Krish Kabra, 404593821
Ranjana Vittal, 605629219
Zhiyuan Chen, 605363281

Project 3: Collaborative Filtering

EE 219 - Large-Scale Data Mining: Models and Algorithms
Winter 2021

2021-02-20

3 MovieLens dataset

Question 1: Compute the sparsity of the movie rating dataset

Sparsity = 0.016999683055613623

Question 2: Plot a histogram showing the frequency of the rating values

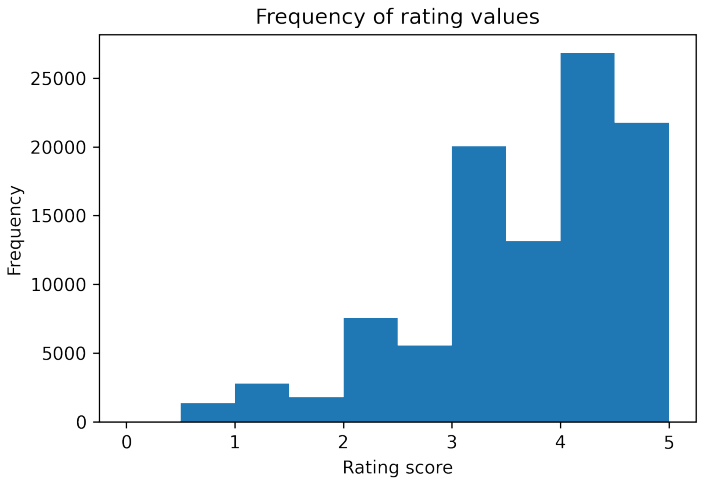

Figure 1: Histogram showing the frequency of the rating values

The histogram has an upward growth trend. This is indicative of the fact that more movies were rated highly than lowly. Most users gave ratings between 3 and 5 suggesting that a majority of the users liked the movies they watched.

Question 3: Plot the distribution of the number of ratings received among movies

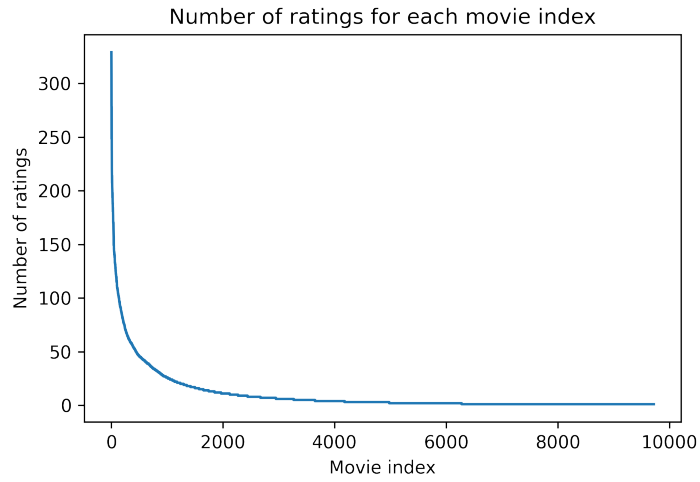

Figure 2: Distribution of the number of ratings received among movies

Question 4: Plot the distribution of ratings among users

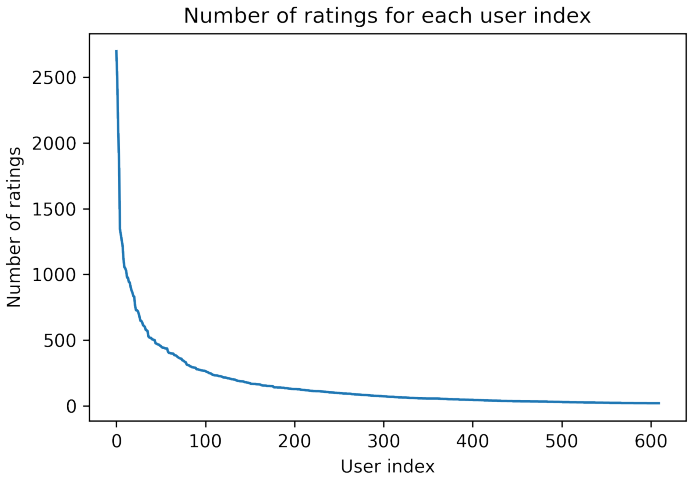

Figure 3: Distribution of ratings among users

Question 5: Explain the salient features of the distribution

The number of ratings has a reciprocal relationship with the movie index. This means that a small number of movies received a majority of the ratings. This implies that a lot of movies received a very small number of ratings. Hence, the rating matrix R is sparse which means heavy regularization needs to be added to the recommendation process to prevent overfitting and false links.

Question 6: Compute the variance of the rating values received by each movie

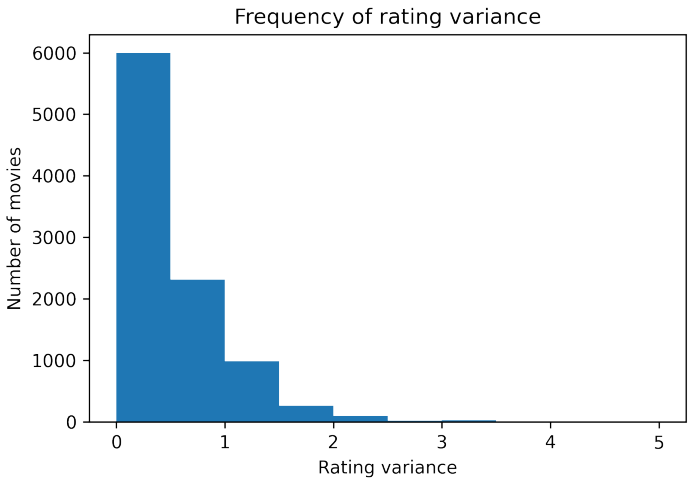

Figure 4: Variance of the rating values received by each movie

It can be seen from the histogram that the rating variance for most movies lies between 0 and 2. This means that most ratings are reliable and consistent.

4 Neighborhood-based collaborative filtering

4.2 Pearson-correlation coefficient

Question 7: Write down the formula for mean rating

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

Question 8: Explain the meaning of the union of item indices for different users

The quantity $I_u \cap I_v$ corresponds to movies that have been rated by both user u and user v. Since the rating matrix R is sparse, it is highly likely that $I_u \cap I_v = \emptyset$ because it is very likely that both users did not watch the same movie and/or did not rate the movie.

4.4 Prediction function

Question 9: Explain the reason behind mean-centering the raw ratings

Mean-centering the raw ratings in the prediction function helps reduce bias and remove extreme data points. For example, users who either rate all items highly or poorly are usually giving extreme opinions which is biased and can be considered noisy. Thus, we can make a more accurate prediction if we mean-center the ratings.

4.5 k-NN collaborative filter

Question 10: k-NN collaborative filter to predict the ratings of the movies in the MovieLens dataset

The k-NN collaborative filter has been implemented using `KNNWithMeans` from `surprise` Python package.

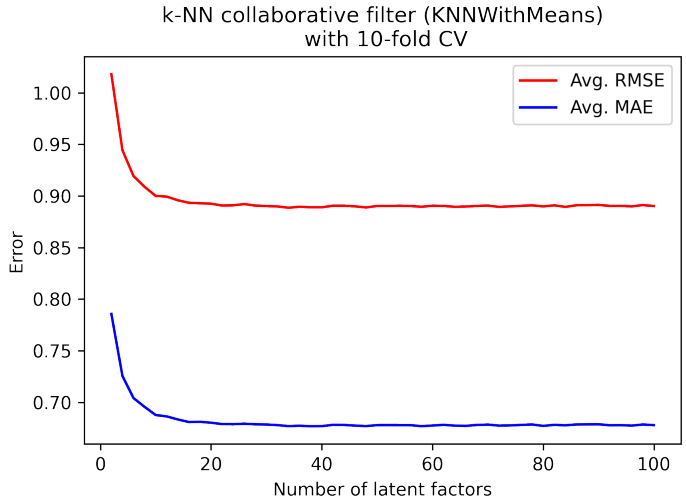

Figure 5: k-NN collaborative filter (KNNWithMeans) with 10-fold cross validation

Question 11: Finding minimum k

The minimum k is about 20 since both RMSE and MAE reach their respective steady state values at $k = 20$.
Steady-state RMSE = 0.8884
Steady-state MAE = 0.6781

4.6 Filter performance on trimmed test set

Question 12: k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set

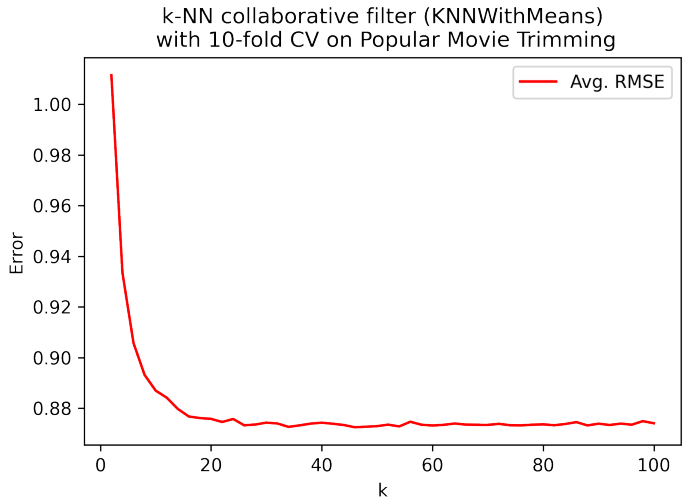

Figure 6: k-NN collaborative filter (`KNNWithMeans`) with 10-fold cross validation on popular movie trimmed test set

Minimum average RMSE for popular movie trimmed test set = 0.8725232799090584

Question 13: k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set

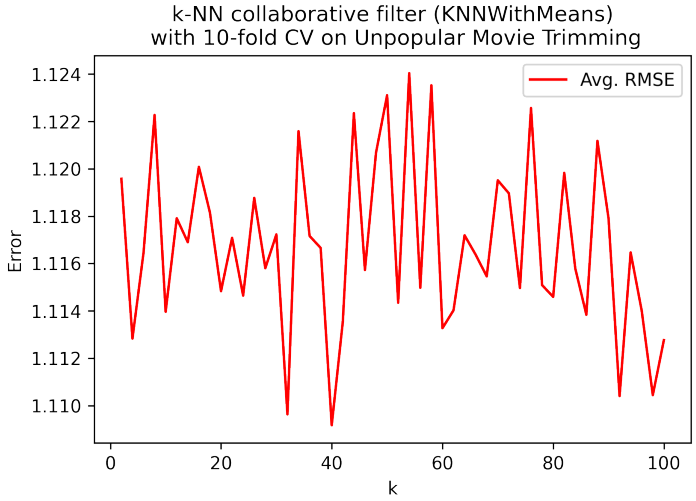

Figure 7: k-NN collaborative filter (`KNNWithMeans`) with 10-fold cross validation on unpopular movie trimmed test set

Minimum average RMSE for unpopular movie trimmed test set = 1.1091637833144685

Question 14: k-NN collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set

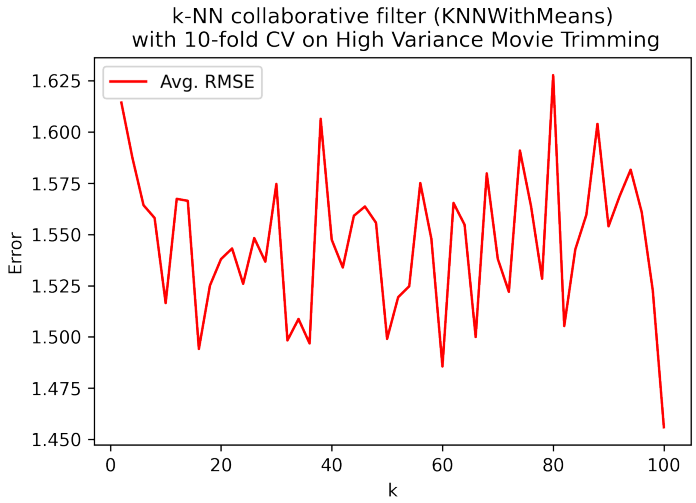

Figure 8: k-NN collaborative filter (`KNNWithMeans`) with 10-fold cross validation on high-variance movie trimmed test set

Minimum average RMSE for high-variance movie trimmed test set = 1.455878783333729

Question 15: Performance evaluation using ROC curve for the k-NN collaborative filter

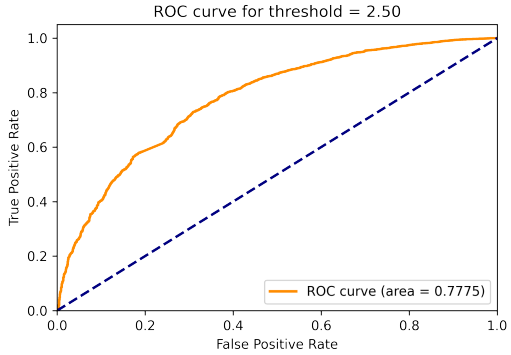

Figure 9: ROC curve for threshold = 2.5

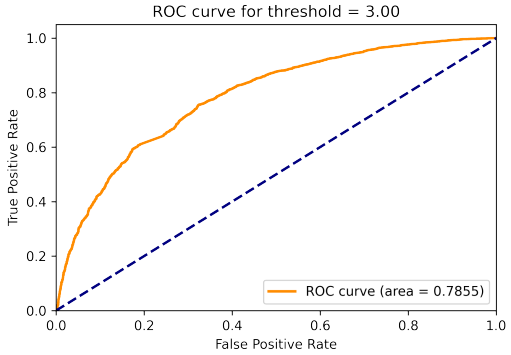

Figure 10: ROC curve for threshold = 3

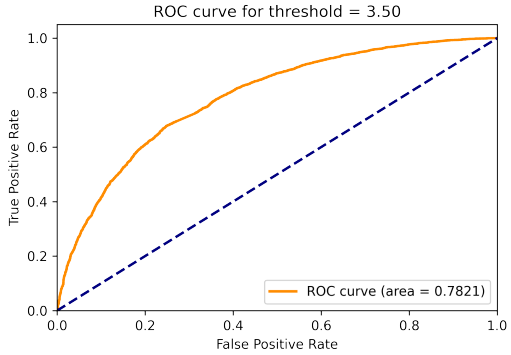

Figure 11: ROC curve for threshold = 3.5

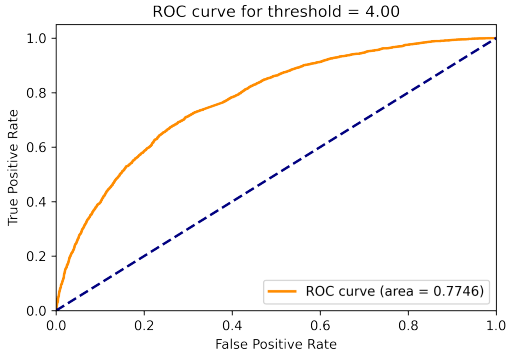

Figure 12: ROC curve for threshold = 4

5 Model-based collaborative filtering

5.2 Non-negative matrix factorization (NNMF)

Question 16: NNMF optimization formulation

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

This function is not convex. Let's analyze the simple scenario where the number of latent factors $k = 1$, and choose some i, j where r_{ij} is known. The function simplifies to:

$$f(u, v) = r - 2ruv + u^2v^2$$

The Hessian of this function is:

$$H(f) = \begin{bmatrix} 2v & 4uv - 2r \\ 4uv - 2r & 2u \end{bmatrix}$$

which is not positive semidefinite. Therefore, $f(u, v)$ is not convex, and hence neither is the optimization problem above.

For fixed U, we formulate the optimization problem into a least-squares problem as follows:

$$\begin{aligned}
&\underset{V}{\text{minimize}} \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T))_{ij})^2 \\
&\underset{V}{\text{minimize}} \sum_{i=1}^{m}\sum_{j=1}^{n} (W_{ij}r_{ij} - W_{ij}(UV^T))_{ij})^2 \\
&\underset{V}{\text{minimize}} \; ||W \circ R - W \circ (UV^T)||_{\mathcal{F}}^2 \\
&\underset{V}{\text{minimize}} \; ||B - W \circ (UV^T)||_{\mathcal{F}}^2
\end{aligned}$$

where we can move the weight matrix W into the summation because the weights are binary i.e. $W_{ij}^2 = W_{ij} \; \forall \, i, j$, and we can define $B = W \circ R$. In order to simplify $W \circ (UV^T)$, we observe the following:

$$\begin{aligned}
U &= \begin{bmatrix} — \; u_1^T \; — \\ \vdots \\ — \; u_m^T \; — \end{bmatrix} \text{ where } u_i \in \mathbb{R}^{k \times 1} \; \forall i \\
V^T &= \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix} \text{ where } v_i \in \mathbb{R}^{k \times 1} \; \forall i \\
W \circ UV^T &= \begin{bmatrix} W_{11}u_1^T v_1 & \cdots & W_{1n}u_1^T v_n \\ \vdots & & \vdots \\ W_{m1}u_m^T v_1 & \cdots & W_{mn}u_m^T v_n \end{bmatrix} \\
&= \begin{bmatrix} \begin{pmatrix} W_{11}u_1^T \\ \vdots \\ W_{m1}u_m^T \end{pmatrix} v_1 & \cdots & \begin{pmatrix} W_{1n}u_1^T \\ \vdots \\ W_{mn}u_m^T \end{pmatrix} v_n \end{bmatrix} \\
&= \begin{bmatrix} | & & | \\ A_1 v_1 & \cdots & A_n v_n \\ | & & | \end{bmatrix} \text{ where } A_i \in \mathbb{R}^{m \times k} \; \forall i
\end{aligned}$$

Now, we redefine B and V^T to be a vectorized version of themselves, $\tilde{B}$ and $\tilde{V}$:

$$\tilde{B} = \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} \in \mathbb{R}^{mn \times 1}, \;\; \tilde{V} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \in \mathbb{R}^{kn \times 1}.$$

From here, we can see that the term $W \circ (UV^T)$ simplifies to $\tilde{A}\tilde{V}$, where:

$$\tilde{A} = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & A_n \end{bmatrix} \in \mathbb{R}^{mn \times kn}$$

Therefore, the optimization problem reduces to:

$$\underset{\tilde{V}}{\text{minimize}} \; ||\tilde{B} - \tilde{A}\tilde{V}||_2^2$$

Question 17: NNMF collaborative filter to predict the ratings of the movies in the MovieLens dataset

The NNMF collaborative filter has been implemented using `NMF` from `surprise` Python package.

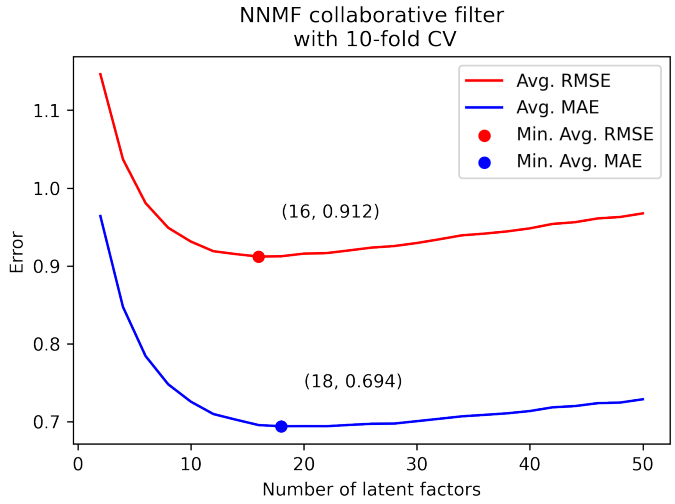

Figure 13: NNMF collaborative filter with 10-fold cross validation

Question 18: Optimal number of latent factors

Minimum RMSE = 0.9121067626592827 achieved at $k = 16$
Minimum MAE = 0.6943231334307327 achieved at $k = 18$

The optimal number of latent factors that minimizes the MAE ($k = 18$) is the same as the number of movie genres.

Question 19: NNMF collaborative filter to predict the ratings of the movies in the popular movie trimmed test set

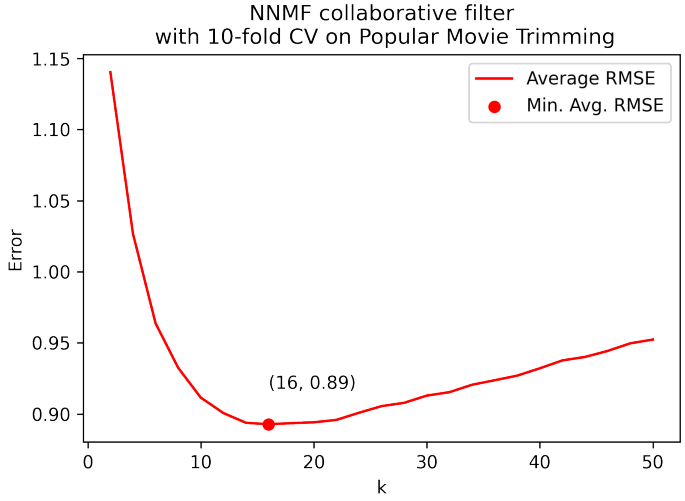

Figure 14: NNMF collaborative filter with 10-fold cross validation on popular movie trimmed test set

Minimum average RMSE for popular movie trimmed test set = 0.8927949119479364

Question 20: NNMF collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set

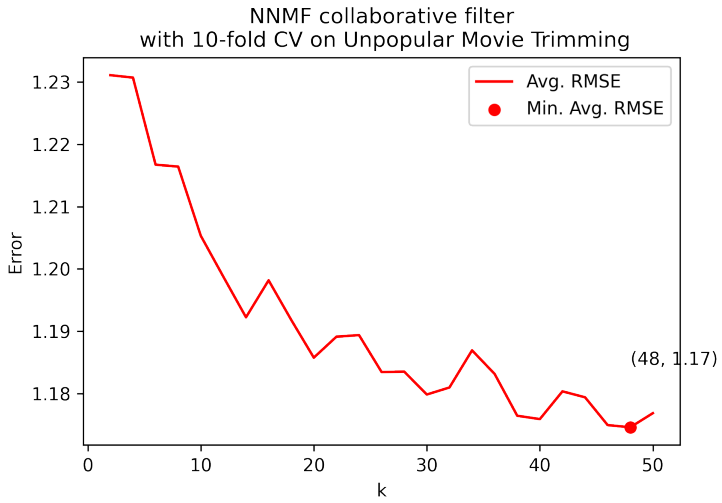

Figure 15: NNMF collaborative filter with 10-fold cross validation on unpopular movie trimmed test set

Minimum average RMSE for unpopular movie trimmed test set = 1.1746053689039906

Question 21: NNMF collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set

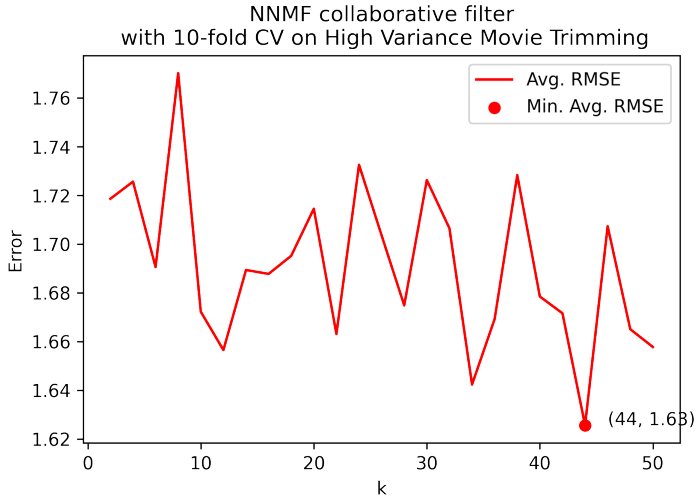

Figure 16: NNMF collaborative filter with 10-fold cross validation on high-variance movie trimmed test set

Minimum average RMSE for high-variance movie trimmed test set = 1.6256578822924233

Question 22: Performance evaluation using ROC curve for the NNMF collaborative filter

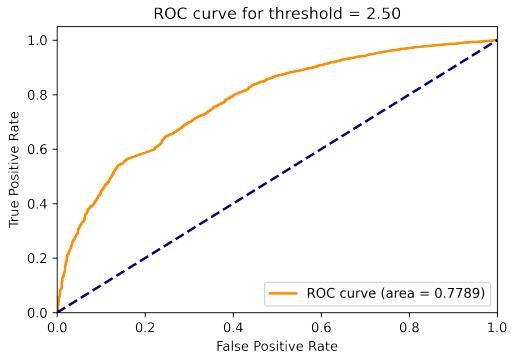

Figure 17: ROC curve for threshold = 2.5

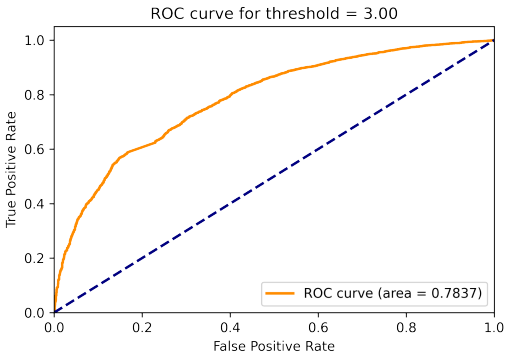

Figure 18: ROC curve for threshold = 3

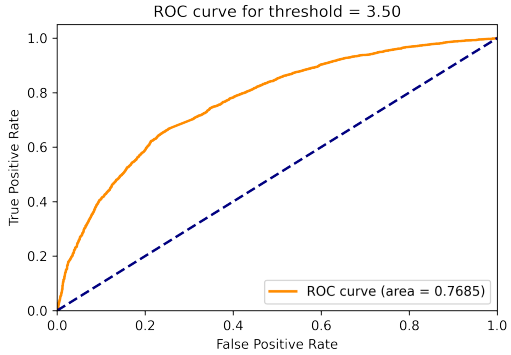

Figure 19: ROC curve for threshold = 3.5

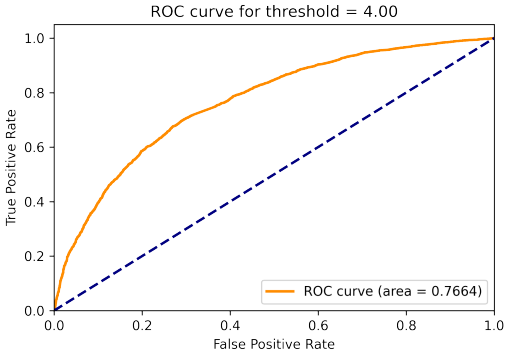

Figure 20: ROC curve for threshold = 4

5.3 Question 23: Interpretability of NNMF

There does appear to be a connection between latent factors and the movie genres, although it is quite noisy. The top 10 movies for a latent factor correspond to a small collection of genres as opposed to one genre outright.

k=1: The top 10 movies of this latent factor primarily correspond to the Action, Drama & Thriller genres.

k=4: The top 10 movies of this latent factor primarily corresponds to the Adventure & Drama genre.

k=6: The top 10 movies of this latent factor primarily correspond to Comedy, Drama & Romance genres.

k=7: The top 10 movies of this latent factor primarily correspond to Action genres.

k=1	k=4
Mystery|Thriller	Drama
Action|Adventure|Thriller|War	Adventure|Drama|Western
Action|Adventure|Comedy|Romance|Thriller	Comedy|Fantasy|Romance
Drama	Adventure|Fantasy|Romance
Comedy|Drama	Adventure|Comedy
Action|Adventure|Drama|Sci-Fi|Thriller	Drama|Romance
Action|Drama	Adventure|Animation|Children
Action|Comedy|Crime|Thriller	Crime|Drama
Thriller	Action|Adventure
Action|Adventure	Comedy|Drama

k=6	k=7
Comedy|Crime|Drama|Thriller	Animation|Children|Comedy|IMAX
Action|Sci-Fi|Thriller|IMAX	Action|Thriller
Fantasy|Horror|Romance|Thriller	Documentary
Comedy|Romance	Comedy|War
Comedy	Western
Romance|Thriller	Action|Adventure|Thriller
Comedy	Action|Adventure|Fantasy|Sci-Fi
Comedy|Drama|Musical|Sci-Fi	Action|Adventure|Sci-Fi|Thriller
Comedy|Drama	Action|Adventure|Animation|Children|Comedy|IMAX
Comedy|Drama	Comedy|Fantasy|Romance

5.4 Matrix factorization with bias (MF with bias)

Question 24: MF with bias collaborative filter to predict the ratings of the movies in the MovieLens dataset

The MF with bias collaborative filter has been implemented using `SVD` from `surprise` Python package. The parameter `bias` was set to `True`.

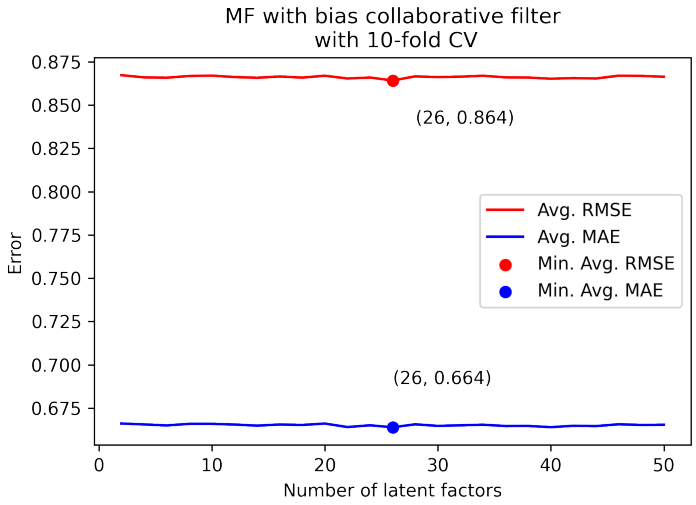

Figure 21: MF with bias collaborative filter with 10-fold cross validation

Question 25: Optimal number of latent factors

Minimum RMSE = 0.8642425340771457 achieved at $k = 26$
Minimum MAE = 0.6638737882629251 achieved at $k = 26$

Question 26: MF with bias collaborative filter to predict the ratings of the movies in the popular movie trimmed test set

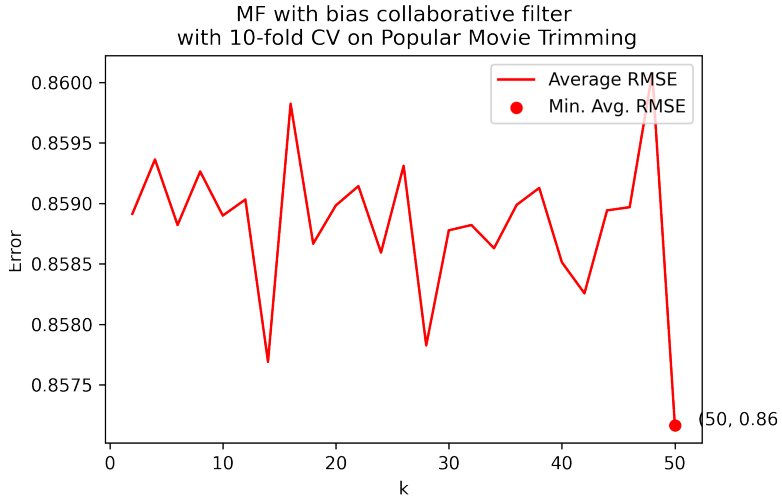

Figure 22: MF with bias collaborative filter with 10-fold cross validation on popular movie trimmed test set

Minimum average RMSE for popular movie trimmed test set = 0.8571661598901098

Question 27: MF with bias collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set

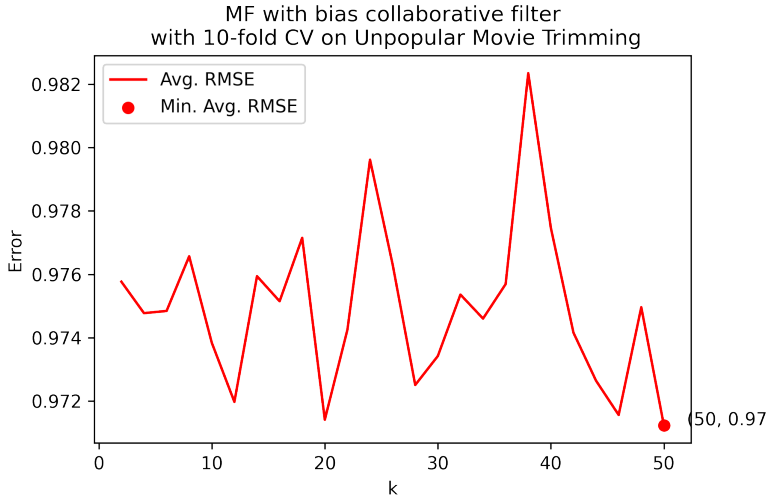

Figure 23: MF with collaborative filter with 10-fold cross validation on unpopular movie trimmed test set

Minimum average RMSE for unpopular movie trimmed test set = 0.9712392451845513

Question 28: MF with bias collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set

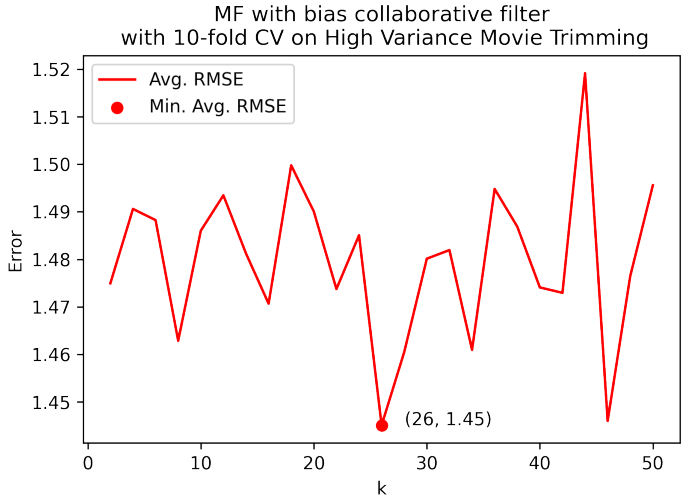

Figure 24: MF with collaborative filter with 10-fold cross validation on high-variance movie trimmed test set

Minimum average RMSE for high-variance movie trimmed test set = 1.4450257650369753

Question 29: Performance evaluation using ROC curve for the MF with bias collaborative filter

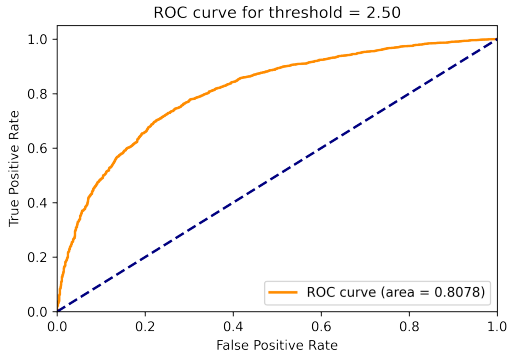

Figure 25: ROC curve for threshold = 2.5

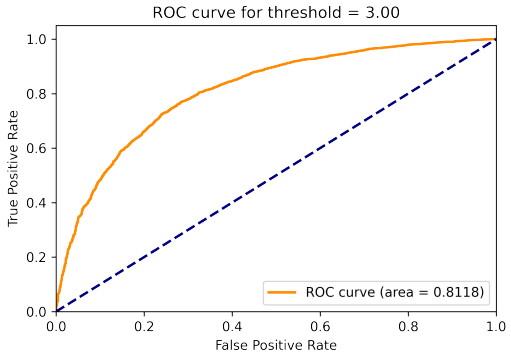

Figure 26: ROC curve for threshold = 3

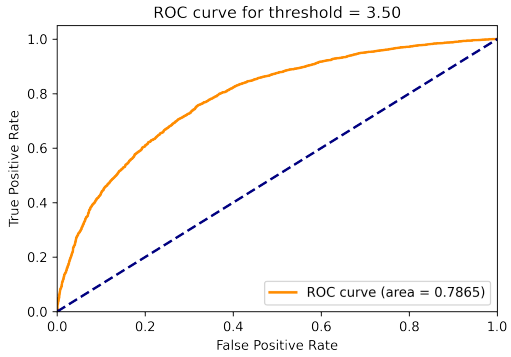

Figure 27: ROC curve for threshold = 3.5

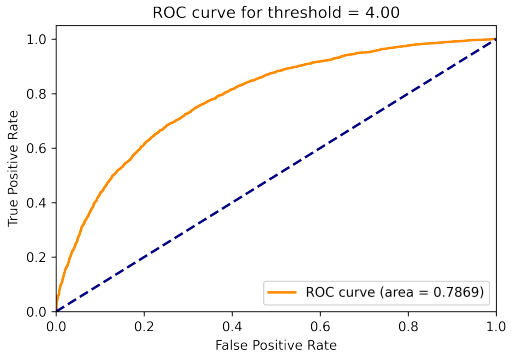

Figure 28: ROC curve for threshold = 4

6 Naive collaborative filtering

6.3 Design and test via cross-validation

Question 30: Naive collaborative filter to predict the ratings of the movies in the MovieLens dataset

Applying the naive collaborative filter which takes the average movie rating per user as the predicted value, the average RMSE for naive collaborative filter: 0.9346686342108225

6.4 Naive collaborative filter performance on trimmed test set

Question 31: Naive collaborative filter to predict the ratings of the movies in the popular movie trimmed test set

On the popular movie trimmed test set, the average RMSE for naive collaborative filter is: 0.9323164164821562

Question 32: Naive collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set

On the unpopular movie trimmed test set, the average RMSE for naive collaborative filter is: 0.9712322165894877

Question 33: Naive collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set

On the high variance movie trimmed test set, the average RMSE for naive collaborative filter is: 1.4786634431139551

7 Performance comparison

Question 34: ROC curves for the k-NN, NNMF, and MF with bias based collaborative filters

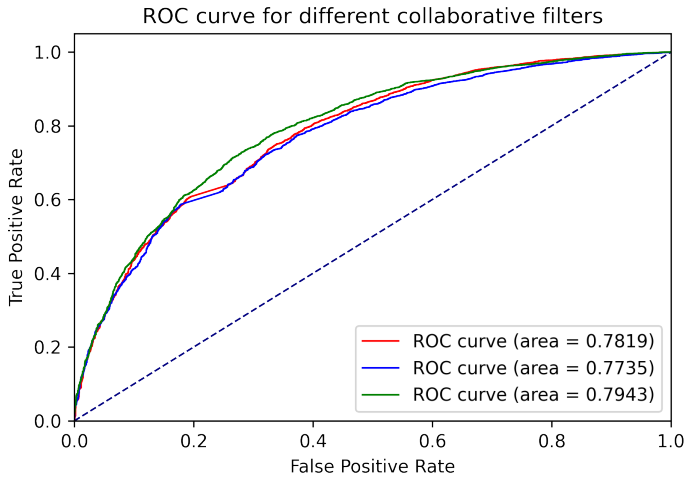

Figure 29: ROC curves for k-NN, NNMF and MF with bias based collaborative filters

It can be seen from the figure that MF with bias based collaborative filter performs better than k-NN and NNMF. The red line represents the k-NN collaborative filter, blue for NNMF, and green for MF with bias. The three are all good classifiers. MF with bias reaches a larger area under ROC curve which represents a higher true positive rate. k-NN performs better than NNMF.

8 Ranking

Question 35: Explain the meaning of precision and recall

Precision is the ratio of a good prediction among the set of recommendation. Recall is how many items in the positive ratings are correctly addressed. Precision reflects how accurate the model is, and recall shows the completeness that positive items are retrieved.

Question 36: Precision-recall curves for k-NN collaborative filter

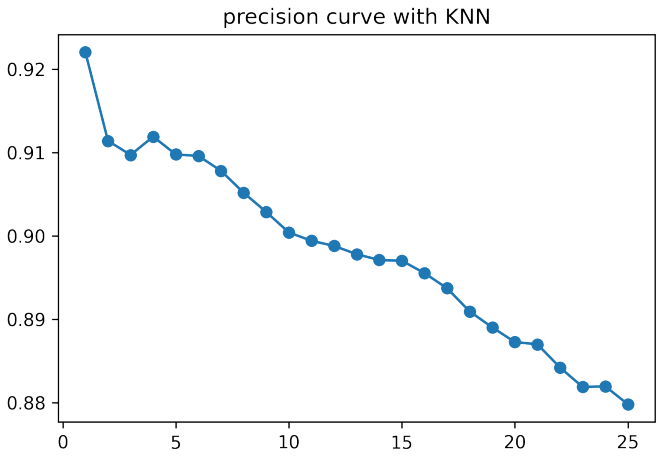

Figure 30: KNN Precision Curve

As the set of recommendation items increases, the precision score drops. As more items get predicted, the error increases.

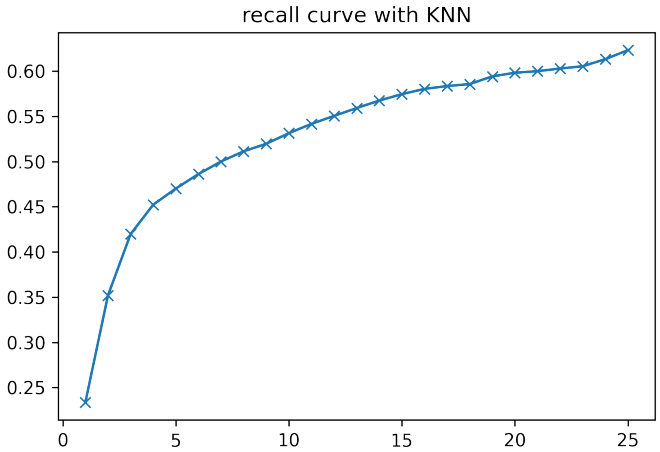

Figure 31: KNN Recall Curve

The recall value increases with the size of recommendation item. As more items get recommended, higher the probability the positive rating items get retrieved.

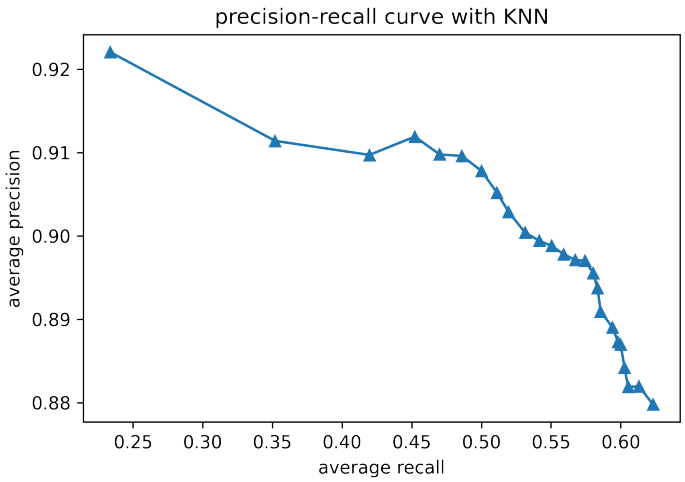

Figure 32: KNN Precision-Recall Curve

The precision-recall curve slopes downwards, with a rapid drop after average recall of 0.5. When more positive items are predicted, it requires a larger base sample, which occurs more false positive predictions and the precision performance gets worse.

Question 37: Precision-recall curves for NNMF-based collaborative filter

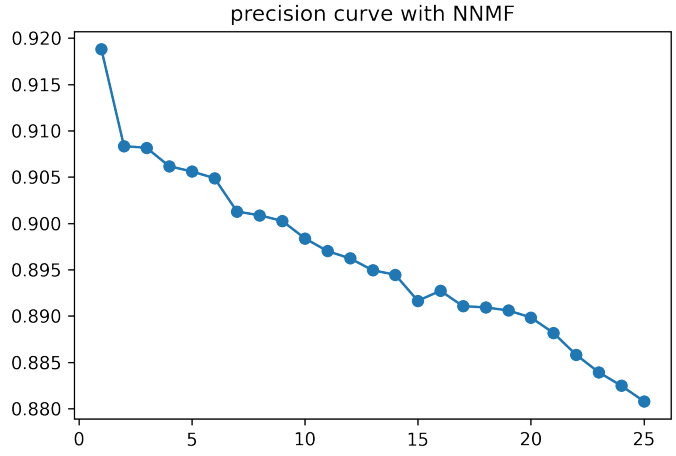

Figure 33: NNMF Precision Curve

The precision drops at an even speed as the size of set increases.

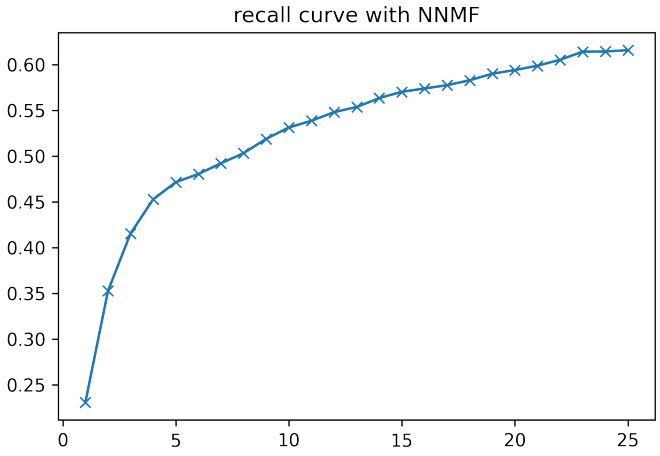

Figure 34: NNMF Recall Curve

The recall score increases rapidly before size of 5, then slowly increases after, it tends to reach a balanced status after size of 23.

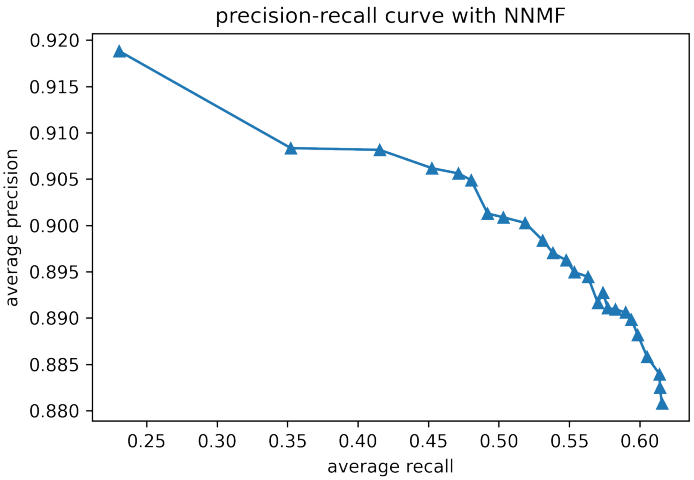

Figure 35: NNMF Precision-Recall Curve

The precision-recall curve is a downward shape, and drops more rapidly after an average recall value of 0.5.

Question 38: Precision-recall curves for MF with bias-based collaborative filter

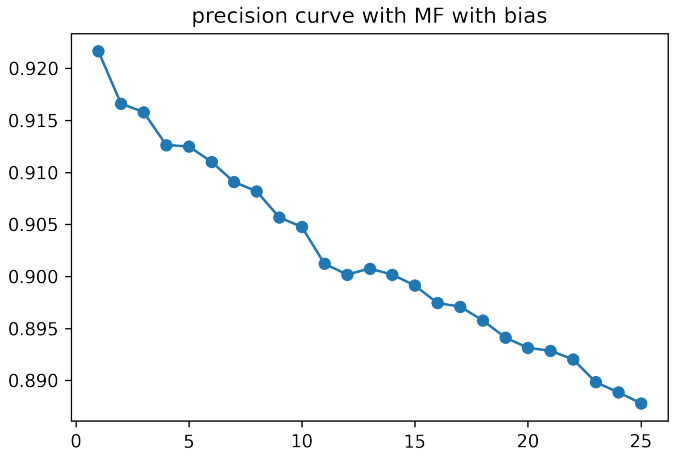

Figure 36: MF with bias Precision Curve

The precision value drops from 0.92 to 0.8 as size t increases from 1 to 25.

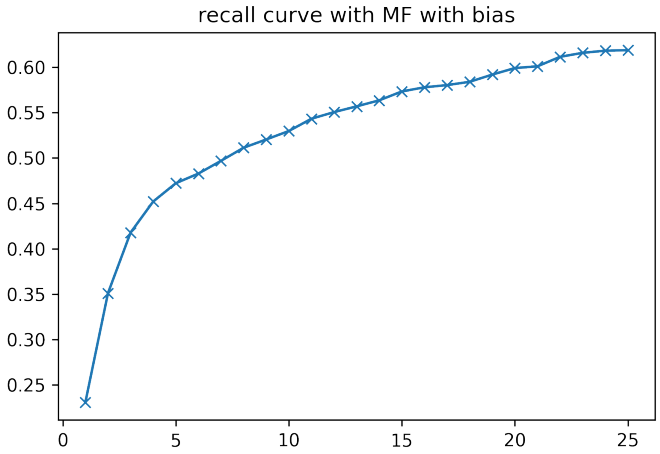

Figure 37: MF with bias Recall Curve

The recall value increases as t increases, and the slope is slower-rising after t of 5.

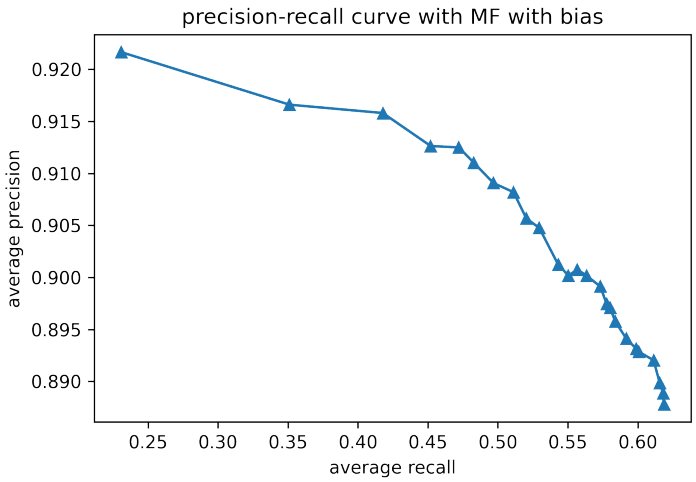

Figure 38: MF with bias Precision-Recall Curve

The precision value decreases as recall value increases, and drops faster when the average recall value reaches between 0.45 and 0.5

Question 39: Precision-recall curves of k-NN, NNMF, and MF with bias collaborative filters

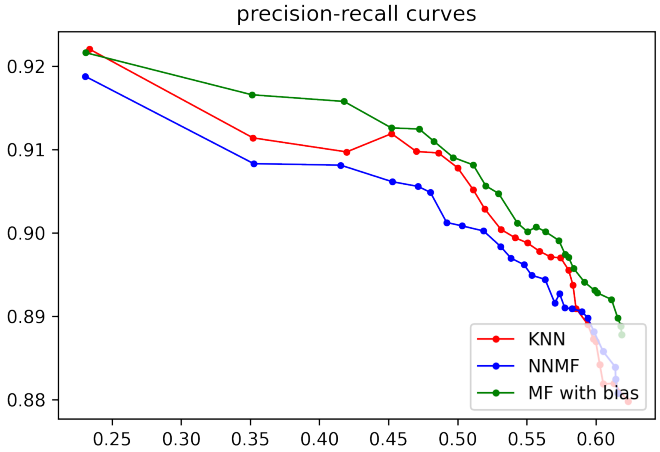

Figure 39: Precision-Recall Curve of the collaborative filters

From the figure, MF with bias based collaborative filter has the best prediction. k-NN is better than NNMF. With the same recall value, MF with bias achives the highest precision score, which indicates that it has better accuracy at the recommendations. For the same precision value, it has higher recall score which shows that it retrieves the positive rating movies more completely.