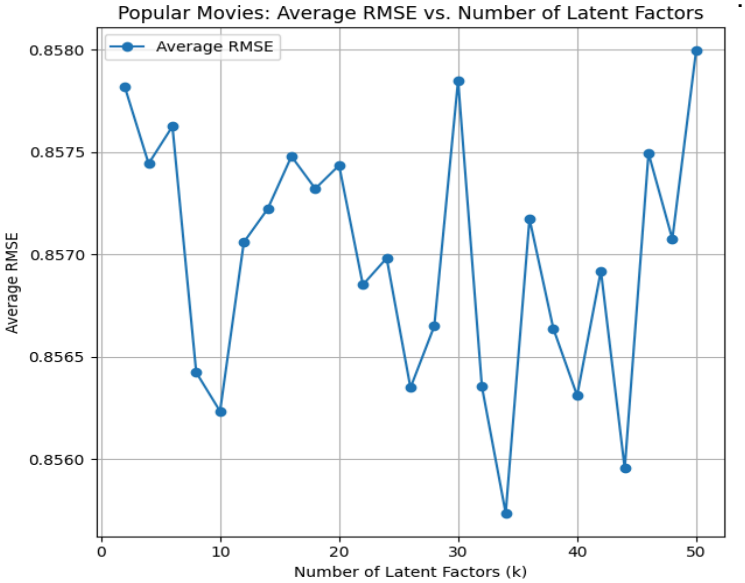
Column 2
Musical|Romance|Western
Action|Adventure|Fantasy
Adventure|Drama
Comedy|Crime|Drama
Drama
Action|Fantasy|Thriller
Animation|Children|Musical
Comedy
Action|Sci-Fi|Thriller
Adventure|Children

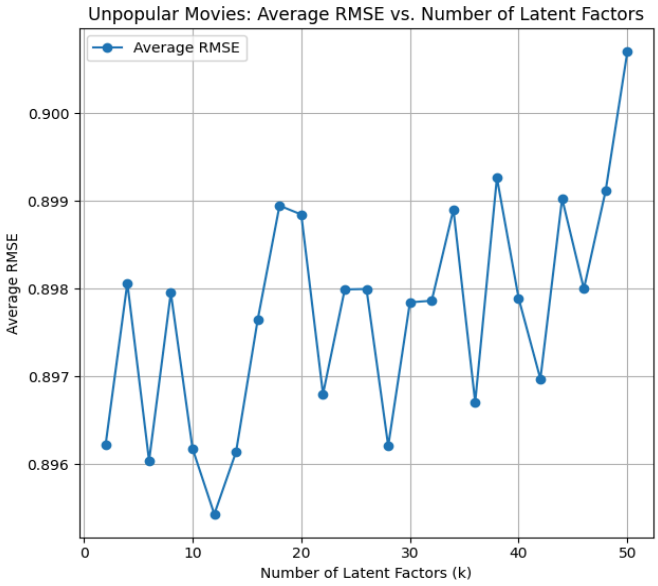
Question 10

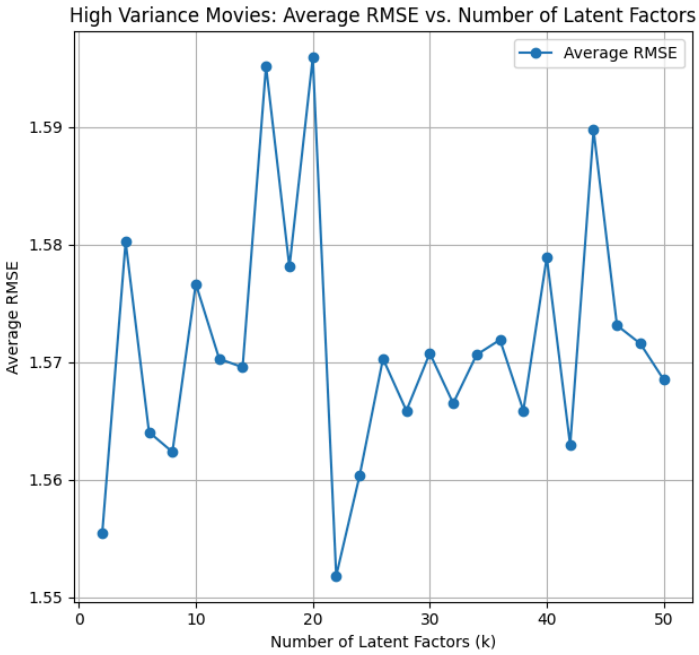




Minimum Average RMSE: 0.8649280737069566, at k = 28
Minimum Average MAE: 0.664179490934243, at k = 28
Optimal Number of Latent Factors: 28; greater than the number of genres, 19



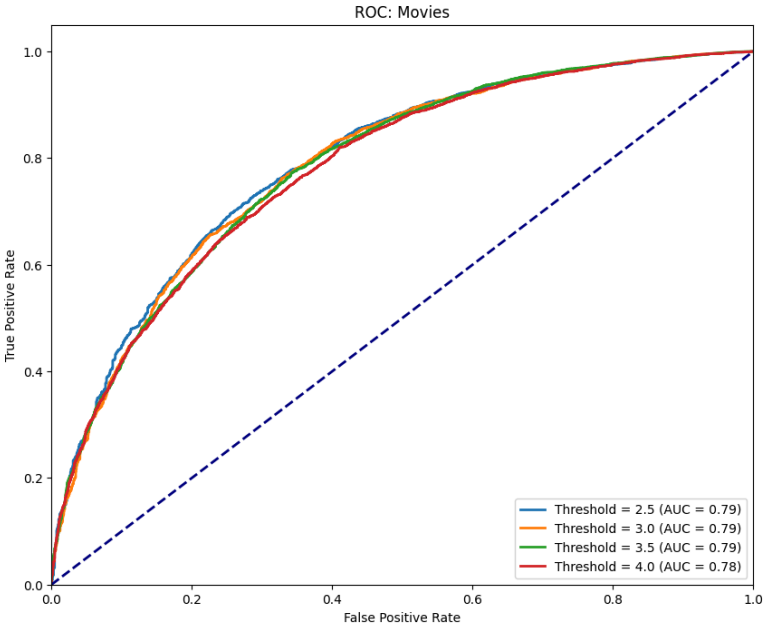


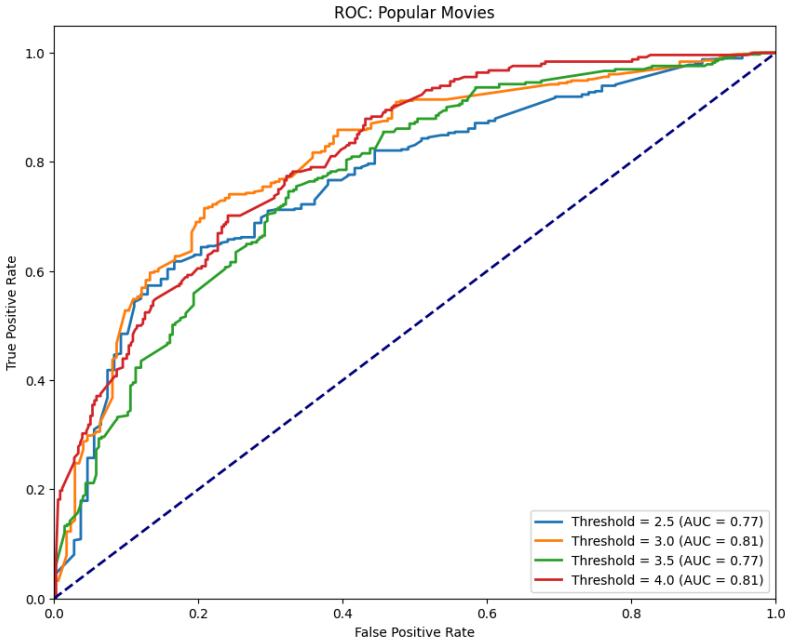


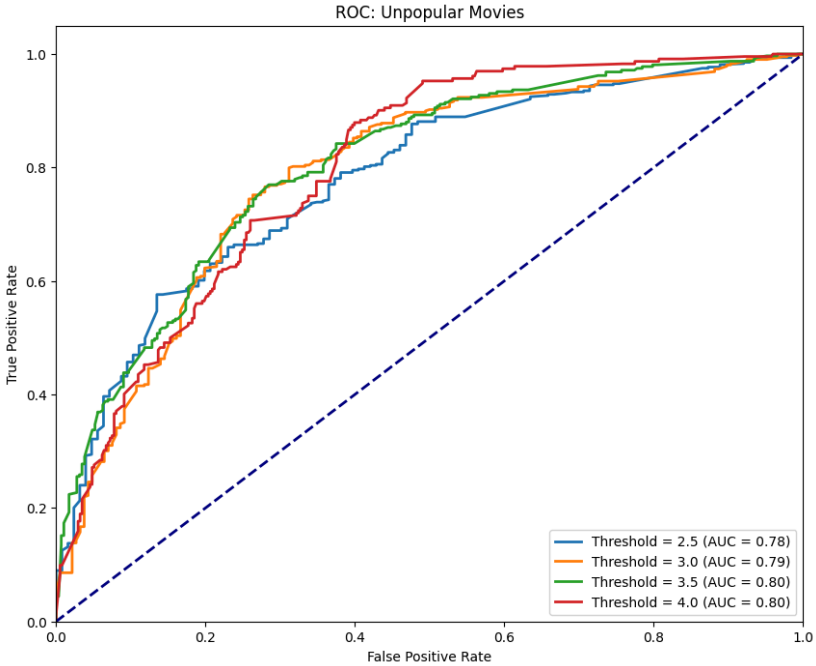
Popular: Minimum Average RMSE: 0.8557358259848286, at k = 34

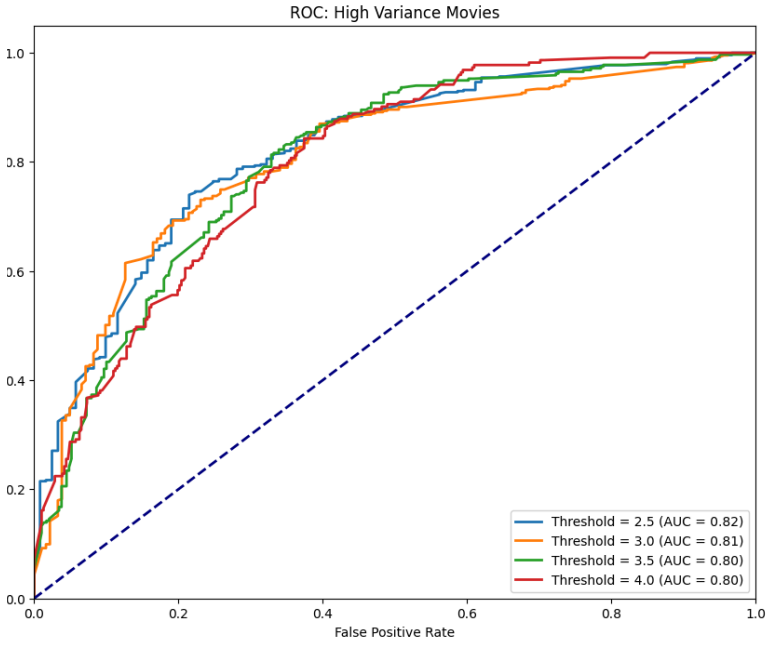
Unpopular: Minimum Average RMSE: 0.8954302497830235, at k = 12

High Variance: Minimum Average RMSE: 1.5518130198791493, at k = 22


ROC: Movies
True Positive Rate
False Positive Rate
Threshold = 2.5 (AUC = 0.79)
Threshold = 3.0 (AUC = 0.79)
Threshold = 3.5 (AUC = 0.79)
Threshold = 4.0 (AUC = 0.78)


ROC: Popular Movies
True Positive Rate
False Positive Rate
Threshold = 2.5 (AUC = 0.77)
Threshold = 3.0 (AUC = 0.81)
Threshold = 3.5 (AUC = 0.77)
Threshold = 4.0 (AUC = 0.81)


ROC: Unpopular Movies
True Positive Rate
False Positive Rate
Threshold = 2.5 (AUC = 0.78)
Threshold = 3.0 (AUC = 0.79)
Threshold = 3.5 (AUC = 0.80)
Threshold = 4.0 (AUC = 0.80)

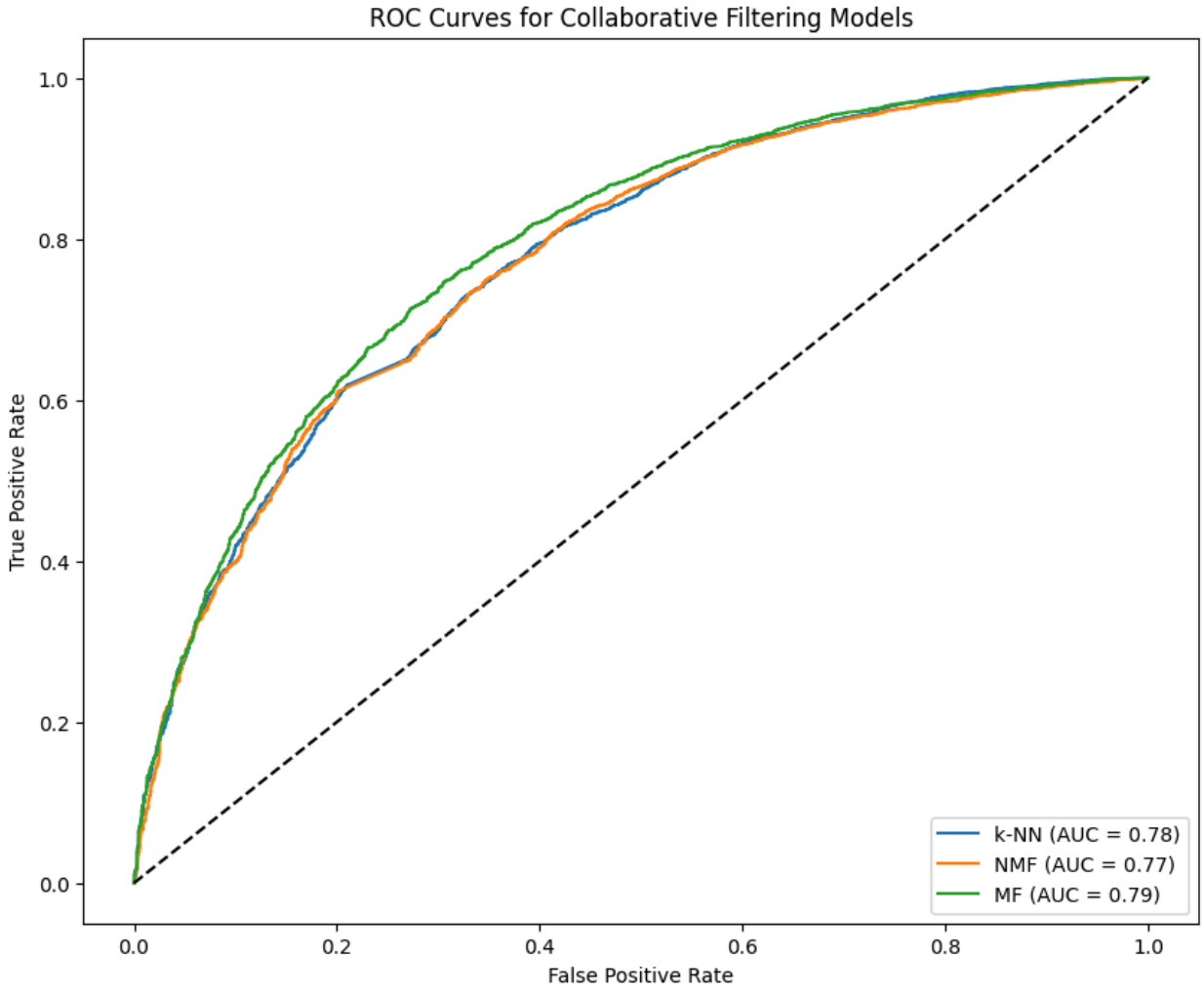

ROC: High Variance Movies
False Positive Rate
Threshold = 2.5 (AUC = 0.82)
Threshold = 3.0 (AUC = 0.81)
Threshold = 3.5 (AUC = 0.80)
Threshold = 4.0 (AUC = 0.80)

Question 11

Original: Average RMSE: 0.9411424347343129
Popular: Average RMSE: 0.941208142060875
Unpopular: Average RMSE: 0.9408932891750219
High Variance: Average RMSE: 0.9411758642153482

Question 12

The three filters roughly perform the same in predicting movie ratings.



Question 13

```
Number of unique queries in training data: 6000
Number of unique queries in testing data: 2000
MSLR-WEB10K/Fold1/: distribution of relevance labels in training data: {0: 377957, 1: 232569, 2: 95082, 3: 12658, 4: 5146}
MSLR-WEB10K/Fold1/: distribution of relevance labels in testing data: {0: 124784, 1: 77896, 2: 32459, 3: 4450, 4: 1932}
Number of unique queries in training data: 6000
Number of unique queries in testing data: 2000
MSLR-WEB10K/Fold2/: distribution of relevance labels in training data: {0: 373029, 1: 230368, 2: 95117, 3: 12814, 4: 5355}
MSLR-WEB10K/Fold2/: distribution of relevance labels in testing data: {0: 126450, 1: 78016, 2: 31875, 3: 4053, 4: 1594}
Number of unique queries in training data: 6000
Number of unique queries in testing data: 2000
MSLR-WEB10K/Fold3/: distribution of relevance labels in training data: {0: 371725, 1: 232302, 2: 96663, 3: 12903, 4: 5518}
MSLR-WEB10K/Fold3/: distribution of relevance labels in testing data: {0: 126088, 1: 75962, 2: 30913, 3: 4361, 4: 1769}
Number of unique queries in training data: 6000
Number of unique queries in testing data: 2000
MSLR-WEB10K/Fold4/: distribution of relevance labels in training data: {0: 372756, 1: 231727, 2: 96244, 3: 12712, 4: 5329}
MSLR-WEB10K/Fold4/: distribution of relevance labels in testing data: {0: 125419, 1: 78591, 2: 32294, 3: 4244, 4: 1783}
Number of unique queries in training data: 6000
Number of unique queries in testing data: 2000
MSLR-WEB10K/Fold5/: distribution of relevance labels in training data: {0: 377322, 1: 231874, 2: 95247, 3: 12864, 4: 5295}
MSLR-WEB10K/Fold5/: distribution of relevance labels in testing data: {0: 121522, 1: 75815, 2: 31910, 3: 4209, 4: 1803}
```

Question 14

Fold 1:

nDCG@3: 0.4507545733421516
nDCG@5: 0.4578584529635828
nDCG@10: 0.47526631675350844

Fold 2

nDCG@3: 0.4553792656858033
nDCG@5: 0.45858527312708586
nDCG@10: 0.47448031694933906

Fold 3

nDCG@3: 0.44264358098686574

nDCG@5: 0.45089828268619103
nDCG@10: 0.46929926593829596

Fold 4

nDCG@3: 0.4537689345392404
nDCG@5: 0.46129047048566274
nDCG@10: 0.48107878993967995

Fold 5

nDCG@3: 0.4602039959641346
nDCG@5: 0.46494845249290095
nDCG@10: 0.4838716356178689

Question 15

Top 5 features for Fold1

1. Feature 133 with importance score 50900.37763404846
2. Feature 7 with importance score 10627.043670654297
3. Feature 54 with importance score 7749.175968170166
4. Feature 107 with importance score 6956.837522506714
5. Feature 129 with importance score 6634.954565525055

Top 5 features for Fold 2:

1. Feature 133 with importance score 50648.40989303589
2. Feature 107 with importance score 11437.986242294312
3. Feature 54 with importance score 8245.528983592987
4. Feature 129 with importance score 7510.931382656097
5. Feature 7 with importance score 6916.260050296783

Top 5 features for Fold 3:

1. Feature 133 with importance score 50574.326808452606
2. Feature 54 with importance score 11866.20094537735
3. Feature 107 with importance score 9030.951606750488
4. Feature 129 with importance score 7123.230594158173
5. Feature 128 with importance score 6921.748646736145

Top 5 features for Fold 4:

1. Feature 133 with importance score 51285.02464342117
2. Feature 54 with importance score 9177.377975463867

3. Feature 7 with importance score 8162.877136230469
4. Feature 107 with importance score 7664.899651527405
5. Feature 128 with importance score 6281.682007312775

Top 5 features for Fold 5:
1. Feature 133 with importance score 50768.39634132385
2. Feature 7 with importance score 12024.163997650146
3. Feature 54 with importance score 7674.576921463013
4. Feature 107 with importance score 6557.297354221344
5. Feature 128 with importance score 6142.607744693756

Question 16

The nDCG scores all went down after removing the top 20 features for each fold, which is aligned with expectations. Removing the top 20 features often decreases nDCG scores because these features are likely the most informative and impactful in predicting the relevance of query-url pairs. Their removal means the model loses key information that helps distinguish between more and less relevant documents, leading to less accurate rankings and thus lower nDCG scores.

Fold 1
nDCG@3: 0.37665442320354864
nDCG@5: 0.38413822877401
nDCG@10: 0.4071277032381892

Fold 2
nDCG@3: 0.3746364353293466
nDCG@5: 0.3824200534838482
nDCG@10: 0.40465018340842035

Fold 3
nDCG@3: 0.38564342488744446
nDCG@5: 0.39209358541294026
nDCG@10: 0.414126048617549

Fold 4
nDCG@3: 0.382484205781995
nDCG@5: 0.39097072525064075
nDCG@10: 0.41385078645421675

Fold 5

nDCG@3: 0.3862838313904278
nDCG@5: 0.3931826839476099
nDCG@10: 0.4156620543154738

Removing the 60 least important features hardly changes the nDCG scores, which is aligned with expectations, as these features contribute minimally to the model's decision-making process. Their low importance indicates they have little influence on the model's ability to rank query-url pairs accurately. As a result, eliminating them does not significantly affect the model's performance, demonstrating that the model relies more heavily on a subset of highly informative features to make accurate predictions.

Fold 1:
nDCG@3: 0.4515562386034378
nDCG@5: 0.45916313597340386
nDCG@10: 0.4771597716694684

Fold 2:
nDCG@3: 0.4529432831508957
nDCG@5: 0.45679748885924365
nDCG@10: 0.47255906199425574

Fold 3:
nDCG@3: 0.4414064368308733
nDCG@5: 0.45034963186744653
nDCG@10: 0.469722748845401

Fold 4:
nDCG@3: 0.45213119122423573
nDCG@5: 0.46009315884010593
nDCG@10: 0.47995869215501286

Fold 5:
nDCG@3: 0.45836563346727144
nDCG@5: 0.46594720737839157
nDCG@10: 0.48467115112574144