Project 3 report
Liyuan Zhao
005692591

Question 1A

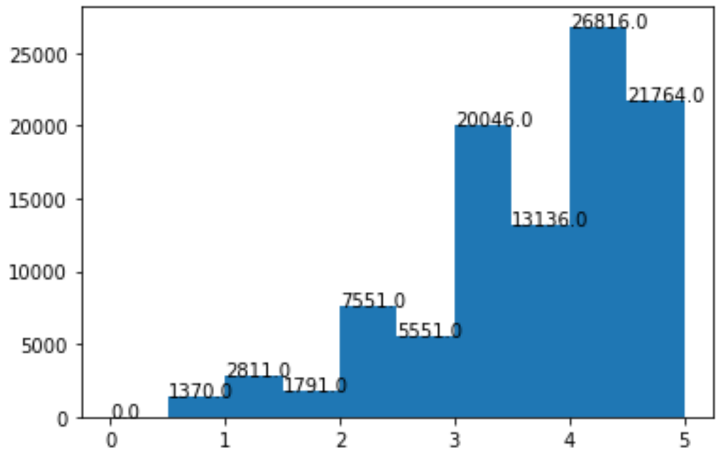
Sparsity: 0.016999683055613623

Question 1B

Question 1B

In [3]:
```
bins = np.linspace(0,5,num=11)
patches = plt.hist(rating, bins)

for i in range(10):
    plt.text(patches[1][i],patches[0][i],str(patches[0][i]))
```



The plot is not monotonic and it is skewed to the left. But there is a general increase in the count of movies from lower to high rating. Movies with rating 4.0 have the most movie counts from the datasets. We can also observe ratings with the whole numbers (3.0, 4.0, etc.) are relatively higher than those neighbors. We can assume that people are more likely to give a whole number of ratings.

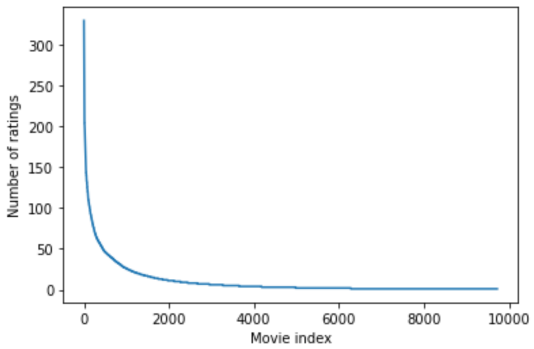
Question 1C

Question 1C

In [4]:
```
from collections import Counter

counterC = Counter(movie_id)
num_ratings = sorted(list(counterC.values()),reverse=True)
plt.plot(num_ratings)
plt.xlabel("Movie index")
plt.ylabel("Number of ratings")
```

Out[4]: Text(0, 0.5, 'Number of ratings')

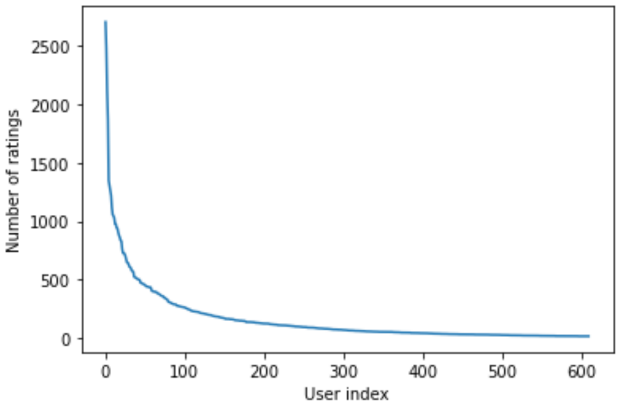


Question 1D

Question 1D

In [5]:
```
counterD = Counter(user_id)
num_users = sorted(list(counterD.values()),reverse=True)
plt.plot(num_users)
plt.xlabel("User index")
plt.ylabel("Number of ratings")
```

Out[5]: Text(0, 0.5, 'Number of ratings')



Question 1E

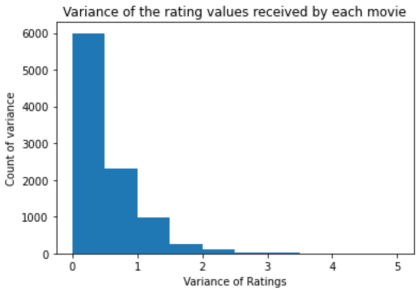
The plot is monotonically decreased.This is because most of the movies are not that ‘popular’ and thus most of the movies do not have sufficient ratings counts and the numbers of ratings decrease dramatically. Those movies with more ratings counts are in minority and treated as ‘popular’ movies. It makes the dimensions of the index of movie and user with the ratings to be too sparse and it is not helpful for implementing the models.

Question 1F

Question 1F

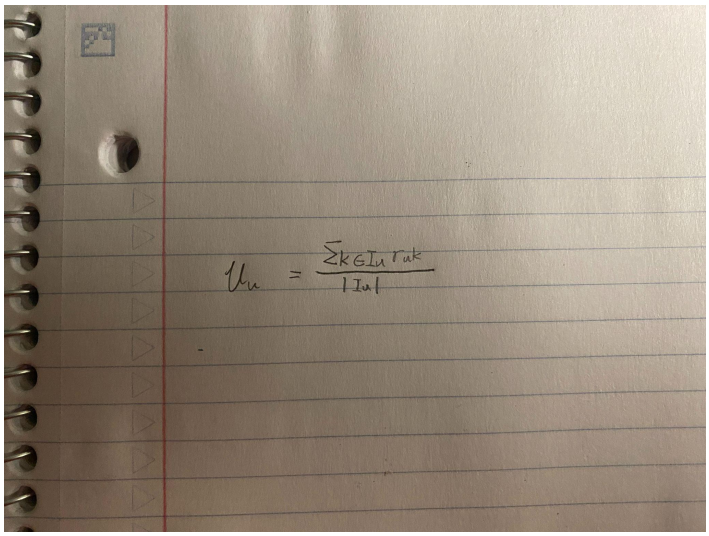
In [6]:
```
ratings_matrix = ratings.pivot(index = 'userId', columns = 'movieId', values = 'rating')
var_movie = np.var(ratings_matrix, axis=0)

plt.hist(var_movie, bins= np.linspace(0,5,num=11))
plt.title('Variance of the rating values received by each movie')
plt.xlabel('Variance of Ratings')
plt.ylabel('Count of variance')
plt.show()
```



The plot skewed to the right. Most of the variances are low with 0 to 1, which means that ratings values received by each movie do not vary much. This basically means that users tend to give ratings of any certain movie with a proper range, so that the ratings should be more consistent.

Question 2A



Question 2B

This represents the set of items that are rated by both user u and user v. This can be null, as some movies are rated by user u and not user v, or rated by user v not user u.

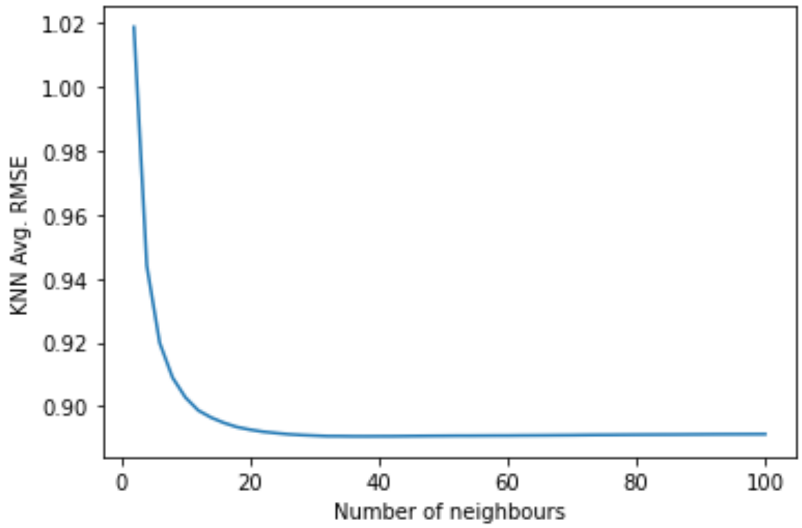
Question 3

Mean-centering will reduce the impact of different rating habits of users. For instance, user u and user u have similar taste for movies. However, if user u tends to rate stricter than user v, without mean-centering, the recommendation system may give inaccurate predictions. By applying the prediction function, ratings would become balanced and potential biases would be removed.

Question 4

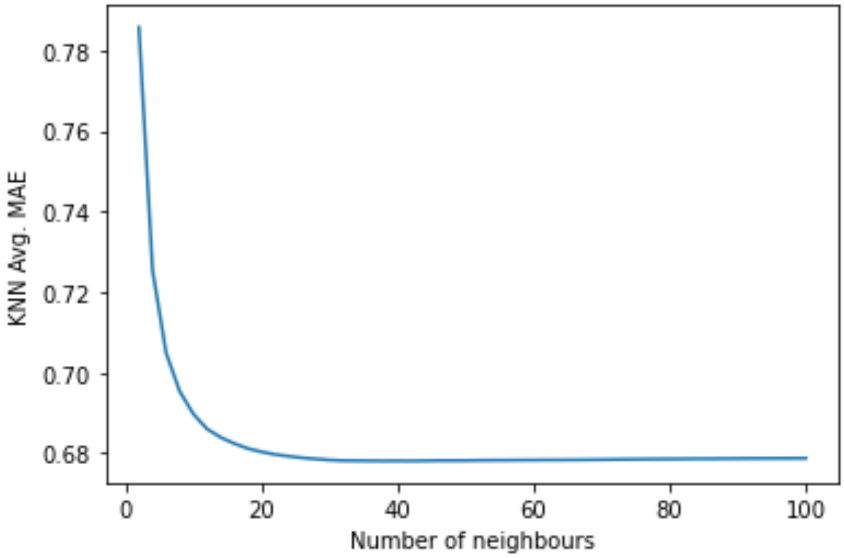
RMSE:

In [10]:
```
ks = np.arange(2,102,2)
plt.plot(ks, RMSE)
plt.ylabel('KNN Avg. RMSE')
plt.xlabel('Number of neighbours')
plt.show()
```



MAE:

In [11]:
```
plt.plot(ks, MAE)
plt.ylabel('KNN Avg. MAE')
plt.xlabel('Number of neighbours')
plt.show()
```



Question 5

Question 5

In [12]:
```
# absolute_min_RMSE = ks[RMSE.index(min(RMSE))]
# print(absolute_min_RMSE)
# absolute_min_RMSE = ks[MAE.index(min(MAE))]
# print(absolute_min_RMSE)
```

In [13]:
```
print('Average RMSE for KNN collaborative filter:', RMSE[20])
print('Average MAE for KNN collaborative filter:', MAE[20])
```

```
Average RMSE for KNN collaborative filter: 0.8906431823649316
Average MAE for KNN collaborative filter: 0.678099934962161
```

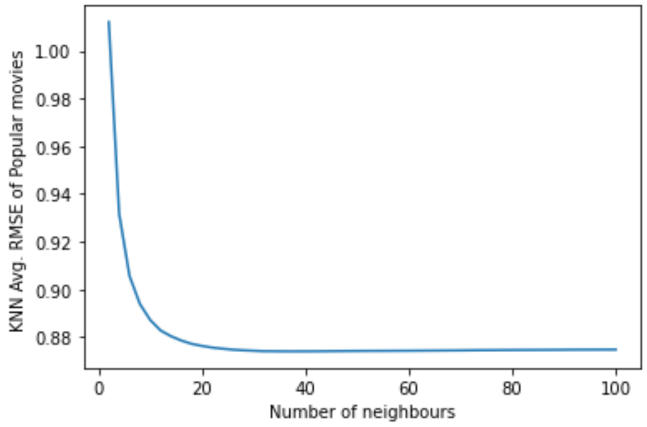
According to the plot from question 4, MAE and RMSE go to steady state at around k = 20

Question 6

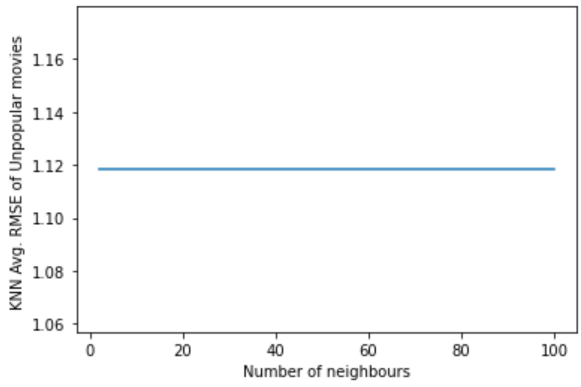
RMSE plots with respective to each k in each trimming

In [17]:
```
plt.plot(ks, popular_knn_RMSE_total)

plt.ylabel('KNN Avg. RMSE of Popular movies ')
plt.xlabel('Number of neighbours')
plt.show()
```



In [18]:
```
#plt.plot(ks, unpopular_knn_RMSE_total)
plt.plot(ks,unpopular_knn_RMSE_total)
plt.ylabel('KNN Avg. RMSE of Unpopular movies ')
plt.xlabel('Number of neighbours')
plt.show()
```

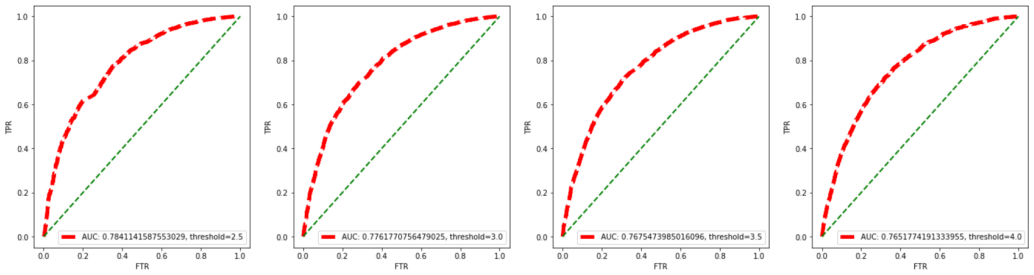


KNN Minimum average RMSE for popular movie 0.8740885885547431
KNN Minimum average RMSE for unpopular movie 1.1183580423196242
KNN Minimum average RMSE for high variance movie 1.483615190158854

This is the ROC curve to split the whole dataset to 90% and 10%

In [24]:
```
print('ROC curves for the k-NN collaborative filters with 90% training set')
plot_ROC(thres, pred_KNN)
```

```
ROC curves for the k-NN collaborative filters with 90% training set
```

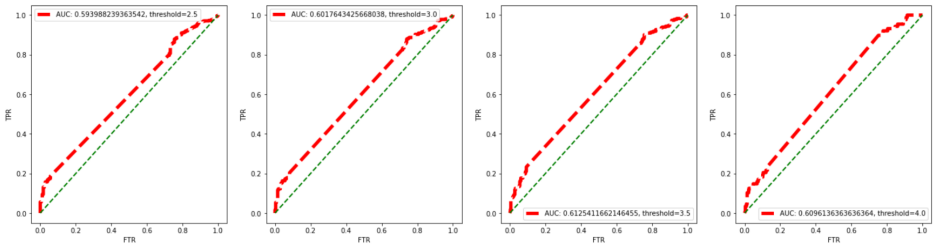


The following ROC curves are for each category's trimming.

In [26]:
```
print('ROC curves for the k-NN collaborative filters for unpopular movie trimming')

unpopular_best_pred_knn = find_best_pred(KNN_min_RMSE_unpopular_index, 'knn', 'unpopular')
plot_ROC(thres, unpopular_best_pred_knn)
```

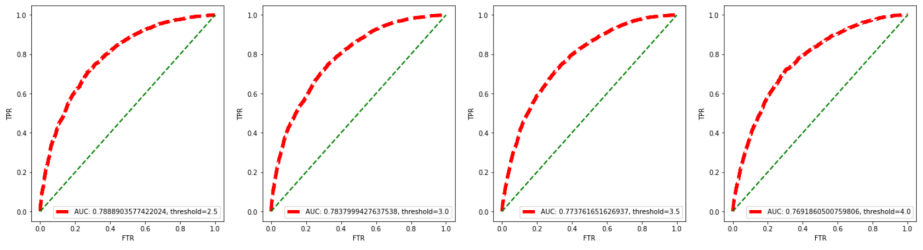
```
ROC curves for the k-NN collaborative filters for unpopular movie trimming
```



In [25]:
```
print('ROC curves for the k-NN collaborative filters for popular movie trimming')

popular_best_pred_knn = find_best_pred(20, 'knn', 'popular')
plot_ROC(thres, popular_best_pred_knn)
```

```
ROC curves for the k-NN collaborative filters for popular movie trimming
```

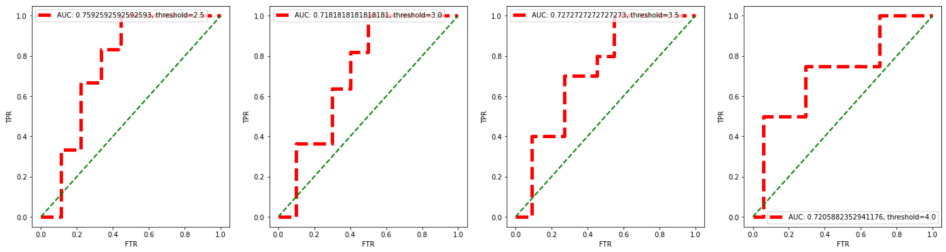


In [27]:

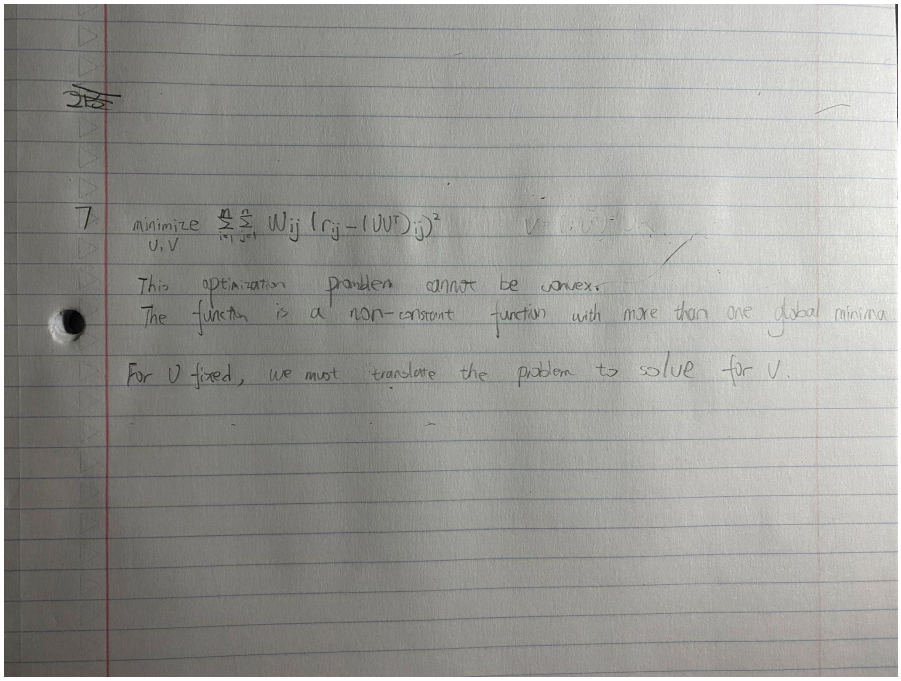
```
print('ROC curves for the k-NN collaborative filters for high variance movie trimming')

variance_best_pred_knn = find_best_pred(KNN_min_RMSE_variance_index, 'knn', 'variance')
plot_ROC(thres, variance_best_pred_knn)
```

```
ROC curves for the k-NN collaborative filters for high variance movie trimming
```

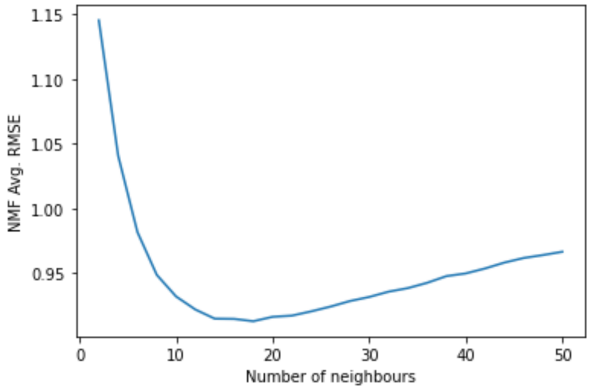


Question 7

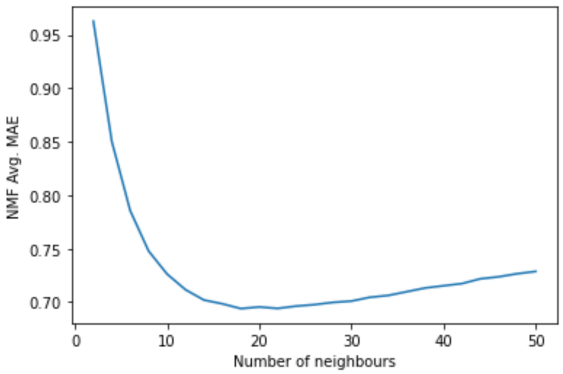


Question 8

In [29]:
```
plt.plot(ks_Q8, NMF_RMSE)
plt.ylabel('NMF Avg. RMSE')
plt.xlabel('Number of neighbours')
plt.show()
```



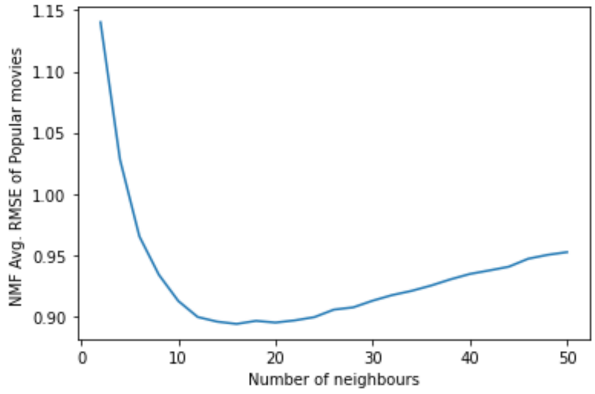
In [30]:
```
plt.plot(ks_Q8, NMF_MAE)
plt.ylabel('NMF Avg. MAE')
plt.xlabel('Number of neighbours')
plt.show()
```



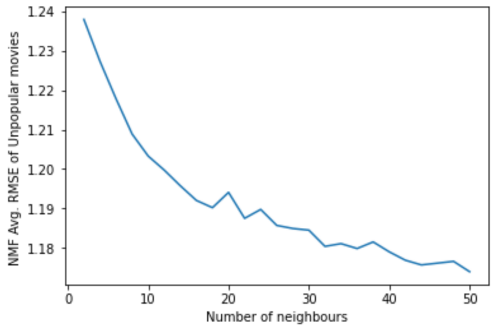
Average RMSE for NMF collaborative filter: 0.9125778741903167, k is: 18
Average MAE for NMF collaborative filter: 0.6939573421921161, k is: 18
The optimal latent for RMSE and MAE are both 18. The genre of the datasets is 19. They are not the same.

RMSE for each trimming are as follow

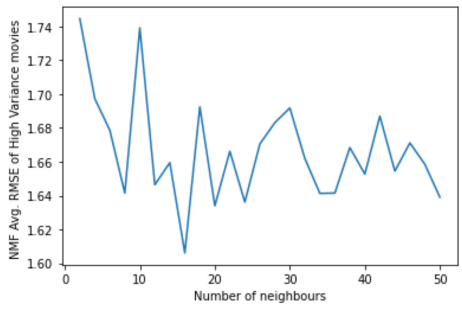
In [33]:
```
plt.plot(ks_Q8, popular_nmf_RMSE_total)
plt.ylabel('NMF Avg. RMSE of Popular movies ')
plt.xlabel('Number of neighbours')
plt.show()
```



In [34]:
```
plt.plot(ks_Q8, unpopular_nmf_RMSE_total)
plt.ylabel('NMF Avg. RMSE of Unpopular movies ')
plt.xlabel('Number of neighbours')
plt.show()
```



In [35]:
```
plt.plot(ks_Q8, variance_nmf_RMSE_total)
plt.ylabel('NMF Avg. RMSE of High Variance movies ')
plt.xlabel('Number of neighbours')
plt.show()
```

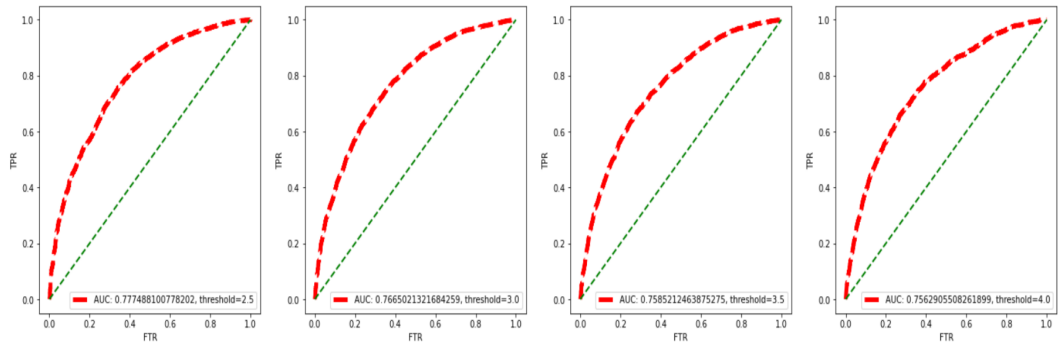


NMF Minimum average RMSE for popular movie 0.8939432266639107
NMF Minimum average RMSE for unpopular movie 1.173941480208685
NMF Minimum average RMSE for high variance movie 1.6061148313268372

The ROC curve for each trimming are as follow

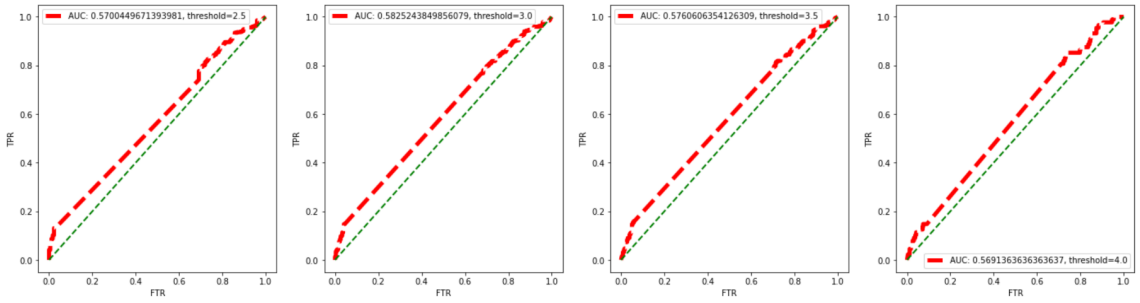
In [37]:
```
print('ROC curves for the NMF collaborative filters for popular movie trimming')
popular_best_pred_nmf = find_best_pred(NMF_min_RMSE_popular_index, 'nmf', 'popular')
plot_ROC(thres, popular_best_pred_nmf)
```

```
ROC curves for the NMF collaborative filters for popular movie trimming
```



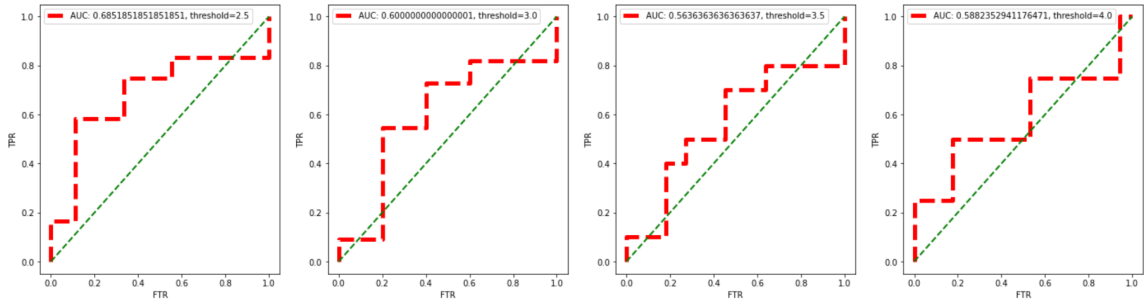
In [38]:
```
print('ROC curves for the NMF collaborative filters for unpopular movie trimming')
unpopular_best_pred_nmf = find_best_pred(NMF_min_RMSE_unpopular_index, 'nmf', 'unpopular')
plot_ROC(thres, unpopular_best_pred_nmf)
```

```
ROC curves for the NMF collaborative filters for unpopular movie trimming
```



In [39]:
```
print('ROC curves for the NMF collaborative filters for variance movie trimming')
variance_best_pred_nmf = find_best_pred(NMF_min_RMSE_variance_index, 'nmf', 'variance')
plot_ROC(thres, variance_best_pred_nmf)
```

```
ROC curves for the NMF collaborative filters for variance movie trimming
```



Question 9

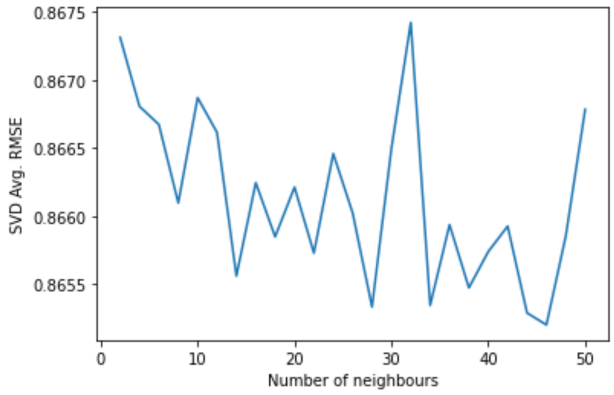
```
column = 0
Musical|Romance
Action|Adventure|Thriller
Adventure|Drama|Western
Documentary
Drama|Sci-Fi
Action|Comedy|Crime|Drama
Drama|Mystery|Romance|Thriller
Drama
Action|Adventure|Comedy|Crime
Horror
-----------------------------------------------------------------
column = 1
Comedy
Fantasy|Horror|Thriller
Action|Comedy
Drama|Thriller
Action|Crime|Thriller
Comedy|Drama
Documentary
Horror|Thriller
Comedy|Mystery
Action|Adventure|Fantasy
-----------------------------------------------------------------
column = 2
Action|Drama
Comedy|Romance
Comedy
Comedy|Thriller
Comedy|Drama|Romance
Action|Adventure|Drama|War
Comedy|Romance
Comedy|Drama|Romance
Comedy|Drama
Documentary
```

I choose columns 0,1 and 2. It belongs to a small collection of genres, including Drama, Action, Comedy, etc. The NMF latent factors seem to show how some of the interest groups can be related to genre.

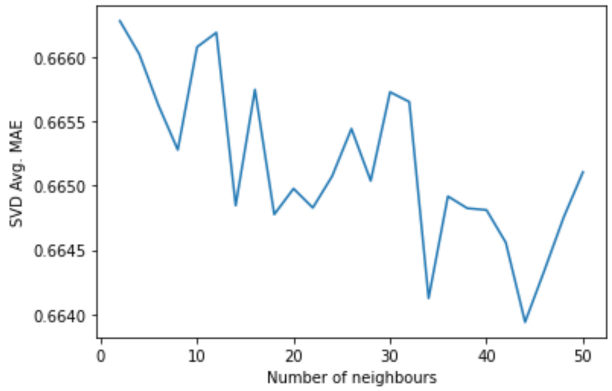
Question 10

In [43]:

```
plt.plot(ks_Q10, SVD_RMSE)
plt.ylabel('SVD Avg. RMSE')
plt.xlabel('Number of neighbours')
plt.show()
```



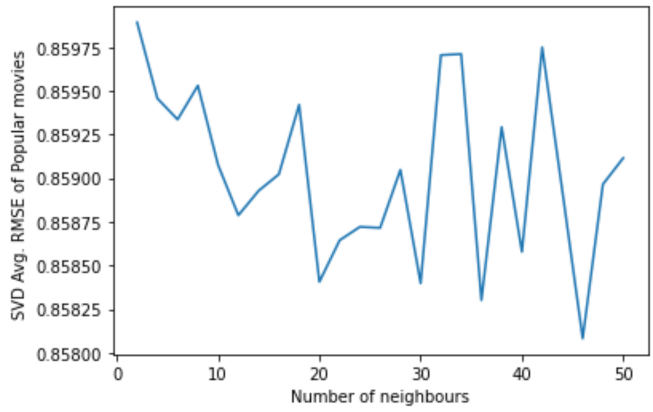
In [44]:
```
plt.plot(ks_Q10, SVD_MAE)
plt.ylabel('SVD Avg. MAE')
plt.xlabel('Number of neighbours')
plt.show()
```



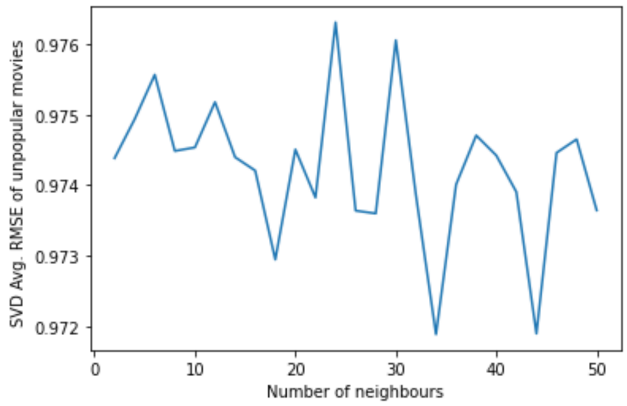
Average RMSE for MF collaborative filter: 0.8652007992734795, k is: 46
Average MAE for MF collaborative filter: 0.6639386364059771, k is: 44
The optimal latent for RMSE and MAE are 46 and 4. The genre of the datasets is 19, which is different.

RMSE for each trimming are as follow

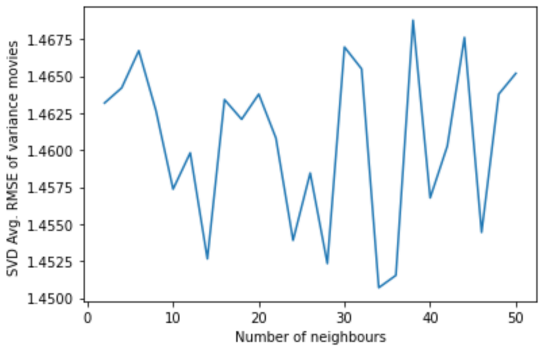
In [47]:
```
plt.plot(ks_Q10, popular_svd_RMSE_total)
plt.ylabel('SVD Avg. RMSE of Popular movies ')
plt.xlabel('Number of neighbours')
plt.show()
```



In [48]:
```
plt.plot(ks_Q10, unpopular_svd_RMSE_total)
plt.ylabel('SVD Avg. RMSE of unpopular movies ')
plt.xlabel('Number of neighbours')
plt.show()
```



In [49]:
```
plt.plot(ks_Q10, variance_svd_RMSE_total)
plt.ylabel('SVD Avg. RMSE of variance movies ')
plt.xlabel('Number of neighbours')
plt.show()
```

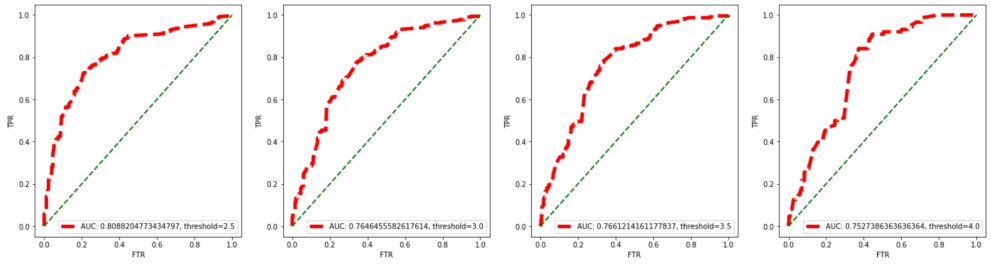


SVD Minimum average RMSE for popular movie 0.858081375714006
SVD Minimum average RMSE for unpopular movie 0.971883228888276
SVD Minimum average RMSE for high variance movie 1.4507015236766354

The ROC curve for each trimming are as follow

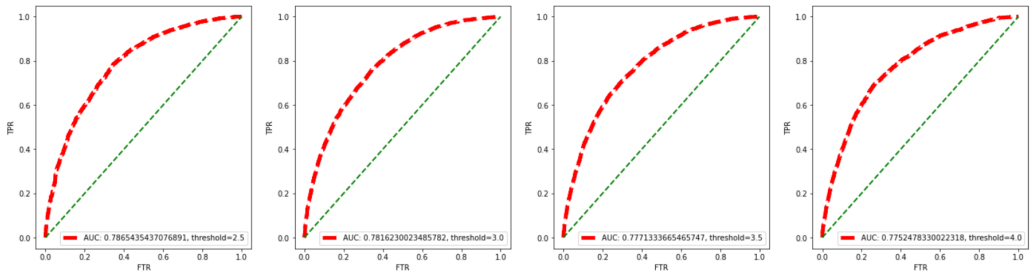
In [52]:
```
print('ROC curves for the MF collaborative filters for unpopular movie trimming')
unpopular_best_pred_svd = find_best_pred(SVD_min_RMSE_unpopular_index, 'svd', 'unpopular')
plot_ROC(thres, unpopular_best_pred_svd)
```

ROC curves for the MF collaborative filters for unpopular movie trimming



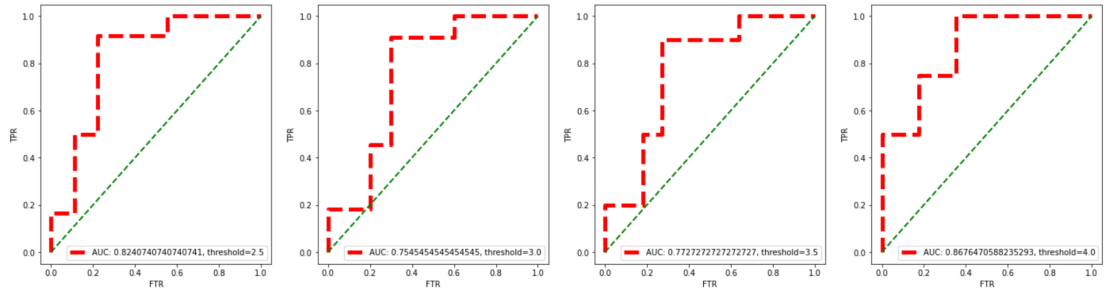
In [51]:
```
print('ROC curves for the MF collaborative filters for popular movie trimming')
popular_best_pred_svd = find_best_pred(SVD_min_RMSE_popular_index, 'svd', 'popular')
plot_ROC(thres, popular_best_pred_svd)
```

ROC curves for the MF collaborative filters for popular movie trimming



In [53]:
```
print('ROC curves for the MF collaborative filters for high variance movie trimming')
variance_best_pred_svd = find_best_pred(SVD_min_RMSE_variance_index, 'svd', 'variance')
plot_ROC(thres, variance_best_pred_svd)
```

ROC curves for the MF collaborative filters for high variance movie trimming

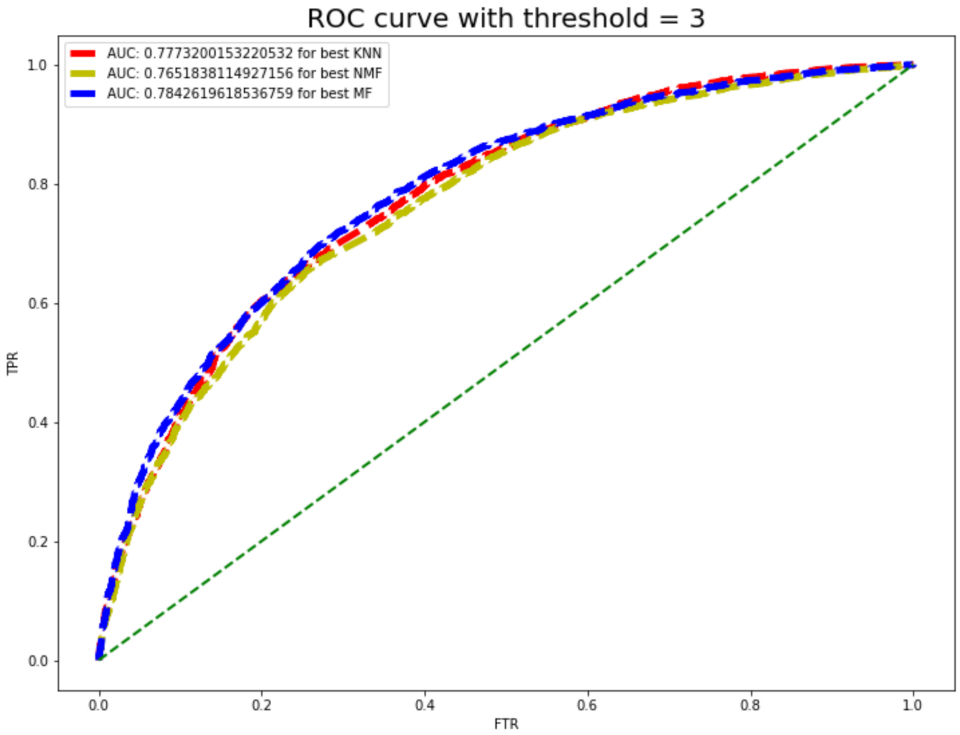


Question 11

Average RMSE for naive collaborative filter: 0.9346962471248572

Average RMSE for naive collaborative filter with popular Movie Trimming: 0.9323053680970579
Average RMSE for naive collaborative filter with unpopular Movie Trimming: 0.9704968354669647
Average RMSE for naive collaborative filter with variance Movie Trimming: 1.502179016093384

Question 12

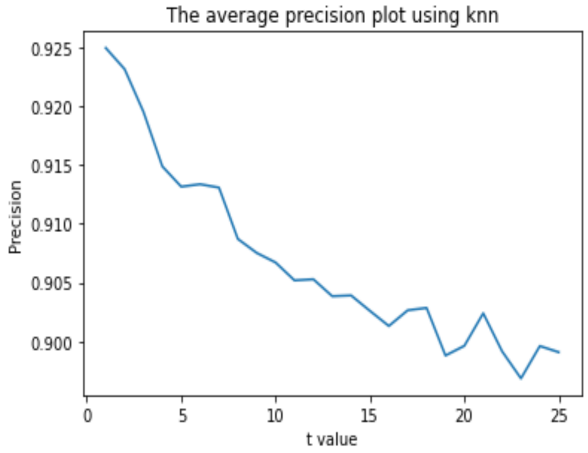


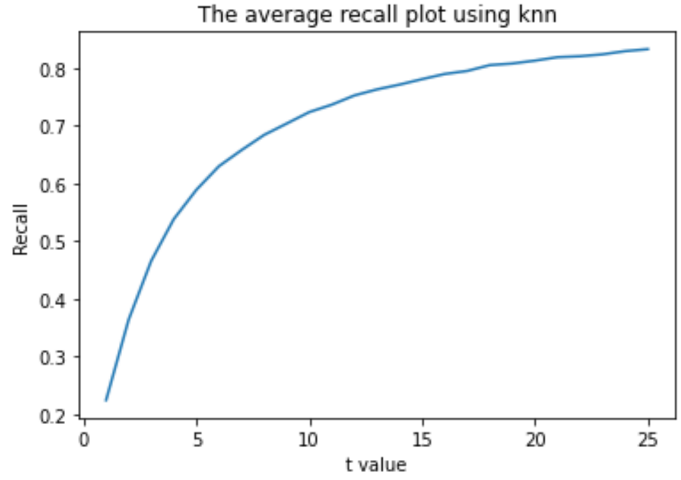
Question 13

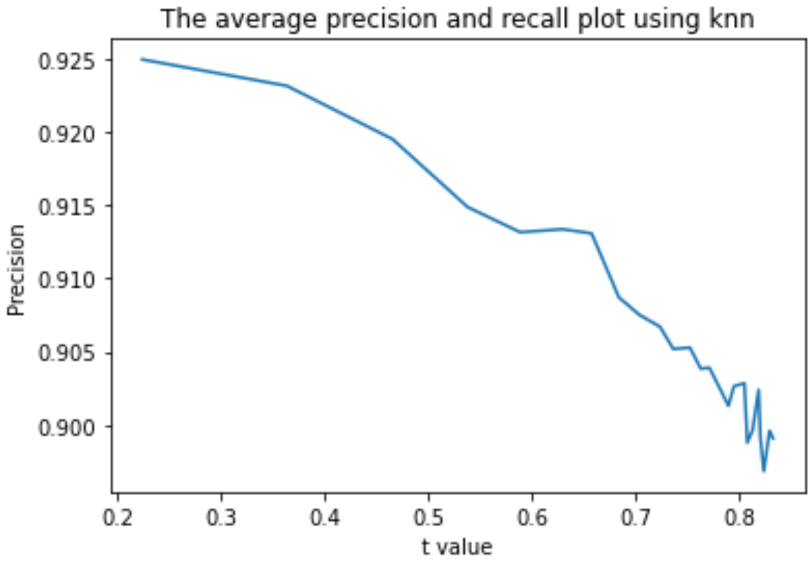
Precision looks at all of the predicted values from the model, and tells us which of those were correct. It tells us about how reliable a prediction is after the model deems it relevant.
Recall looks at all the ratings that the user actually likes, and within that set, how many overlap with the model's predictions. This gives us a sense of how much of the real values the model actually retrieves.

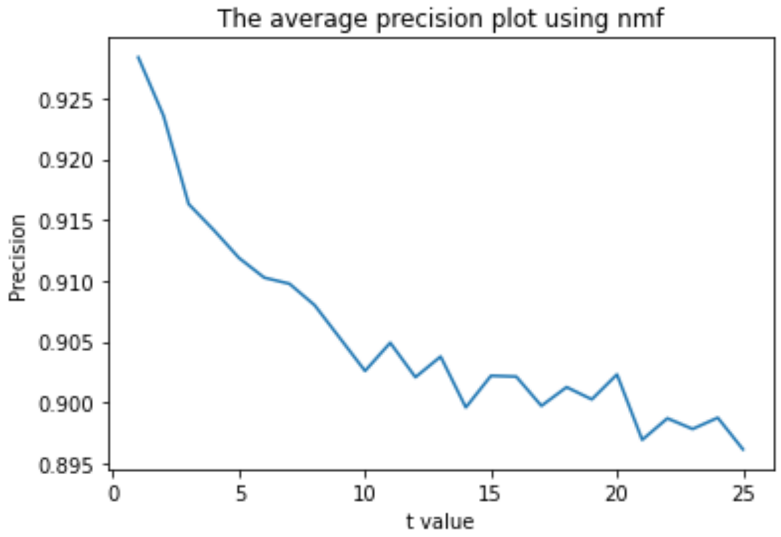
A model can have high precision and low recall if it's really good at predicting true positives from the actual positives. It can have low precision and high recall if it predicts everything to be positive while there are many non-positives in the data. It has high precision and high recall if it identifies all the positive values in the ground truth as positive.

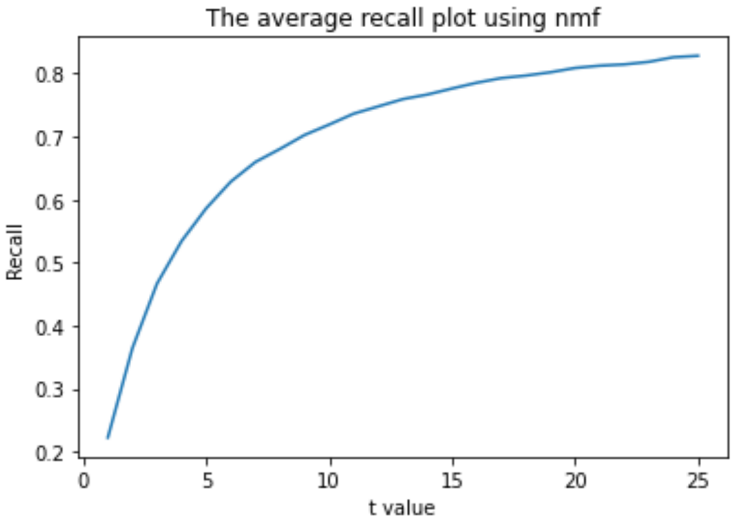
Question 14

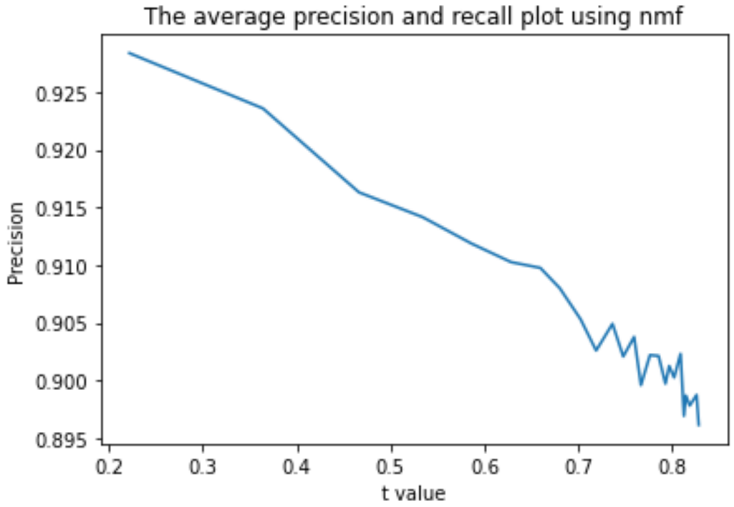

The average precision plot using knn
Precision
t value
0.925
0.920
0.915
0.910
0.905
0.900
0
5
10
15
20
25

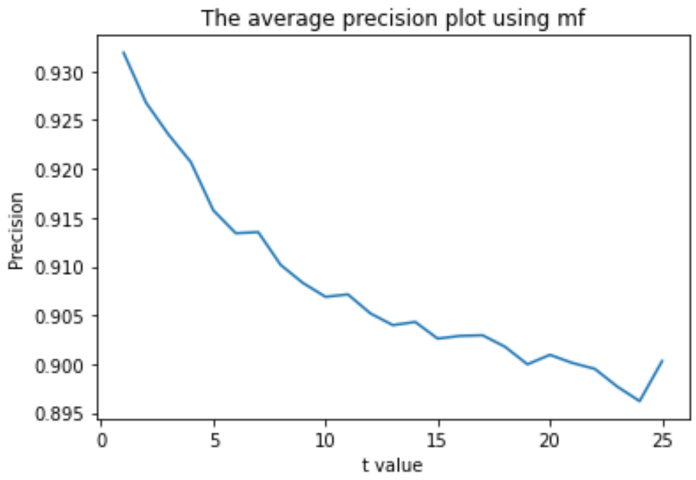

The average recall plot using knn
Recall
t value
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
5
10
15
20
25

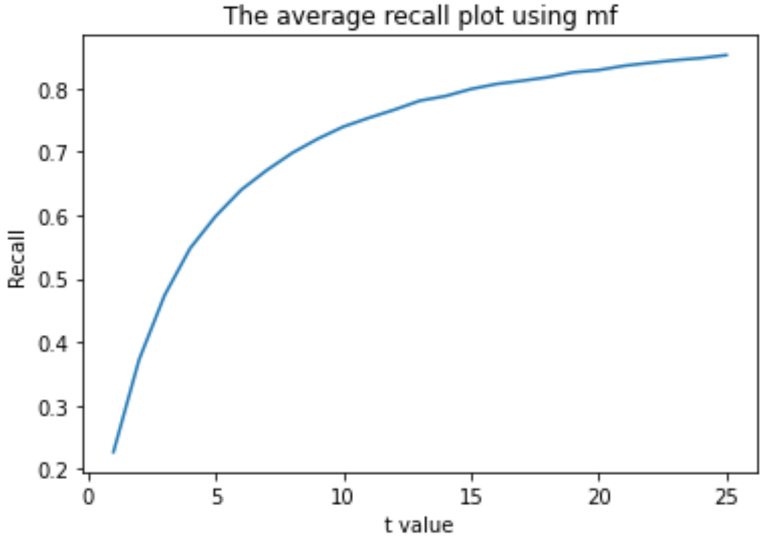

The average precision and recall plot using knn
Precision
t value
0.925
0.920
0.915
0.910
0.905
0.900
0.2
0.3
0.4
0.5
0.6
0.7
0.8

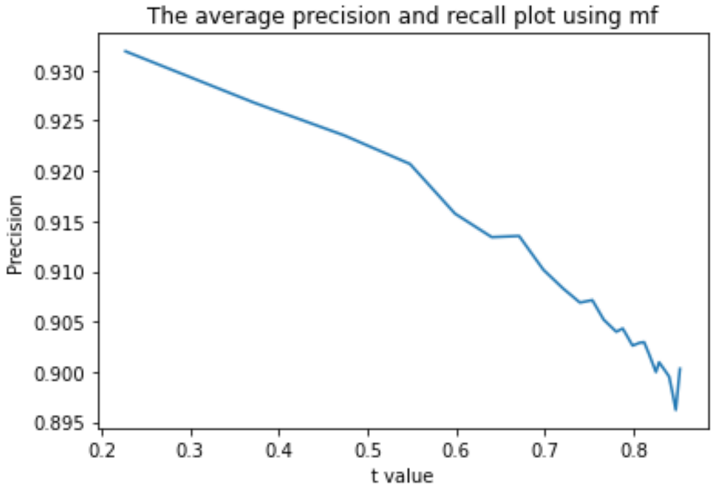

The average precision plot using nmf
Precision
t value
0.925
0.920
0.915
0.910
0.905
0.900
0.895
0
5
10
15
20
25

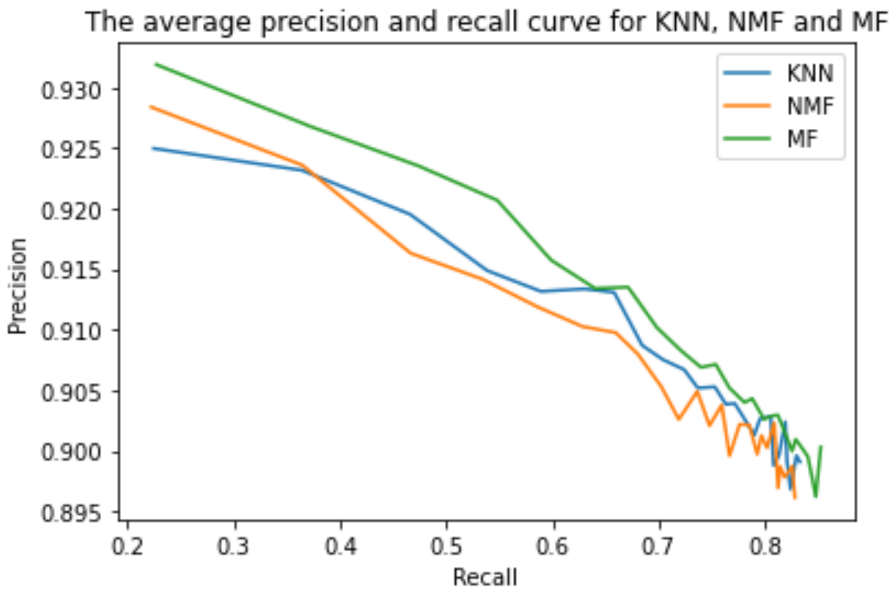

The average recall plot using nmf
Recall
t value
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
5
10
15
20
25


The average precision and recall plot using nmf
Precision
t value
0.925
0.920
0.915
0.910
0.905
0.900
0.895
0.2
0.3
0.4
0.5
0.6
0.7
0.8


The average precision plot using mf
Precision
t value
0.930
0.925
0.920
0.915
0.910
0.905
0.900
0.895
0
5
10
15
20
25


The average recall plot using mf
Recall
t value
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
5
10
15
20
25


The average precision and recall plot using mf
Precision
t value
0.930
0.925
0.920
0.915
0.910
0.905
0.900
0.895
0.2
0.3
0.4
0.5
0.6
0.7
0.8



For three models, they have a similar plot for each subplot. For the precision and t, we observe a decreasing trend of precision when t increases. This is because when the size increases, it will probably hit the ground-truth positives by picking the movie with the highest predicted rating. For example, if t = 20, then the top 20th movie may not be the one liked by the user with some predicted rating relatively closer to the threshold 3.
For the recall and t, there is a logarithmic increase for recall as t increases. From the recall function, when t increases,it will lead to the monotonic increase in recall.
For the last plot, there is a negative relationship between precision and recall. Recall increases will lead to the precision decrease. Recall increases will make the model include more items that are not liked by users, which will cause low precision.