***EE-219 Project 3***

Project 3: Collaborative Filtering

***February 22, 2018***

***Ashish Shah (804946005)***

***Ayush Dattagupta (305024749)***

***Shrey Agarwal (004943082)***

***Varun Saboo (505028591)***

**Objective –**

In this project, we will build a recommendation system to predict the ratings of the movies in the MovieLens dataset. The concept behind a recommendation system is that we use the user data to interpret customer interests. Recommendation systems are widely used today by ecommerce companies such as Amazon or video streaming services such as Netflix. Based on users trend, we can suggest other users a recommendation as well.

**Dataset –**

We work with MovieLens dataset which is a collection of movies and users along with their ratings and timestamp. Each movie belongs to certain genres which we are not considering in this project.

Question 1:

**Compute Sparsity:**

The sparsity of the data set is given by the following equation.

Density=

In our data set, the following information is:

1. Number of unique users: 671
2. Number of unique movies: 9066
3. Number of available ratings: 100004

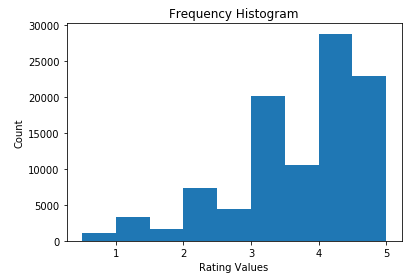
Sparsity is: 1 - Density = 0.9835608583913366

Observation:

For calculating the sparsity of our ratings data, we use the surprise() python package to import data and find the statistics above. From these stats were were able to calculate the sparsity. From our results, we can see that the given ratings data is indeed very sparse.

Question 2:

**Plot the histogram showing frequency of the rating values:**

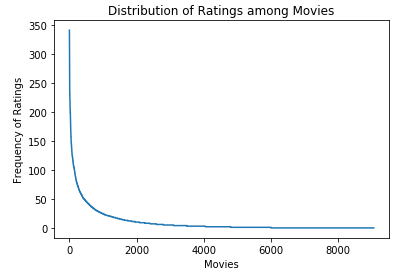


*Figure 1: Histogram of frequency*

Observation:

We have used the surprise() package to extract the ratings column. Once that was extracted, we binned the values in a range from 1-5 to produce the histogram in Figure 1. From the histogram, we can infer that majority of the movies have got higher ratings. Most of the users who rated the movies gave a higher rating to the movies. Hence the histogram is skewed to the right.

Question 3:

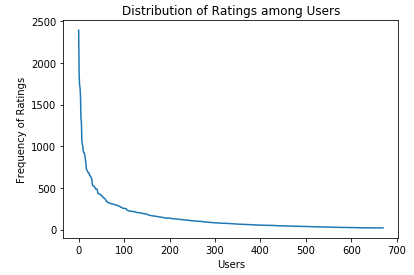
**Plot the distribution of ratings among movies:**

*Figure 2: Distribution of ratings versus movies*

As we can observe, the number of movies that receive ratings decrease exponentially. That is, there are very few movies that have high number of ratings. We see that movies with the most number of ratings received around 350 ratings.

Question 4:

**Plot distribution of ratings among users:**



*Figure 3: Variance in Rating Values for each movie*

From the graph, we see that users who rated the most movies, rated around 2500 movies.

Question 5:

**Explain salient features of distribution in Question 3:**

The number of movies are: 9066.

Out of which, only 3099 movies have >5 ratings in total.

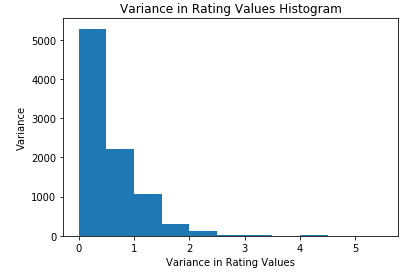
Total of 3063 movies have only one rating.

The above statistics along with the Sparsity of 0.98 tells us that the data is very skewed and hence the need for a method to predict ratings arises. The movie with maximum ratings is 350 ratings.

The recommendation systems will recommend popular movies more so we should use rating to reduce the effect of the probability of recommending a popular movie. We want to give equal probability to each movie when recommending them. Do not take the popular movies as a similarity measure since they are popular. In recommendation system, similarity measures are calculated between users based on eatings and the when many users rate movies which are popular movies, they should have more weight. One such solution is to take the ratings given by user and see how they are similar to other people. If the movie is popular, it will have a larger weight. Then we can apply normalization techniques by dividing the weight by sum of similarities of all people who rated the movie. We can avoid taking this popular movies as a similarity measure since they are popular.

Question 6:

**Compute Variance of rating values received by each movie:**



*Figure 4: Variance in Rating Values for each movie*

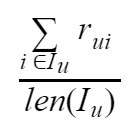
The above histogram shows us that the ratings received by all users for each movie is in between [0, 1] majorly and there are very few movies with high variance (> 1). The lesser the variance in ratings, the precise the prediction of ratings will be.

For this part, we created a dictionary for all the movie IDs and its ratings per movie. We then calculated the variances of each list and plotted the results in the histogram. We can see that for any movie, the ratings do not vary by a lot. This tells us that people tend to pick a similar rating for a movie. Hence the graph is skewed towards the left.

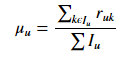
Question 7:

**Write formula for µu in terms of Iu and ruk**

The formula for µ\_u is the sum of all the r\_ui’s in the set I\_u divided by the total number of elements in I\_u:



OR



Question 8:

**In plain words, explain the meaning of Iu ∩ Iv. Can Iu ∩ Iv = Ø**

Iu ∩ Iv: This means we are trying to find the set of items that both the users u and v have rated.

Can Iu ∩ Iv = Ø: Yes, it can!

Reason:  
Iu ∩ Iv = Ø: If the set of items that both the users have rated is empty, it means that the users u and v have no items common to each other. This also indicates that the users u and v are very unlikely to be neighbours.

It tell us that the set of indices for which both user u and user v have given ratings. From above we calculated that the data is sparse, hence it is likely that there is no interaction between the 2 sets.

Question 9

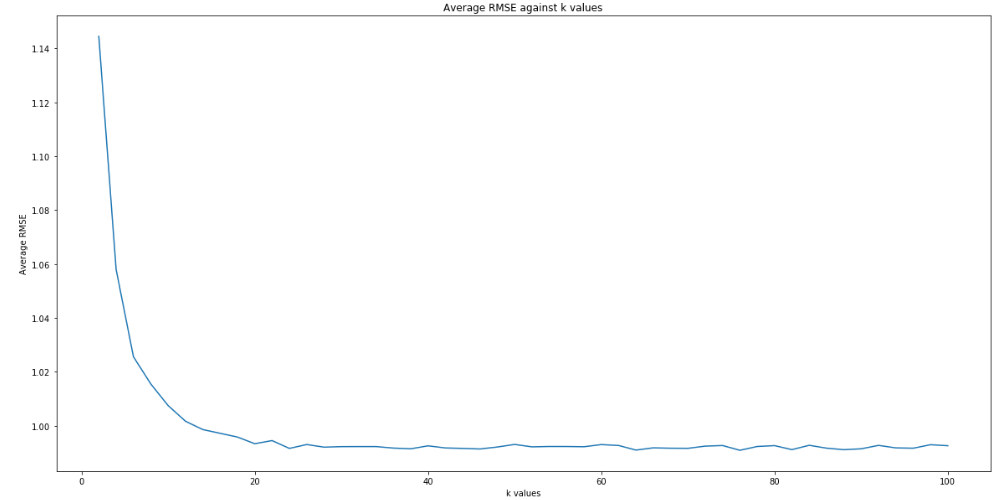
**Can you explain the reason behind mean-centering the raw ratings (rvj − µv) in the prediction function?**

Different users rate movies on different scales. For example, one user might rate all movies with high values whereas another user might rate all movies badly. Hence, the raw ratings of each user needs to be mean centered before determining average ratings for the peer group.

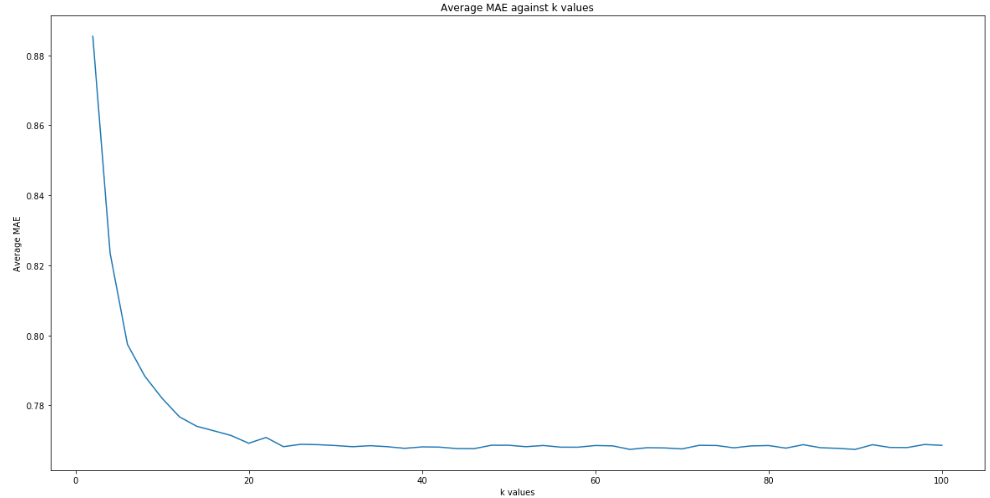
The logic behind collaborative filtering is to identify the likes and dislikes of a particular user and then find the people who have similar taste as that user. If that is the case then there is a high probability that movies liked by this user might also be like by the other users. The way we can find similar people is by using this concept of euclidean distances in some high dimensional data space or by finding the similarity in rating of movies they might have in common. Mean centered rating help in this similarity determination. The disadvantage with naive euclidean and similarity methods is that people are different by nature and while two different people may have given different ratings to the same movie they might like it equally. eg: One user is a strict rater and gives ratings of no more than 4 for their favorite movies. Another user on the other hand gives 5 to their loved movies and gives a minimum of 3 to movies they hate. Both users may have similar taste but due to difference in raw ratings, similarity measures may not be of any use. If we instead mean center these ratings, while the users may be working on different ‘scales’ the mean centered correlation of rating for both the users would be identified by similarity measures. Therefore using mean centered ratings gives a better method to identify similarities between users and improves the performance of the model.

Question 10:

**Design a k-NN collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross validation. Plot average RMSE (Y-axis) against k (X-axis) and average MAE (Y-axis) against k (X-axis).**



*Figure 5: Variance in Rating Values for each movie*



*Figure 6: Variance in Rating Values for each movie*

We can observe that the Average MAE and RMSE are converging to a steady state after certain k value which we will determine in the next section. Steady state is reached when the average MAE and RMSE don’t change significantly with increase in k value.

Here we first swept k from 2 to 100 in steps of 2 and for each of them calculated RMSE and MSE across the 10 folds. Then we plot the respective graphs.

Question 11:

**Find minimum k:**

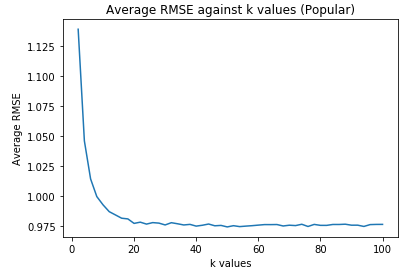
From the graph in Figure 5, we can see that the minimum k is around k = 22

From k = 22, we see a plateau in both the RMSE and the MAE values.

Average RMSE for minimum k (22) is: 0.991639416994 Average MAE for minimum k (22) is: 0.768208330664

Question 12:

**Design a k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set 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 obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.**



*Figure 7: Rmse against K for popular trimmed dataset*

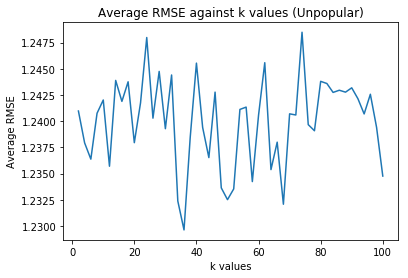
Minimum Average RMSE is: 0.9743907478134

The KNN model was executed on the popular trimmed movie set. The popular movie set is defined as the list of all movies in the dataset that have more than two ratings. Essentially we do not predict the rating for those movies for which there were less than or equal to two ratings in the whole dataset. According to intuition the rmse should decrease a bit for for such a dataset as for the movies that have not been rated (or rated less) cannot be trained as well by the model and therefore the predictions for such movies are going to be more accurate than for those movies which have been trained extensively by the model.

The results confirm this hypothesis and the pattern of RMSE against k is the same as that for the full dataset but the minimum average RMSE is slightly lower.

Question 13

**Design a k-NN collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set 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 obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.**



*Figure 8: Rmse against K for unpopular trimmed dataset*

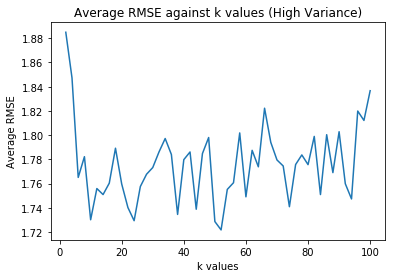
Minimum Average RMSE is: 1.229630221149743

For this model prediction were only made on unpopular movie. i.e those movies which have 2 or fewer ratings in the whole database. The hypothesis here is that for such less frequent movies it would be difficult to study or learn any good pattern from the dataset and there is a high probability of examples that were never a part of the training set during fitting but are a part of the test set. This implies that we expect a poor performance in terms of RMSE and there would be no real correlation between k values and the model performance (as irrespective of k the contribution of these unpopular movies to the model in minimal).

The results confirm this hypothesis and it can be seen that the rmse for the unpopular movies dataset is much higher than the popular movies dataset. There is als no clear correlation between the k values and the rmse.

Question 14

**Design a k-NN collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set 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 obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE**



*Figure 9: Rmse against K for high variance trimmed dataset*

Minimum Average RMSE is: 1.7217094067203256

For this model while we predict the ratings for movies with more than 5 ratings in the dataset we only predict the ratings for movies with high variance, i.e. movies which have a very high deviation of ratings from the mean which in simpler terms implies that most people either love the movie or hate the movie (as variance is greater than 2 on a rating scale of 5) It is difficult to predict the ratings of such movies as there is an extreme opinion on these movies, and as shown by the results the rmse is high for this dataset with no definite correlation with the k values.

Question 15

**Plot the ROC curves for the k-NN collaborative lter designed in question 10 for threshold values [2.5; 3; 3.5; 4]. For the ROC plotting use the k found in question 11. For each of the plots, also report the area under the curve (AUC) value.**

Using k=22 from question 11, we get the following plots:

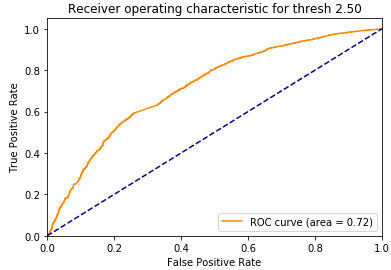


Fig 10: ROC curve for 2.5

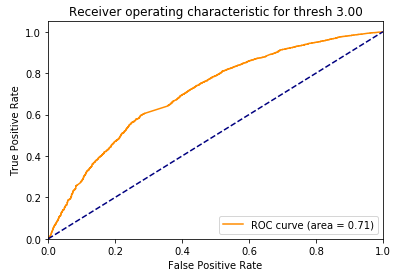


Fig 11: ROC curve for 3.0

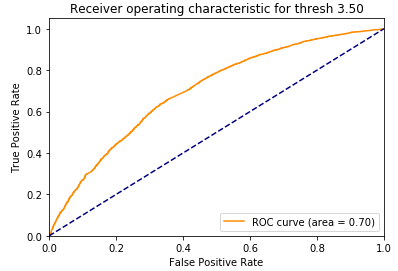


Fig 12: ROC curve for 3.5

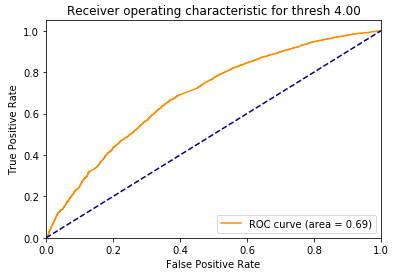


Fig 13: ROC curve for 4

threshold= 2.5 : 0.72

threshold= 3 : 0.71

threshold= 3.5 : 0.70

threshold= 4 : 0.69

Hence we can conclude that the best result is with threshold = 3

Question 16

**Is the optimization problem given by equation 5 convex? Consider the optimization problem given by equation 5. For U fixed, formulate it as a least-squares problem.**

The optimization problem given in equation 5 is non-convex.

A function f(x) is said to be convex if it satisfies the following property:

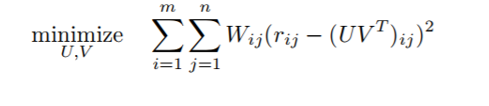
and the domain of f is a convex set.

A simple matrix factorization based model predicts ratings, using a product of the item and user latent factors. With U fixed, the criterion is convex in V and vice versa.

The objective function includes this term minimizing

And the above sort of function is non convex in nature.

This objective function



is bi-convex and fixing either U or V converges it to an alternating least squares problem.

On fixing U, our ALS equation becomes:

Where vj are v1, . . . , vn ∈ Rk -the factors for the items. (V is n x k movie matrix with n movies) and rj  will be the ratings.

Question 17

**Design a NNMF-based collaborative lter to predict the ratings of the movies in the MovieLens 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. 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.**

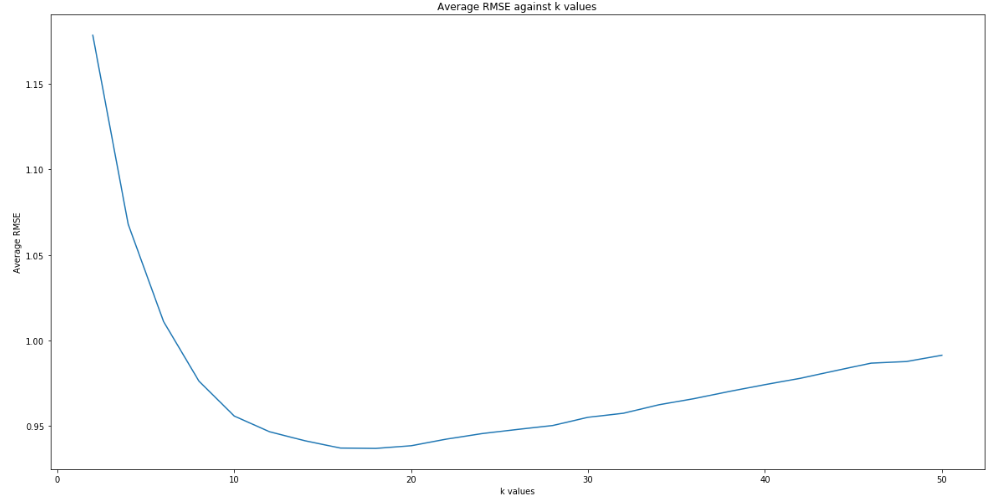


Fig 14: Graph showing Avg RMSE vs k values

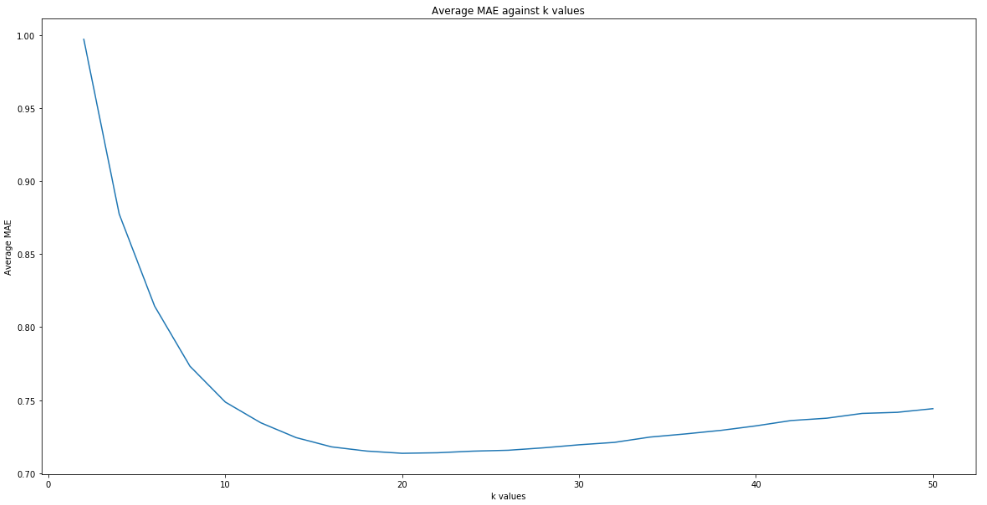


Fig 15: Graph showing Avg MAE vs k values

Question 18

**Use the plot from question 17, to nd 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?**

We see that the optimal number of latent factors is 18 to 20. Hence we consider optimal number of latent factors = 18. And the total movie genres are 20.

**Is the optimal number of latent factors same as the number of movie genres?**

No, the number of genres is 20 but the optimal number of latent factors for RMSE is 18.

**Please report the minimum average RMSE and MAE:**

Average RMSE = 0.938441920639

Average MAE = 0.713494787859

Question 19

**Design a NNMF collaborative filter to predict the ratings of the movies in the popular movie trimmed test set 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). Also, report the minimum average RMSE**

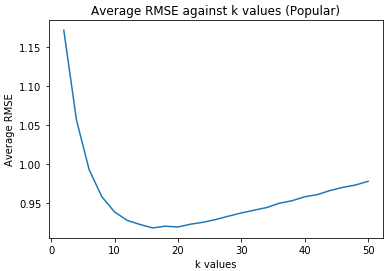


Fig 15: Graph showing Avg RMSE vs k values

Fig 12: Graph showing Avg RMSE vs k value (popular movies)

optimalk\_RMSE\_np = 16

the minimum average RMSE = 0.917925230022

The pattern observed is in accordance with popular movie hypothesis discussed during KNN. The average RMSE is slightly lower than the full dataset.

Question 20

**Design a NNMF collaborative filter to predict the ratings of the movies in the unpopular movie trimmed test set 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). Also, report the minimum average RMSE.**

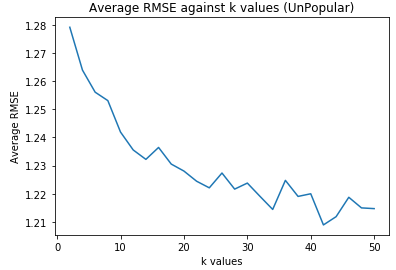


Fig 16: Graph showing Avg RMSE vs k value (Unpopular movies)

Minimum Average RMSE is: 1.20890667847

optimalk\_RMSE\_nu = 50

Observation: It can be observed from the graph that in case of unpopular movie trimming, RMSE stays between the fairly small range of [1.22,1.29] for k=2 to 50. Increasing k seems to decrease RMSE(not by much though) and the minimum average RMSE is higher in this case than the popular movie trimming case.

Question 21

**Design a NNMF collaborative filter to predict the ratings of the movies in the high variance movie trimmed test set 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 obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.**

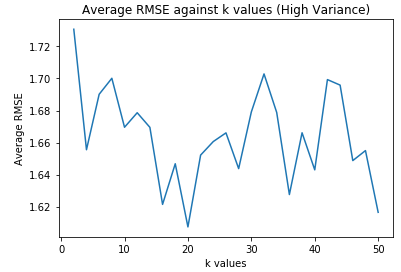


Fig 17: Graph showing Avg RMSE vs k value (High Variance movies)

optimalk\_RMSE\_nv = 20

minimum average RMSE = 1.60732413779

From the above graphs, we can conclude that the popular movies tend to converge better and don’t exhibit any erratic behaviour. Whereas, unpopular movies and High Variance movies show an unusual pattern and the graphs are random. This makes sense for high variance movies since their variance is so high that the model fails to predict the ratings. Unpopular movies have less ratings which makes it difficult to predict the ratings.

Question 22

**Plot the ROC curves for the NNMF-based collaborative filter designed in question 17 for threshold values [2:5; 3; 3:5; 4]. For the ROC plotting use the optimal number of latent factors found in question 18. For each of the plots, also report the area under the curve (AUC) value.**

Optimal k =18 for RMSE

Area under the curve (AUC) value for threshold = 2.5

0.750924287216

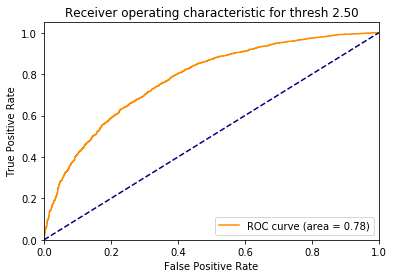


Fig 18: ROC curve for 2.5

Area under the curve (AUC) value for threshold = 3.0

0.759079554655

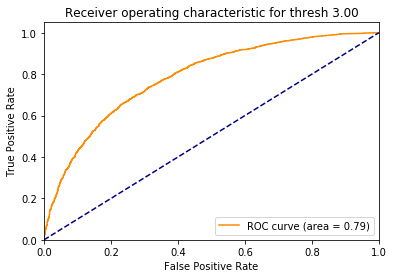


Fig 19: ROC curve for 3.0

Area under the curve (AUC) value for threshold = 3.5

0.75166809332

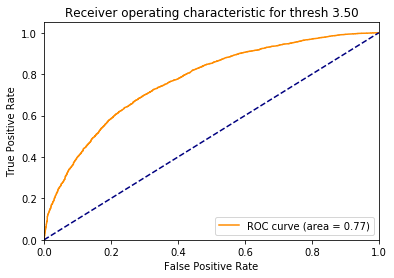


Fig 20: ROC curve for 3.5

Area under the curve (AUC) value for threshold = 4.0

0.755921951454

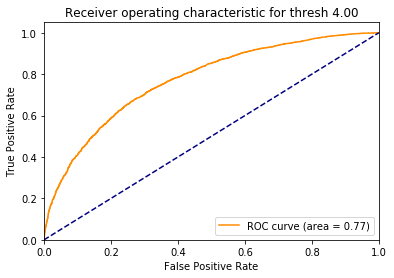


Fig 21: ROC curve for 4.0

From the above graphs we can conclude that the ROC curve is best for threshold value = 3. with the Area Under Curve = 0.79

Question 23

**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?**

Each column which represents latent factors of a movie have some correlation between their genres. For example column 7, drama is one if the dominating genres and it also well spread across the other movie latent factors. It is safe to say that these movie latent factors explain the hidden features of a movie which is why latent factor to genre mapping is a good strategy to identify the types of movies spread across different latent factors/columns.

k=20

column 0

Comedy|Drama|Romance

Children|Drama

Children|Comedy

Horror

Drama|Thriller|War

Comedy|Crime|Drama

Crime|Drama|Thriller

Adventure|Comedy|Thriller

Comedy|Musical

Comedy|Drama

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

column 1

Action|Drama|Fantasy

Action|Drama|War

Fantasy|Horror|Mystery|Thriller

Horror|Sci-Fi

Documentary|War

Action|Adventure|Sci-Fi

Comedy|Horror

Adventure|Animation|Children|Comedy|Fantasy

Adventure|Documentary

Drama

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

column 2

Comedy|Drama

Drama|Thriller

Drama|Romance

Drama|Mystery

Drama

Comedy|Crime|Drama

Adventure|Children|Comedy

Comedy

Comedy|Romance|Sci-Fi

Animation|Comedy|Musical

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

column 3

Documentary

Documentary|Drama

Comedy|Drama|Romance

Drama

Horror|Thriller

Documentary

Action|Comedy|Crime|Drama

Thriller

Crime|Drama|Thriller

Comedy|Crime

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

column 4

Horror|Thriller

Comedy|Fantasy|Romance

Action|Adventure|Sci-Fi|Thriller

Crime|Drama|Film-Noir

Action|Crime|Thriller

Action|Adventure|Comedy|Fantasy

Drama|Romance

Drama

Action|Adventure|Comedy

Thriller

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

column 5

Drama|Romance

Sci-Fi

Action|Crime|Thriller

Action|Adventure|Animation

Horror|Sci-Fi|Thriller

Crime|Drama

Documentary|War

Comedy|Romance

Drama

Comedy|Drama|Musical

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

column 6

Drama|War

Action|Crime|Thriller

Action|Drama|War

Action|Adventure|Fantasy|Horror

Horror

Comedy|Drama

Adventure|Drama|Western

Adventure|Drama|Sci-Fi

Comedy

Action|Comedy|Fantasy|Horror

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

column 7

Drama

Action|Drama|Fantasy

Comedy|Drama|Romance

Comedy|Romance

Drama

Drama|Thriller|War

Animation|Drama|Sci-Fi|IMAX

Drama|Thriller

Drama|Thriller

Drama|War

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

column 8

Comedy|Drama

Crime|Mystery|Thriller

Drama|War

Comedy|Crime|Mystery|Thriller

Drama|Mystery

Comedy|Sci-Fi

Comedy|Drama

Action|Crime|Fantasy

Drama

Action|Comedy|Crime|Drama

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

column 9

Comedy

Drama|Thriller

Musical

Comedy|Romance

Action|Crime|Romance|Thriller

Action|Adventure|Sci-Fi

Comedy|Fantasy

Drama

Comedy

Comedy

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

column 10

Horror

Drama

Film-Noir|Thriller

Romance|Thriller

Adventure|Western

Comedy

Western

Drama

Comedy|Romance

Drama|Horror|Thriller

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

column 11

Drama|Thriller

Drama

Comedy|Drama|Romance

Action|Adventure|Comedy

Action|Adventure

Drama

Fantasy|Sci-Fi

Drama|Horror

Action|Adventure|Comedy|Crime|Drama

Comedy|Documentary

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

column 12

Drama|Horror|Thriller

Action|Crime|Drama|Thriller

Comedy|Crime|Drama|Thriller

Drama

Action|Drama

Action|Drama

Drama

Drama|Horror|Mystery|Thriller

Thriller

Adventure|Children|Comedy

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

column 13

Comedy|Drama|Romance

Drama

Drama|Sci-Fi

Comedy

Drama|War

Comedy|Horror|Sci-Fi

Action|Thriller

Adventure|Children|Drama|Fantasy

Action|Adventure|Comedy|Thriller

Comedy

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

column 14

Documentary

Drama|Fantasy|Mystery|Romance

Horror

Comedy|Drama|Fantasy|Romance

Action|Comedy

Comedy|Romance

Action|Horror|Sci-Fi

Action

Comedy|Drama

Crime|Drama

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

column 15

Drama

Documentary|War

Adventure

Action|Drama

Action|Romance

Comedy

Comedy|Drama

Action|Drama|Thriller

Comedy

Comedy|Drama

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

column 16

Crime|Drama|Thriller

Action|Drama|War

Drama

Comedy

Comedy|Romance

Documentary

Comedy|Romance

Action|Comedy|Fantasy|IMAX

Adventure|Documentary

Documentary

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

column 17

Adventure|Drama|Sci-Fi

Action|Crime|Drama

Adventure|Comedy|Crime|Thriller

Comedy

Action|Comedy|Sci-Fi

Drama|War

Drama|Romance|Sci-Fi

Crime|Film-Noir|Thriller

Drama

Drama|Romance|War

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

column 18

Action|Adventure|Children|Fantasy

Crime|Mystery|Thriller

Action

Horror

Drama|Thriller

Action|Thriller

Horror|Thriller

Crime|Drama|Mystery|Romance|Thriller

Comedy|Romance

Action|Adventure|Sci-Fi|IMAX

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

column 19

Comedy

Adventure|Comedy|Fantasy|Romance

Documentary

Action|Crime|Horror

Drama|Romance

Drama|Fantasy|Mystery|Romance|Thriller

Mystery|Thriller

Animation|Horror|Mystery|Thriller

Drama|Film-Noir|Thriller

Comedy

Question 24

**Design a MF with bias collaborative lter to predict the ratings of the movies in the MovieLens 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.**

The following graph plots Avg RMSE vs k values.

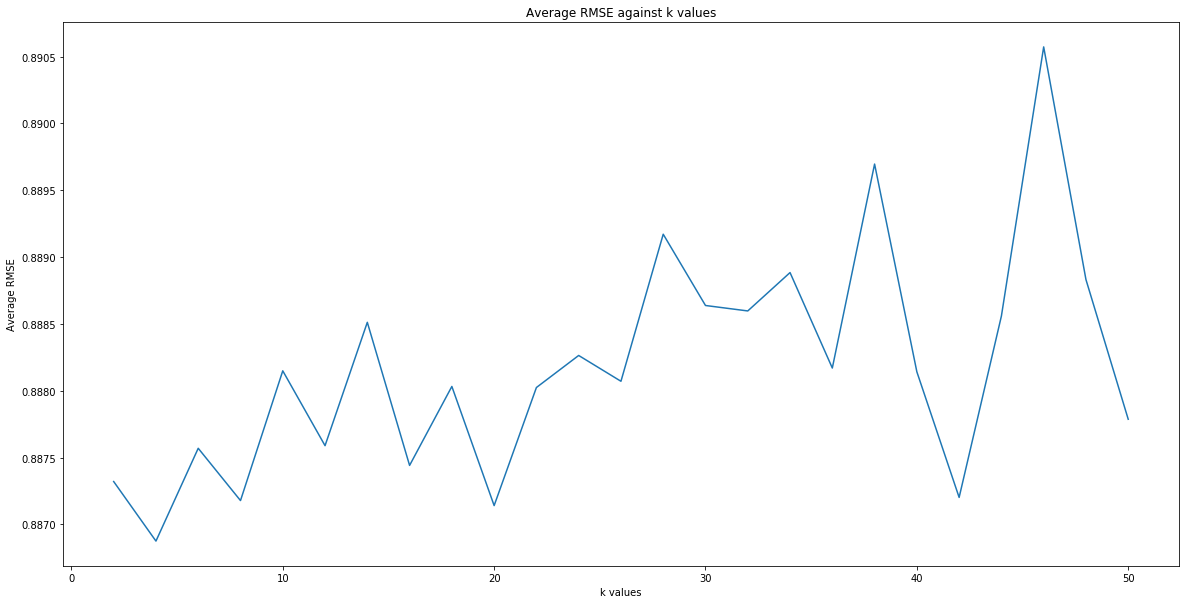


Fig 22: Graph showing Avg RMSE vs k values

The following graph plots Avg MAE vs k values.

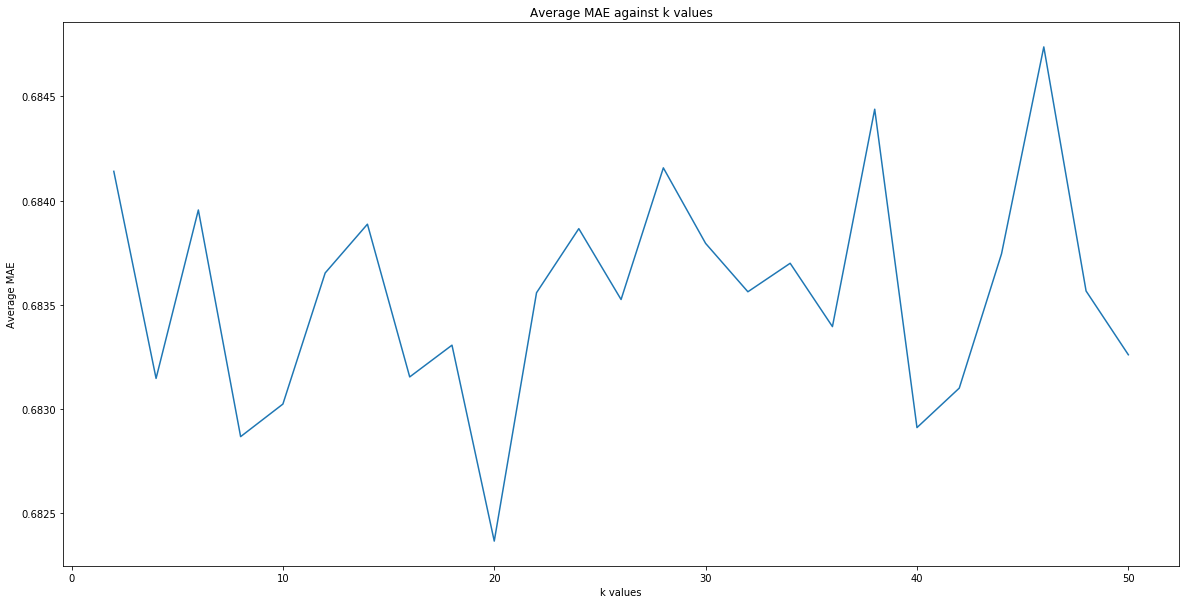


Fig 23: Graph showing Avg MAE vs k values

Question 25

**Use the plot from question 24, to nd 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.**

optimalk\_RMSE\_m, optimalk\_MAE\_m=16, 48

We see that the optimal number of latent factors is between 16 to 48. And the total movie genres are 20.

Average RMSE for minimum latent\_factors 16 is 0.888131550012

Average MAE for minimum latent\_factors 16 is 0.683600800394.

Question 26

**Design a MF with bias collaborative lter to predict the ratings of the movies in the popular movie trimmed test set 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 obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE**

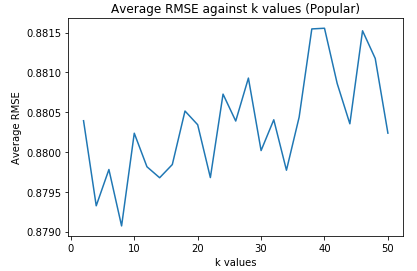


Fig 24: Graph showing Avg RMSE vs k value (Popular movies)

Minimum Average RMSE is 0.879070655298 for popular movies.

optimalk\_RMSE\_mp = 8

Question 27

**Design a MF with bias collaborative lter to predict the ratings of the movies in the unpopular movie trimmed test set 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 obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE**

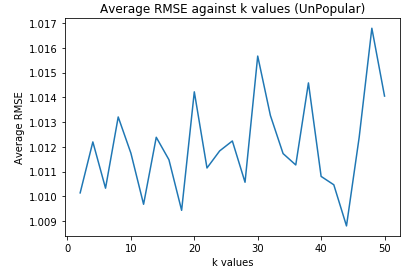


Fig 25: Graph showing Avg RMSE vs k value (Unpopular movies)

Minimum Average RMSE is 1.00880407369 for unpopular movies.

optimalk\_RMSE\_mu = 46

Question 28

**Design a MF with bias collaborative lter to predict the ratings of the movies in the high variance movie trimmed test set 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 obtained by averaging the RMSE across all 10 folds. Plot average RMSE (Y-axis) against k (X-axis). Also, report the minimum average RMSE.**

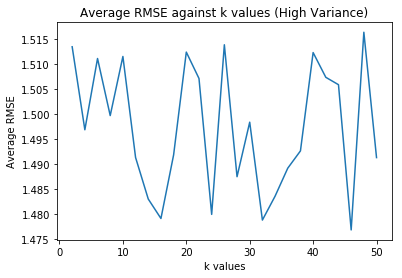


Fig 26: Graph showing Avg RMSE vs k value (High variance movies)

Minimum Average RMSE is 0.88978579912771527 for High Variance movies.

optimalk\_RMSE\_mv =26

minimum average RMSE = 1.18565331783

Question 29

**Plot the ROC curves for the MF with bias collaborative filter designed in question 24 for threshold values [2:5; 3; 3:5; 4]. For the ROC plotting use the optimal number of latent factors found in question 25. For each of the plots, also report the area under the curve (AUC) value.**

Optimal k =18 for RMSE

Area under the curve (AUC) value for threshold = 2.5

0.80

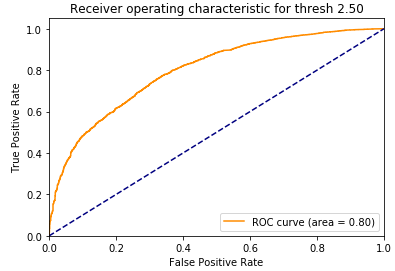


Fig 27: ROC curve for 2.5

Area under the curve (AUC) value for threshold = 3.0

0.80

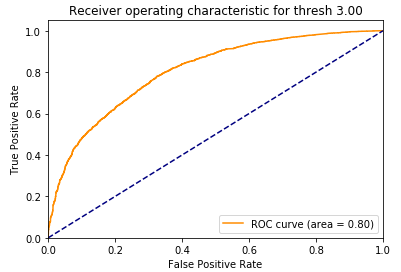


Fig 28: ROC curve for 3

Area under the curve (AUC) value for threshold = 3.5

0.78

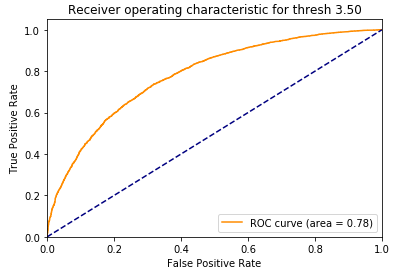


Fig 29: ROC curve for 3.5

Area under the curve (AUC) value for threshold = 4.0

0.79

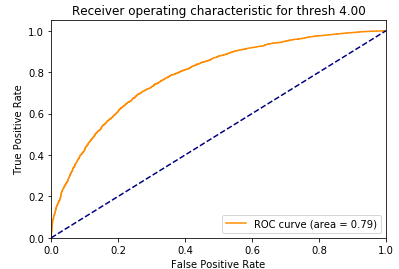


Fig 30: ROC curve for 4.0

Question 30:

**Design a naive collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.**

The Naive filter is very basic in nature. The reason it seems to have a decent performance even though it is just the mean rating for the user is because the rating scale of 1-5 with steps of 0.5 is a small range. To add to this, mose users do not really use the full spectrum from 0-5. Rather most movies are average / above average with some exceptional and some bad movies. Therefore often most of the ratings of a user are concentrated in a particular region with very little variance. For all the users, the max - min rating given by them is much lesser than the range of 1-5. Therefore a simple method such as predicting based on the mean rating is also able to give somewhat decent results for the dataset.

The average RMSE is: 0.9554178586618095

Question 31:

**Design a naive collaborative lter to predict the ratings of the movies in the popular movie trimmed test set and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.**

The average RMSE is: 0.9520945747748563

Question 32:

**Design a naive collaborative lter to predict the ratings of the movies in the unpopular movie trimmed test set and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.**

The average RMSE is: 1.010053315129645

Question 33:

**Design a naive collaborative lter to predict the ratings of the movies in the high variance movie trimmed test set and evaluate its performance using 10-fold cross validation. Compute the average RMSE by averaging the RMSE across all 10 folds. Report the average RMSE.**

The average RMSE is: 1.5217043595996964

Question 34:

**Plot the ROC curves (threshold = 3) for the k-NN, NNMF, 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.**

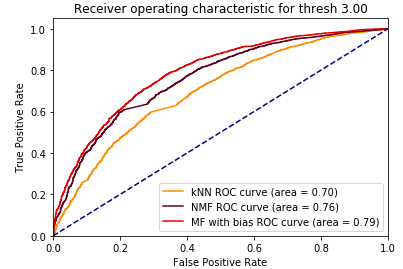


Fig 31: ROC curve

From the above figure we can observe that MF with bias performs the best with highest AUC of 0.79. NMF performs second best but knn performs the worst of all three. The reason is as follows:

In collaborative filtering, most of the matrix entries are missing, so something like KNN would not be possible since the nearest neighbors calculated on a handful of present features would largely be “unstable.” (as we see movies have sparsity of 0.98). It might be possible to make the nearest neighbor prediction more stable by imputing the missing values in the matrix, but this is precisely what matrix factorization does using the low-rank assumption and hence performs better than knn.

Question 35:

**Precision and Recall are defined by the mathematical expressions given by equations 12 and 13 respectively. Please explain the meaning of precision and recall in your own words.**

### Precision: Precision can be defined as the ratio of "number of correctly recalled events from the total number of events recalled"

### 

### Recall: Recall can be defined as the ratio of "total number of correctly recalled events from all correct events"

(Continued below)

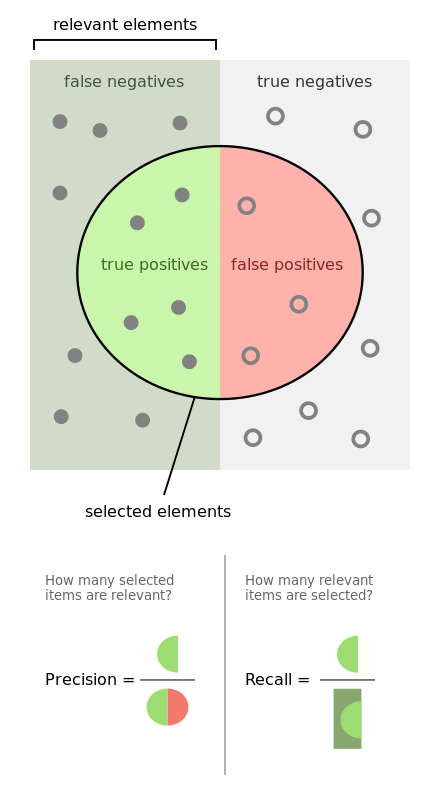


Fig: Precision and Recall (Source: Wikipedia)

Given a set of elements in the space. Let the dark gray area be the set of all positives (or relevant) elements and the light gray region to be the negative elements. The circular region represents the set of elements that were classified as relevant or positive by any model. As shown in the figure there are some false positives (i.e elements that were negative but the model called positive) and there are also false negative(elements that were relevant but we called negative).

Precision is defined as (tp / tp + fp) i.e. from the elements predicted positive how many were actually positive.

Recall is defined as (tp / tp + fn) i.e from all positive elements what ratio of elements was the model able to correctly predict as positive.

Precision and Recall are important parameters as often accuracy can be misleading, especially when the dataset is extremely skewed. eg: if the dataset has only 1% positive elements, predicting all elements to be negative would give an accuracy of 99% but if we look at the precision and recall, it would be 0.

Quite often there is an inverse relation between precision and recall and a tradeoff is required. This is because if our model has a lower ‘threshold’ for acceptance, it might have a good recall (catch more positive elements) but might also lead to lower precision as more false positives also sneak into the readings.

Conversely, a model that is extremely strict and has a high threshold might not let false positives sneak in and therefore have high precision but it also often means that it struggles to identify all positive examples from the superset itself and has lower recall.

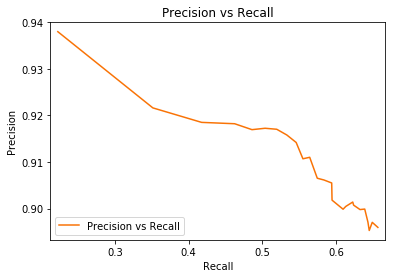
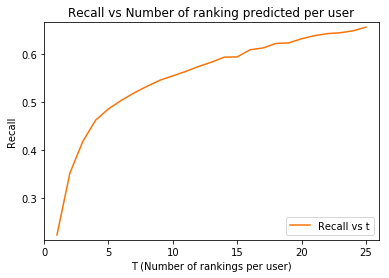
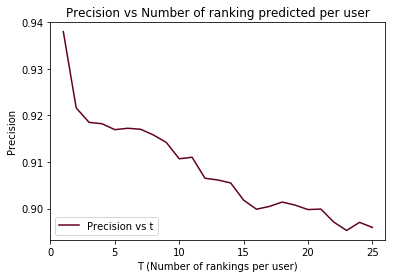
Based on the application for which the model is being built the importance to each parameter might be different.

eg: In recommendation systems false positives might be considered a bigger issue and therefore implying higher importance to precision. This is because missing out on some movies that the user may like( false negative) has less severe consequences as opposed to showing a user a movie that they do not like (false positive).

Whereas in medical applications, recall is more important than precision as it is better to have false positive( wrongly diagnose someone with cancer) as opposed to having a false negative. ( classifying a sick patient has healthy)

Question 36:

**Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using k-NN collaborative filter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use the k found in question 11 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.**

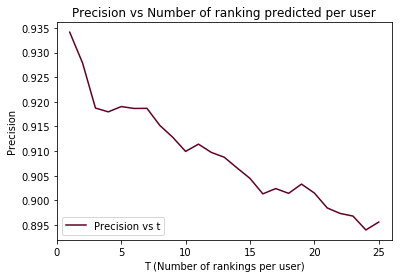


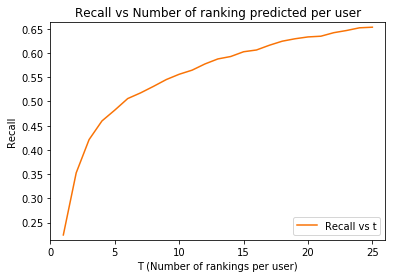
As discussed in the in definition of precision and recall, there seems to be an inverse relationship between the two. As the number of rankings increase, the precision decreases as we introduce more false positives into our model. Much lesser chances of false positive while guessing 1 favourite movie, as opposed to 25 favourite movies!!! (where there are bound to be errors). On the other hand as T increases the recall increases as we include more positive examples in our result. I.e , our ground truth is all the movies liked by the user(greater than 3.0 rating). For t = 1 we just find one movie and the denominator is much larger(ground set). As t increases we cover more ground truth examples and the recall increases.

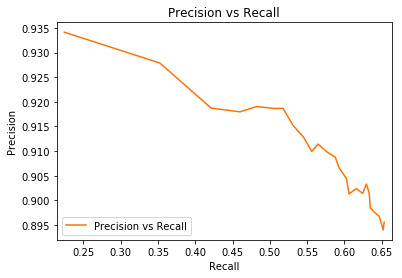
An interesting point to note would be modify this problem such that the ground truth does not have a threshold of 3.0 but instead is generated by adding all movies for the user where the rating given by the user is higher than the mean rating given by the user. This is much more intuitive and the drawbacks of a fixed threshold is to do with working with raw ratings as opposed to mean centered ratings as discussed in an problem earlier.

Question 37:

**Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using NNMF-based collaborative lter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use optimal number of latent factors found in question 18 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.**



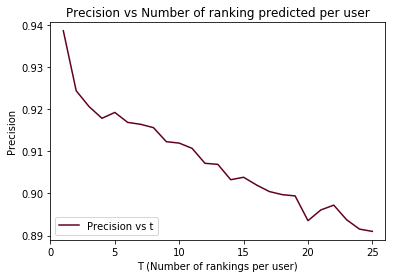


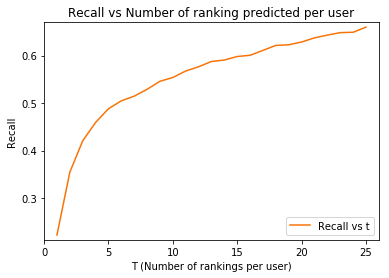


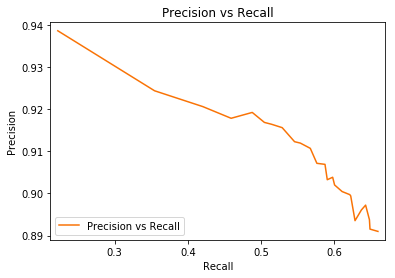
The patterns observed for NNMF with precision vs recall follow the same inverse relation as discussed for the case of KNN

Question 38:

**Plot average precision (Y-axis) against t (X-axis) for the ranking obtained using MF with bias-based collaborative lter predictions. Also, plot the average recall (Y-axis) against t (X-axis) and average precision (Y-axis) against average recall (X-axis). Use optimal number of latent factors found in question 25 and sweep t from 1 to 25 in step sizes of 1. For each plot, briefly comment on the shape of the plot.**



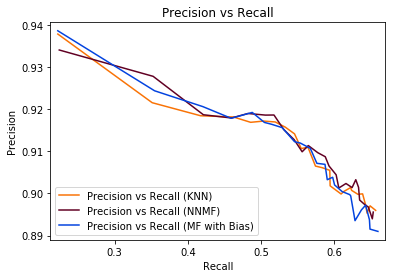




The pattern of precision vs recall follows the same inverse pattern for MF with bias as discussed for KNN and NNMF

Question 39:

**Plot the precision-recall curve obtained in questions 36,37, and 38 in the same gure. Use this gure to compare the relevance of the recommendation list generated using k-NN, NNMF, and MF with bias predictions.**

****

Observations: From the data above, we had observed that MF based collaborative filter is performing the best out of all the other filters (with the minimum RMSE value).

But in a precision recall scenario while the performances are comparable, MF with bias seems to perform better for high precision low recall scenarios, whereas KNN and NNMF seem to work better for high recall low precision scenarios.