EE-219 Project 1

Shrey Agarwal (004943082)

Varun Saboo (505028591)

February 12, 2018

Project 2: Clustering

**Objective –**

To find proper representations of the data, s.t. the clustering is efficient and gives out reasonable results.

To perform K-means clustering on the dataset, and evaluate the performance of the clustering.

To try different preprocess methods which may increase the performance of the clustering.

**Dataset –**

We work with “20 Newsgroups” dataset. It is a collection of approximately 20,000 documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic. Each topic can be viewed as a “class”.

TASK 1:

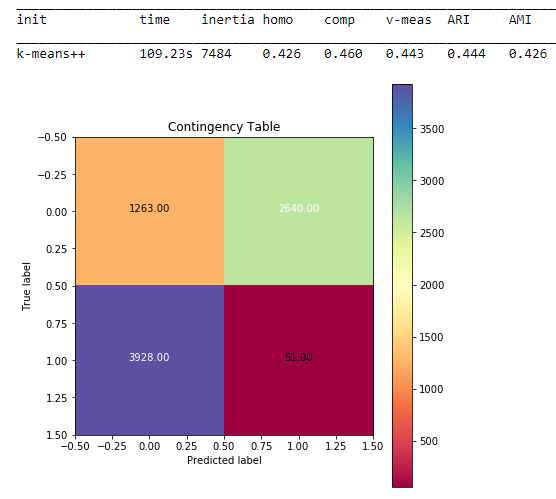
We perform TF-IDF on the documents by performing certain preprocessing steps such as stemming, word tokenizing, special characters removal and stopwords removal.

With min\_df = 3, the dimension of the TF-IDF matrix is (7882, 16564), where 7882 are the total number of documents in 8 classes and 16564 are the unique tokens identified.

TASK 2:

After applying k-means clustering on the above dataset, we got the following results.

With number of clusters = 2:



TASK 3:

Preprocess the data:

Dimensionality reduction:

1. LSI (Latent Semantic Indexing):

After performing LSI on the TF-IDF matrix, following are the obtained results:

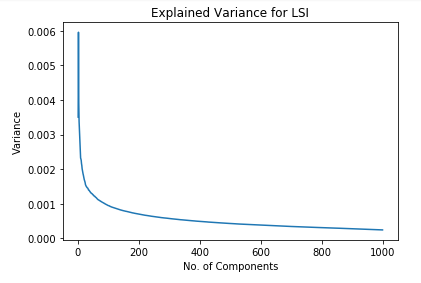
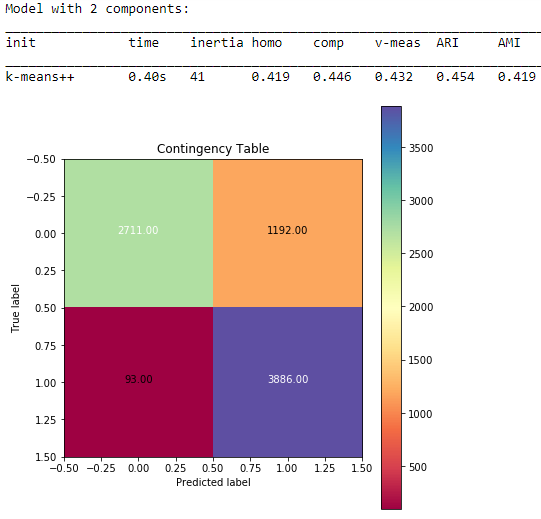
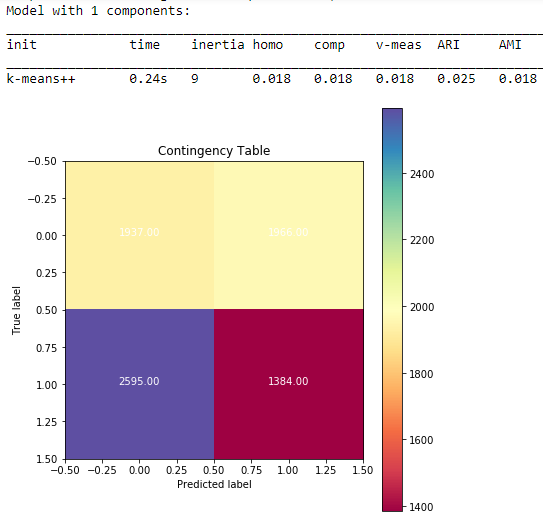
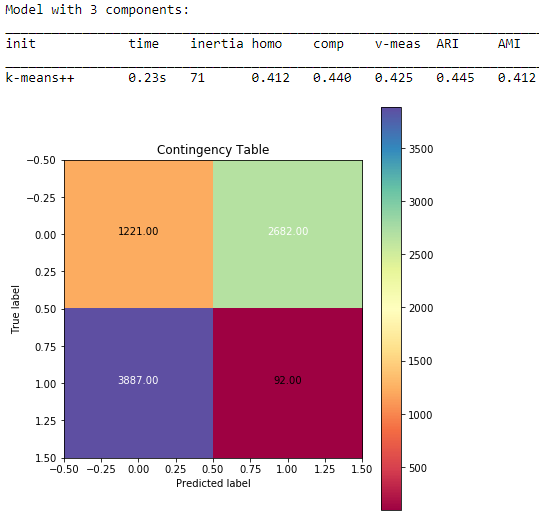
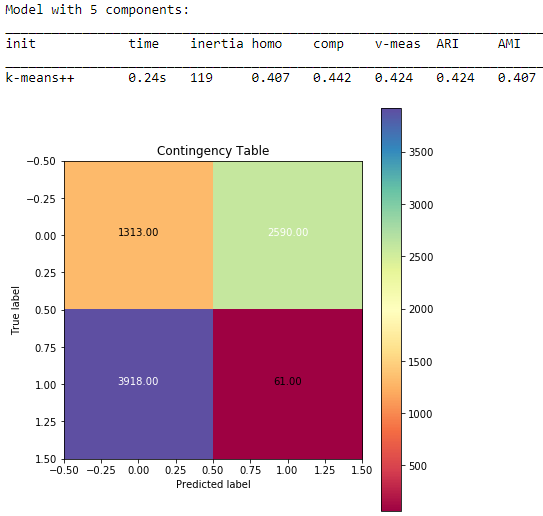
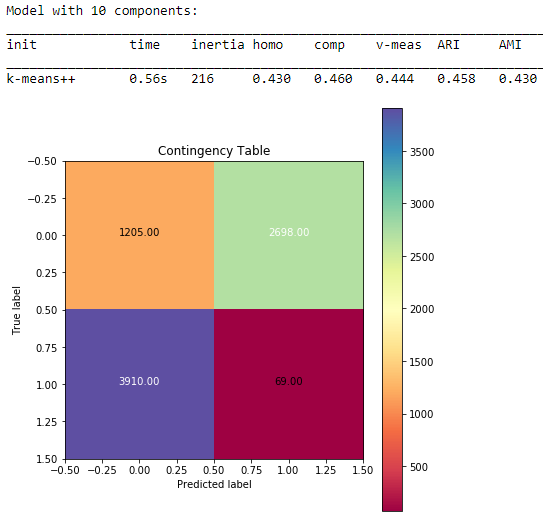


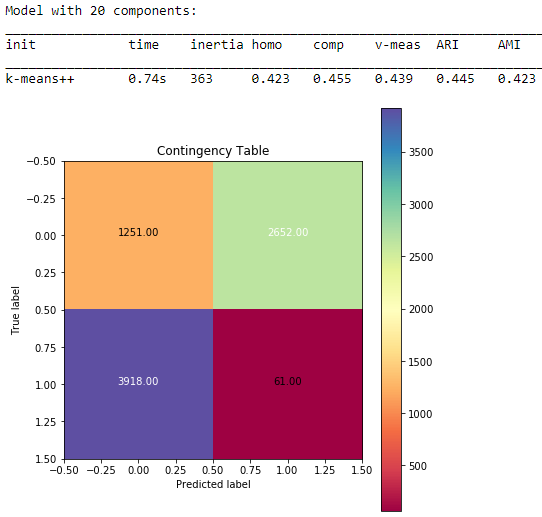
Figure 1: We can observe that the variance decreases as the no. of components increase.

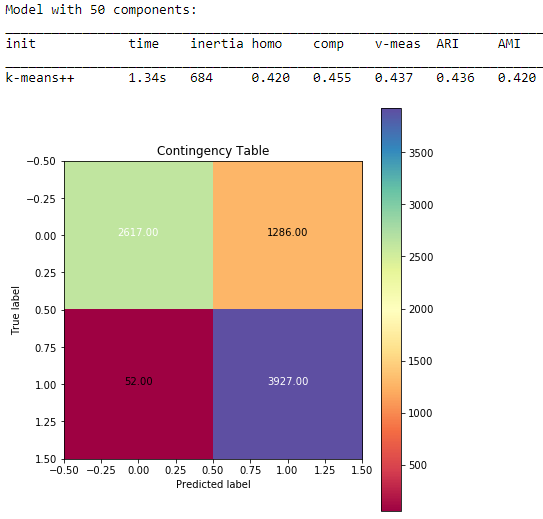


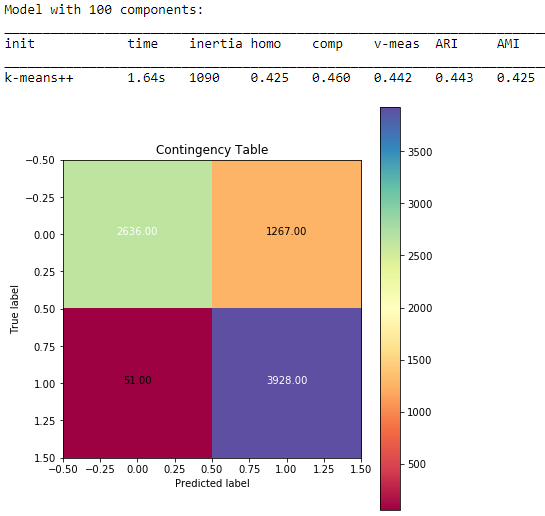


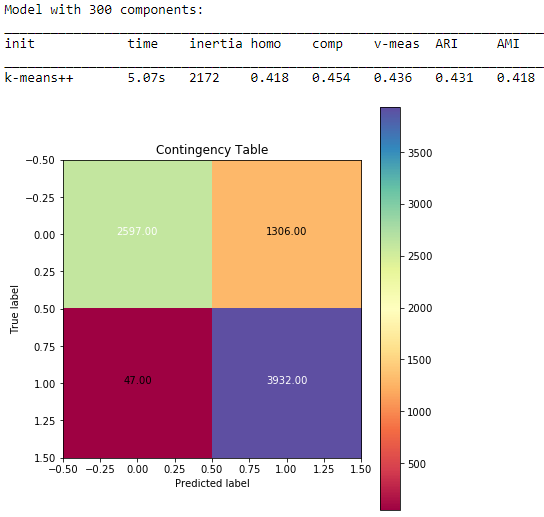


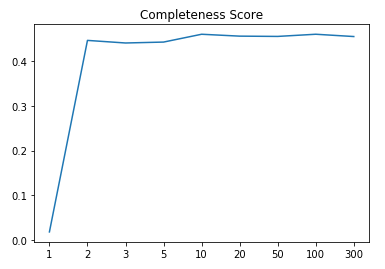
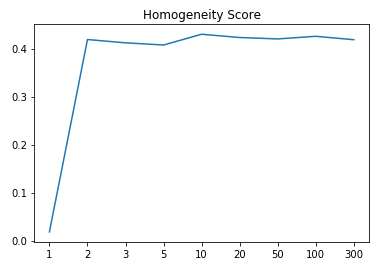
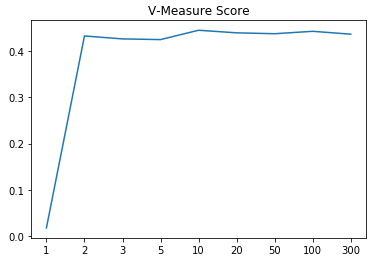
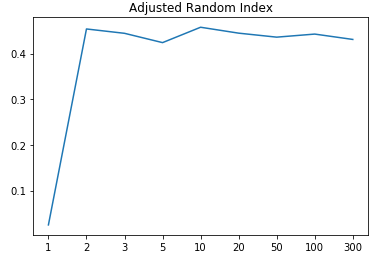
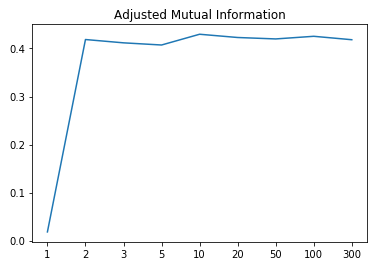








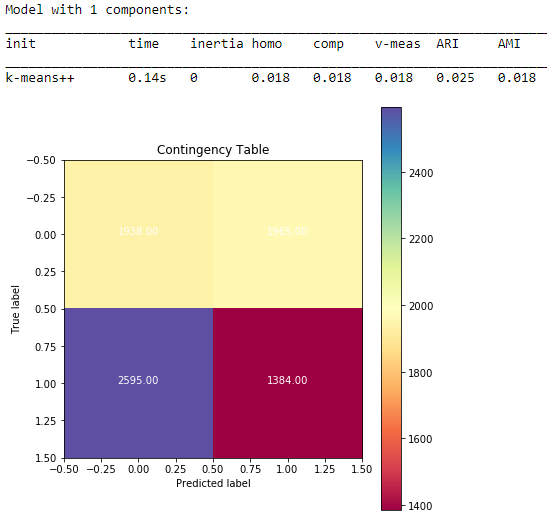


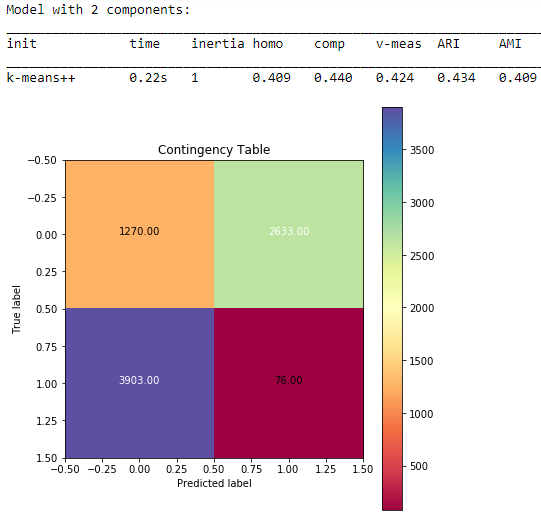
   

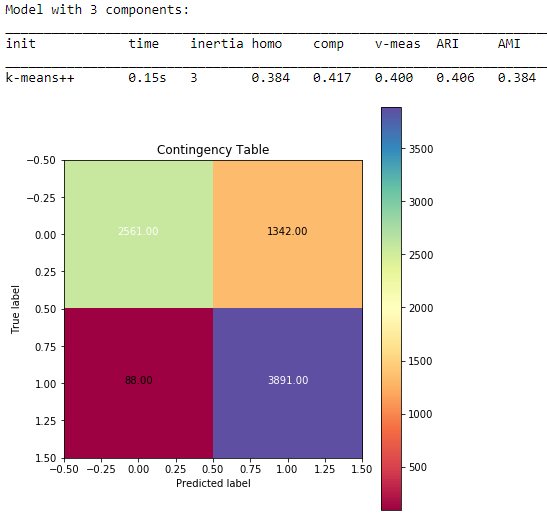
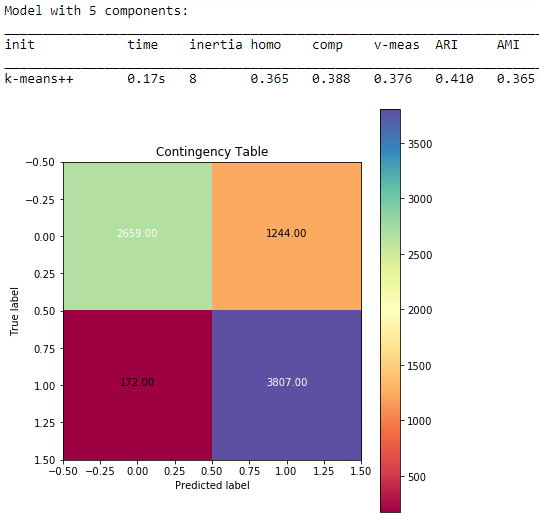
As we can observe from the contingency matrix as well as the 4 measures above, the best r value for LSI is 10. i.e, 10 components are best suitable to represent each document and cluster with maximum accuracy.

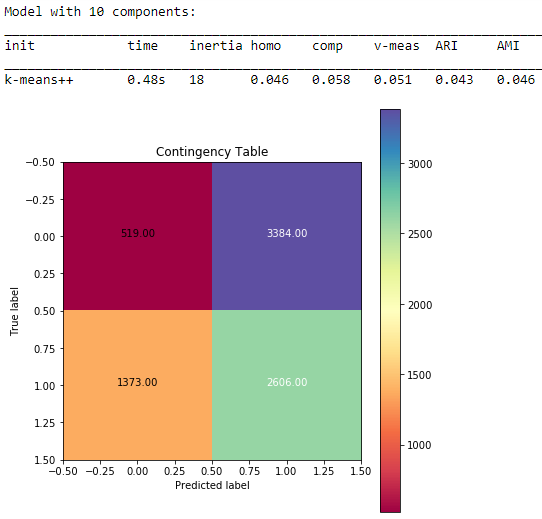
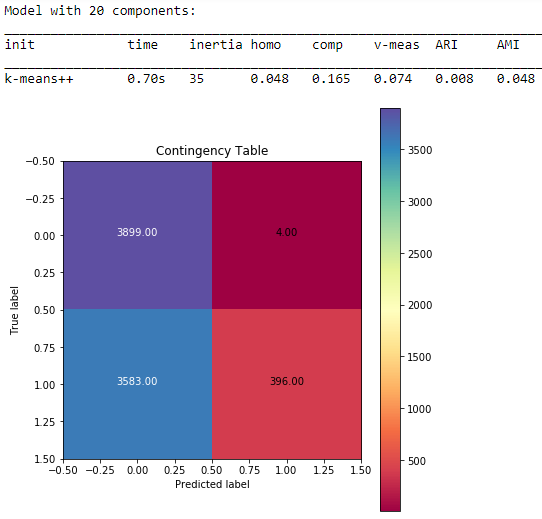
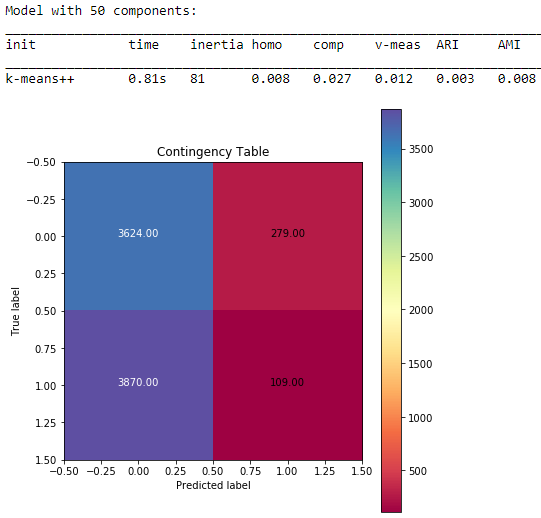
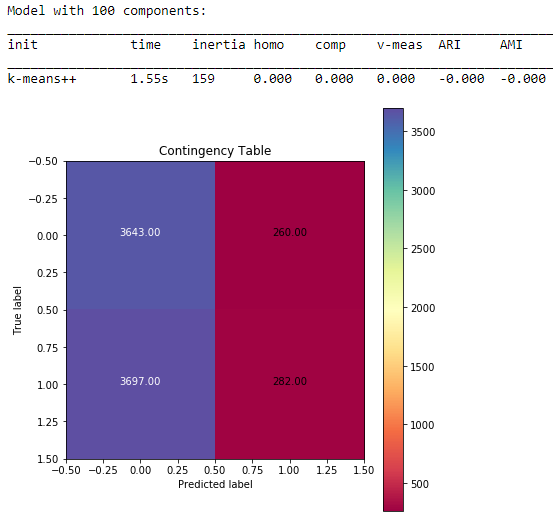
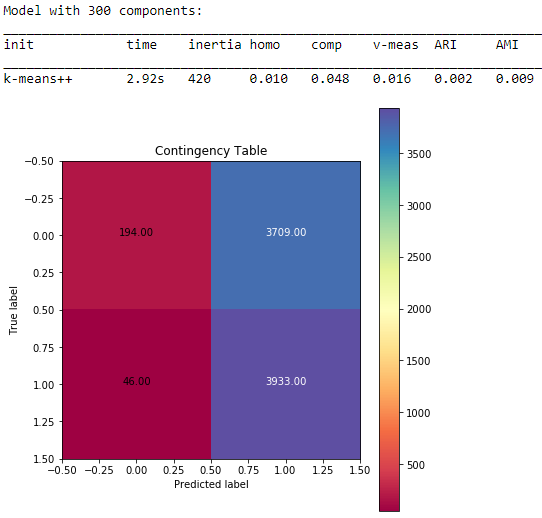
1. NMF (Non-Negative Matrix Factorization):

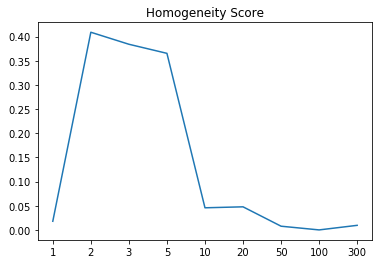
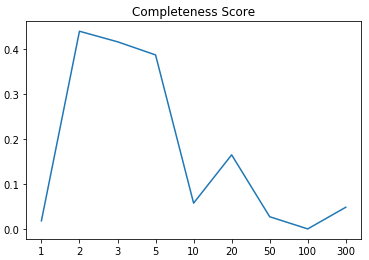
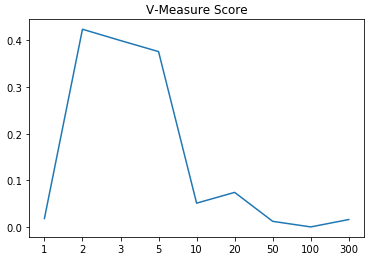
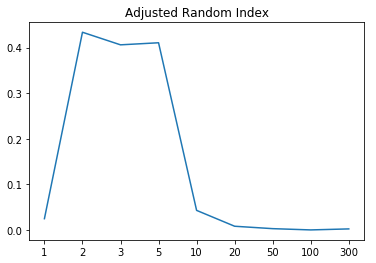
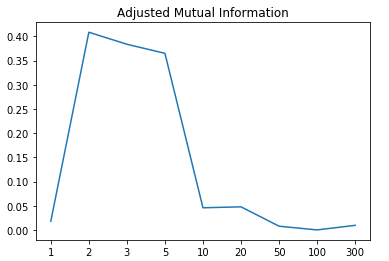
After performing NMF on the TF-IDF matrix, following are the obtained results:





As we can observe, the best r value is at r = 2. i.e, only 2 components/features are more than enough to best cluster the documents in two classes with maximum accuracy.

The non-monotonic behavior can be explained by the following reasoning. As we increase the dimension of representation of the data, the data points begin to get close to each other. On the other hand, increasing the dimension of data representation provides intricate details which helps in distinguishing each data point. Hence, a proper balance needs to be found to represent data in best dimension space. Here, we observed that the clustering improved performance as we increase dimension but started performing worse for higher dimensions. Hence, it exhibits the non-monotonic behavior.

TASK 4:

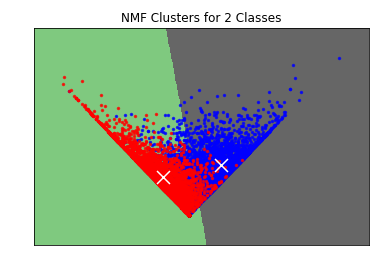
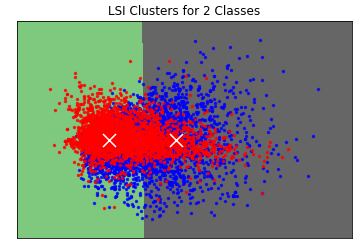
Visualize the performance:

1. We found that,

r=5 works best in case of LSI, and

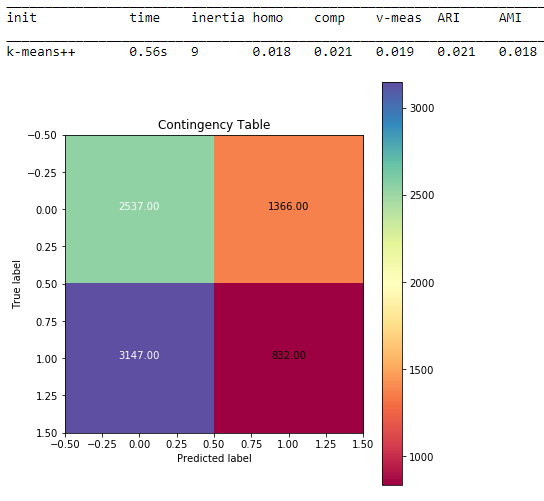
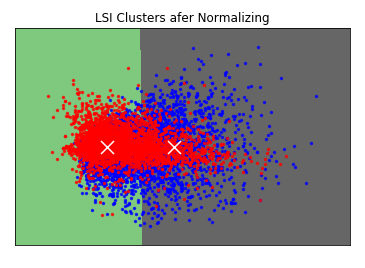
r=2 works best in case of NMF.

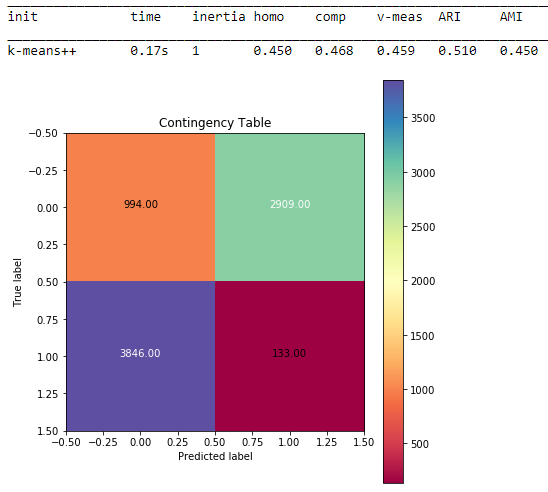
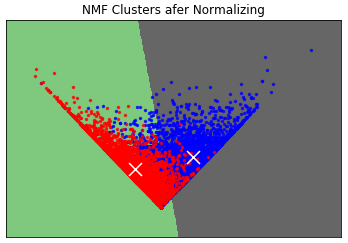
Hence, following are the clusters with their decision boundary for the two methods.



1. 3 Methods:
   1. Normalizing:

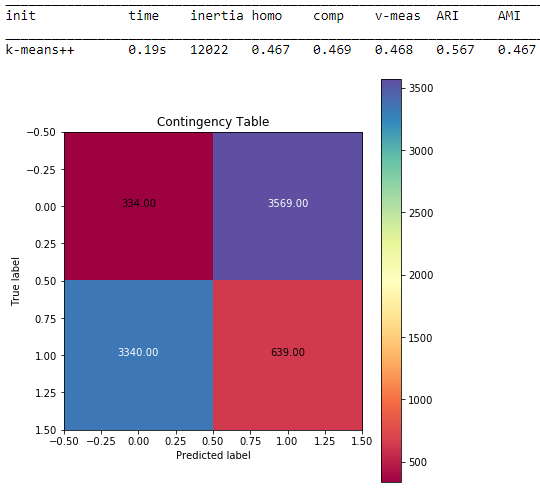
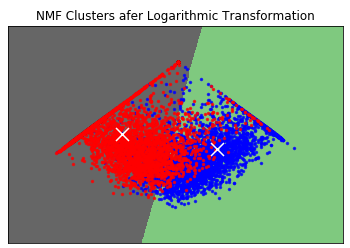
We performed scaling using sklearn “scale” to get unit variance and then normalized the data. Following are the results for LSI and NMF:

As we can observe, the performance has improved from 83.69% to 85.7% which is a good improvement.

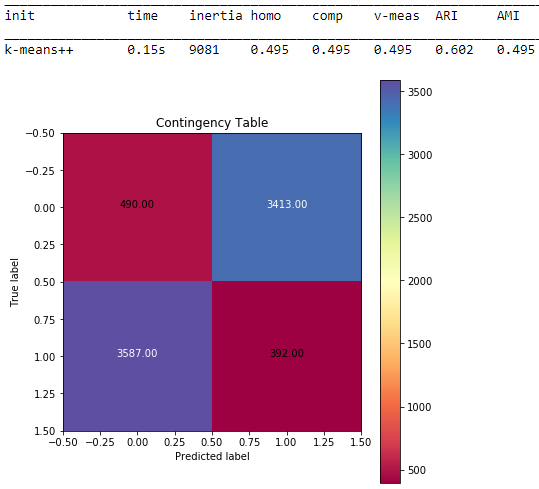
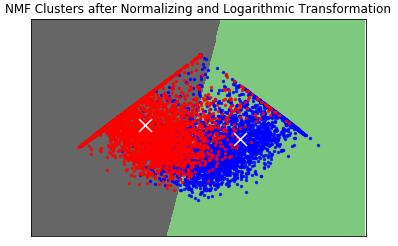
* 1. Logarithmic Transformation:

As we can observe, the performance has improved from 83.69% to 87.65% which is a very good improvement.

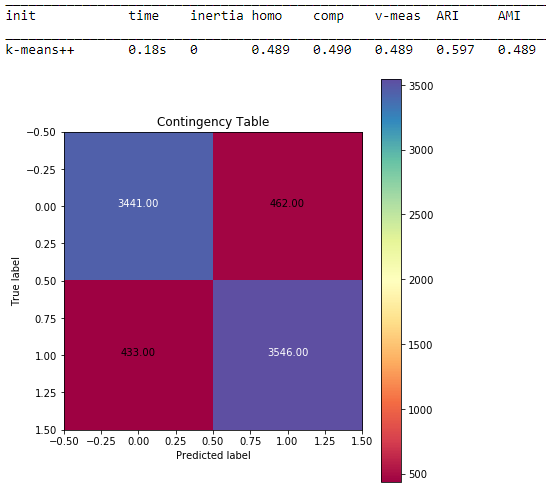
* 1. Combination
     1. Normalization and then Log Transformation

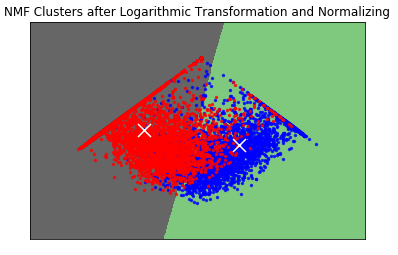
We perform normalization and then apply log transformation on NMF data to get the following results:

As we can observe, the performance has improved from 83.69% to 88.80% which is higher than normalization or log transformation separately. Infact, it is the highest and best combination so far.

* + 1. Log Transformation and then Normalization





As we can observe, the performance has improved from 83.69% to 88.64% which is higher than normalization or log transformation separately.

TASK 5: Expand to 20 categories

We follow the same workflow as above with all 20 categories and produce following results and observations:

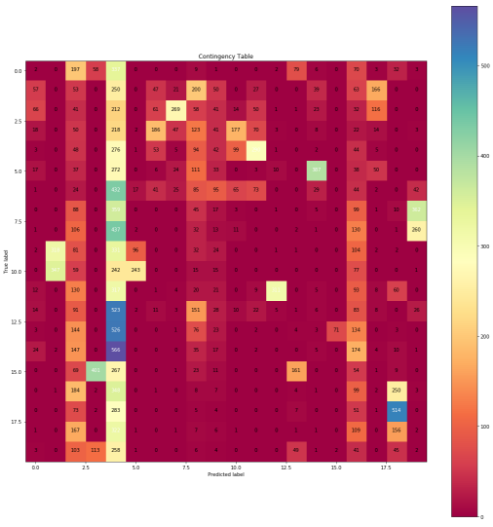
1. We perform TF-IDF on the documents by performing certain preprocessing steps such as stemming, word tokenizing, special characters removal and stopwords removal.

With min\_df = 3, the dimension of the TF-IDF matrix is (18846, 33158), where 18846 are the total number of documents in 20 categories and 33158 are the unique tokens identified.

1. After applying k-means clustering on the above dataset, we got the following results.

With number of clusters = 2:





1. Preprocess the data:

Dimensionality reduction:

* 1. LSI (Latent Semantic Indexing):

After performing LSI on the TF-IDF matrix, following are the obtained results:

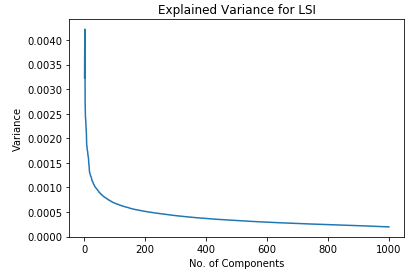
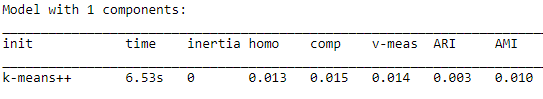
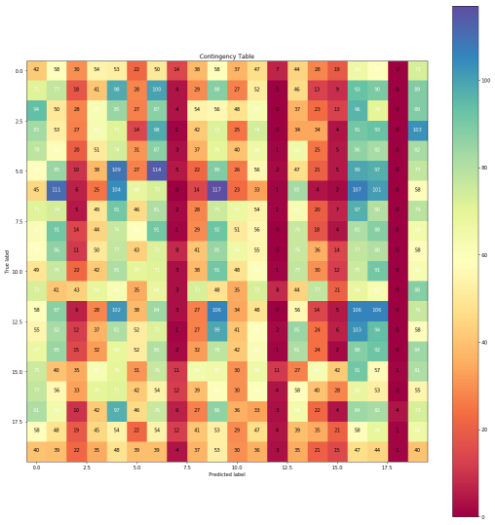
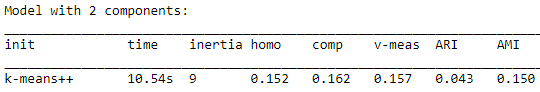
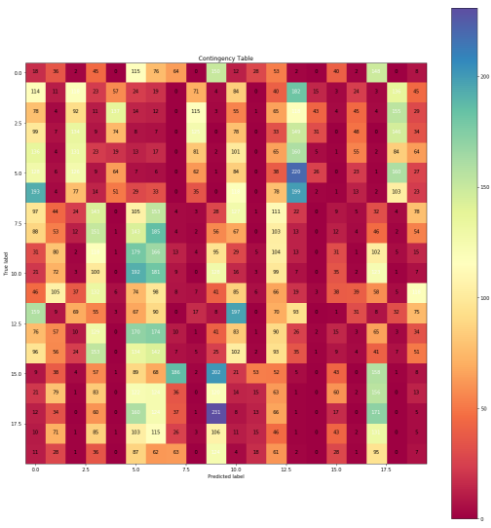


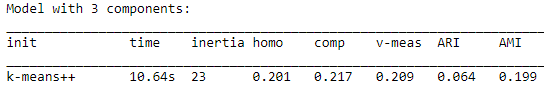
Figure 2: We can observe that the variance decreases as the no. of components increase. Interestingly, the variance is even more lesser than previous 8 categories.

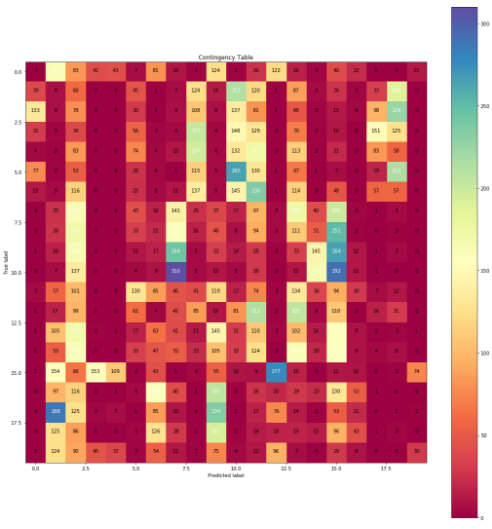


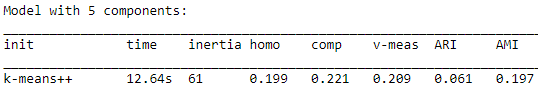


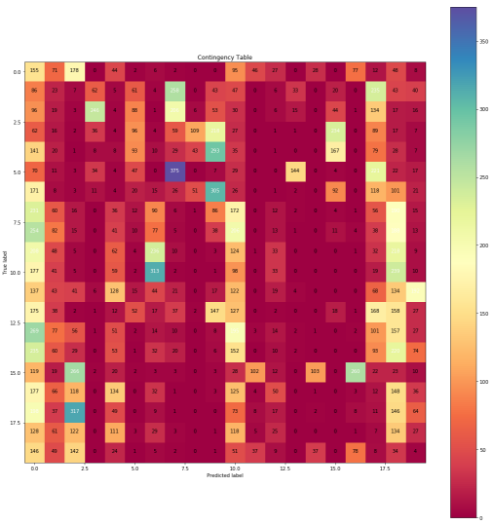


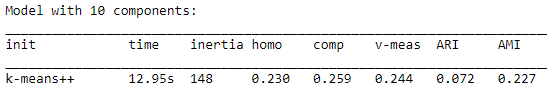


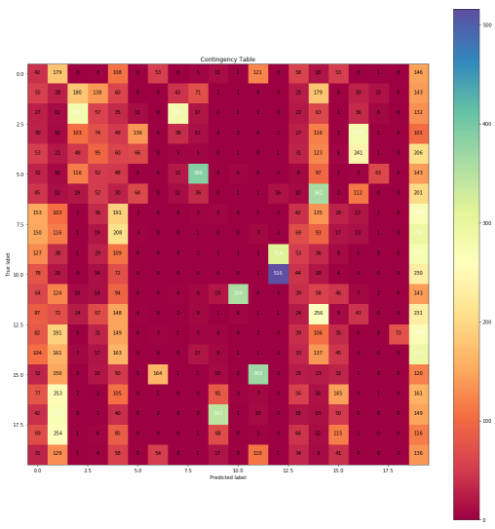


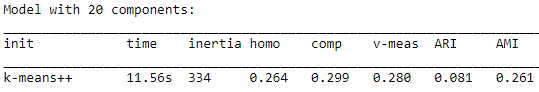


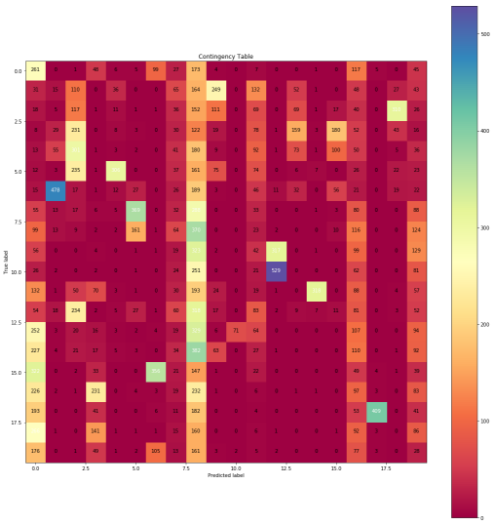


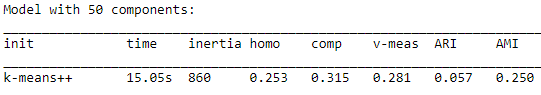


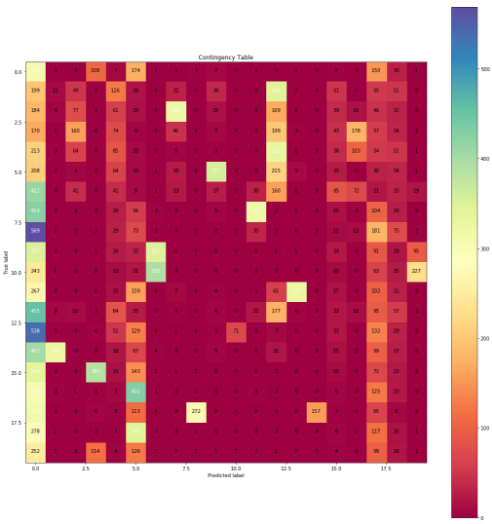


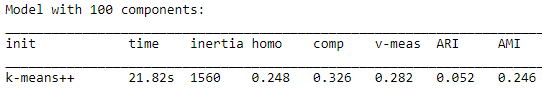


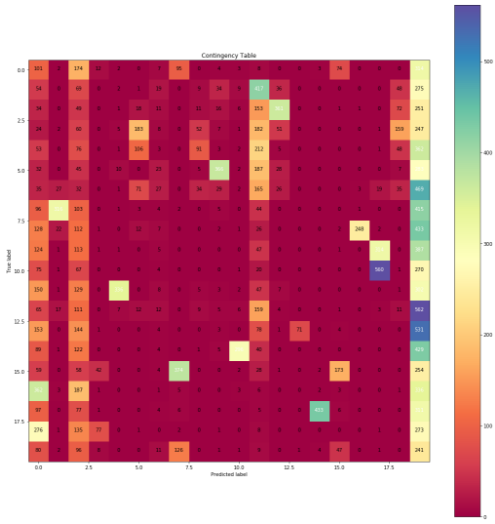


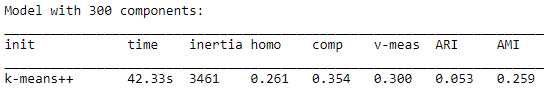


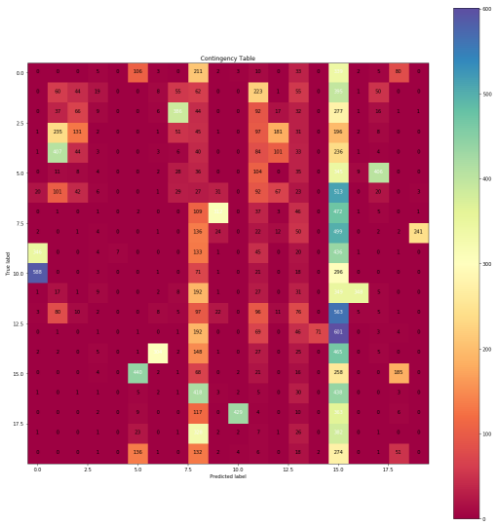


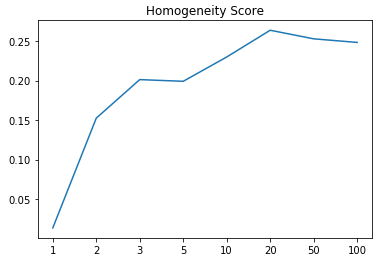
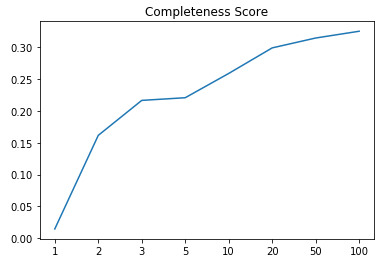
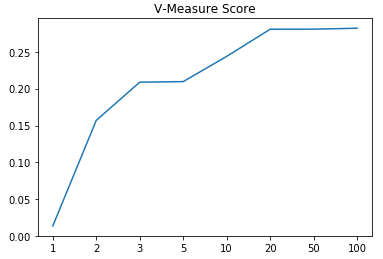
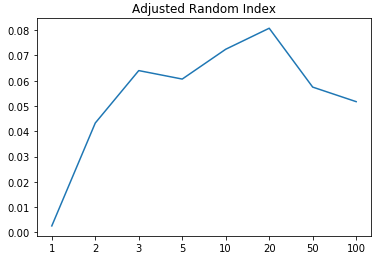
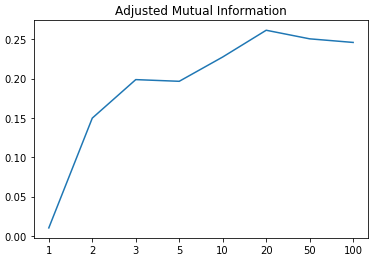








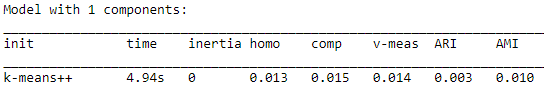


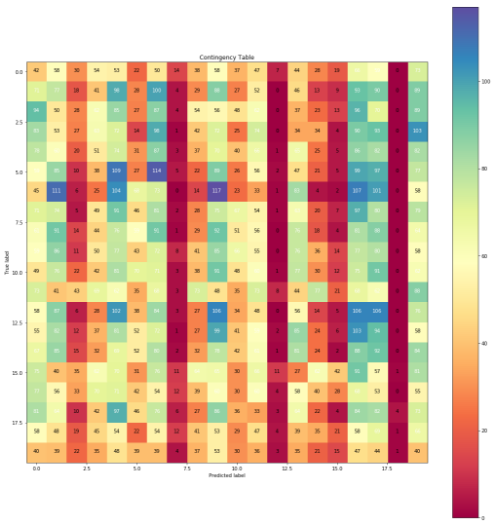
    

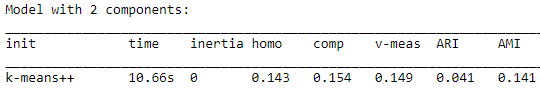
As we can observe from the contingency matrix as well as the 5 measures above, the best r value for LSI is 20. i.e, 20 components are best suitable to represent each document and cluster with maximum accuracy. Although, the graph for completeness score and V-Measure are somewhat monotonic in nature, we consider all the measures and decide 20 to be the best value.

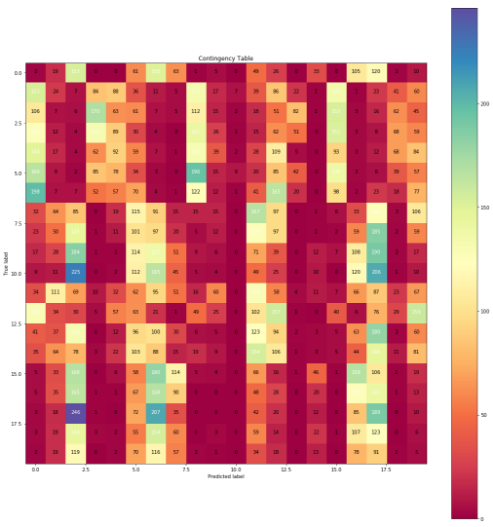
* 1. NMF (Non-Negative Matrix Factorization):

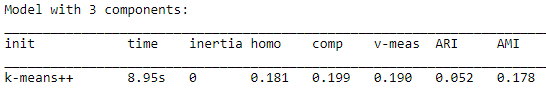
After performing NMF on the TF-IDF matrix, following are the obtained results:

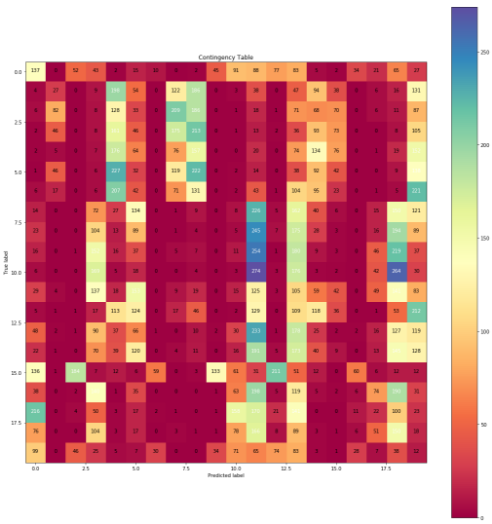


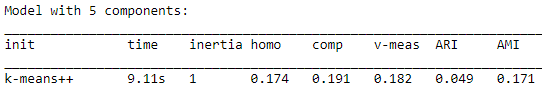


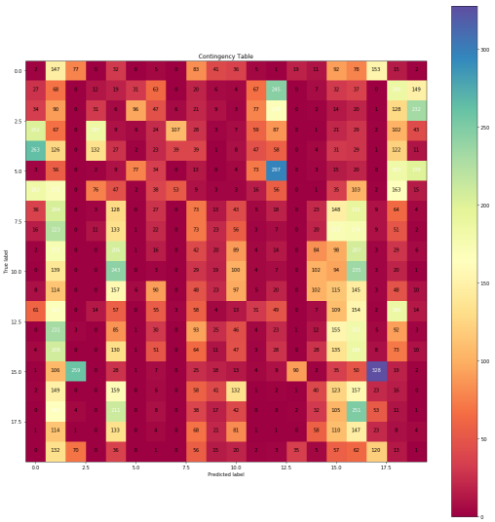


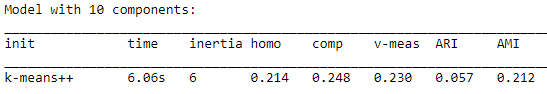


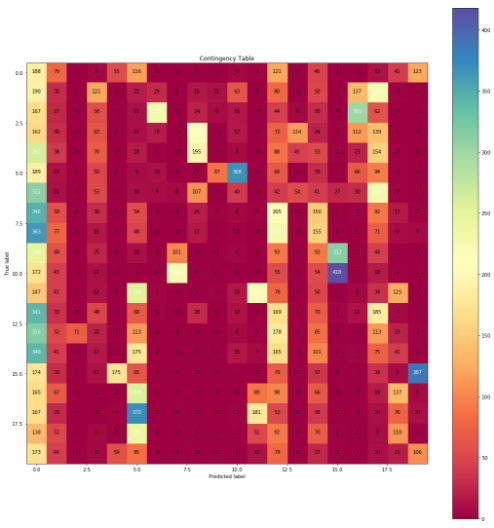


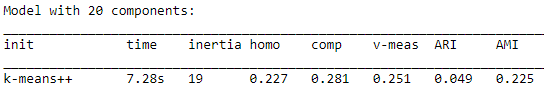


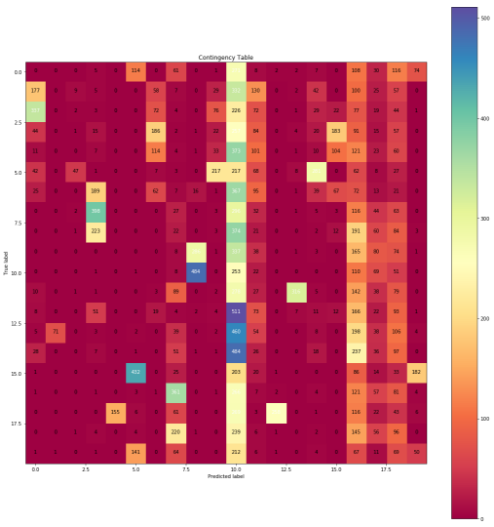


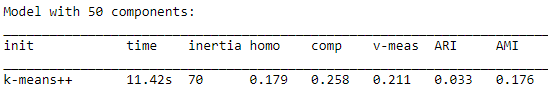


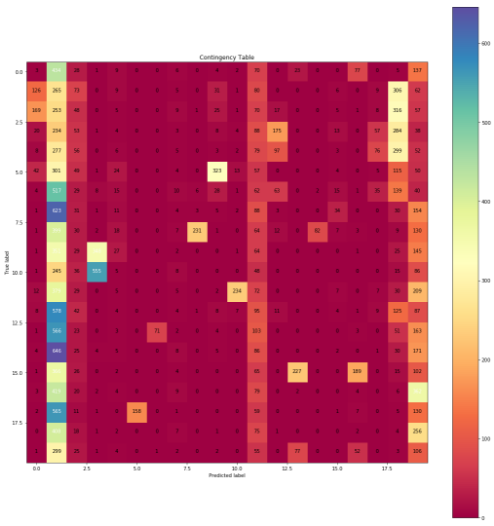


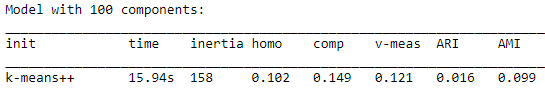


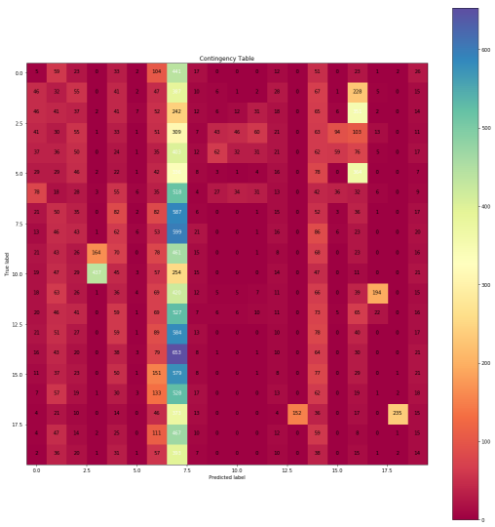


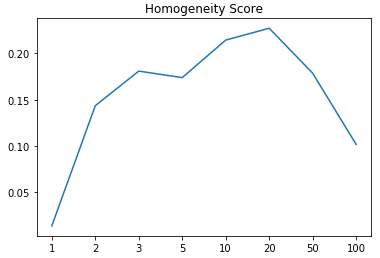
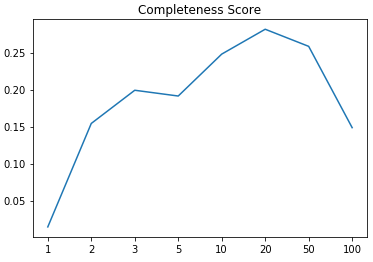
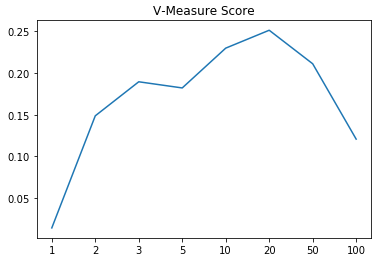
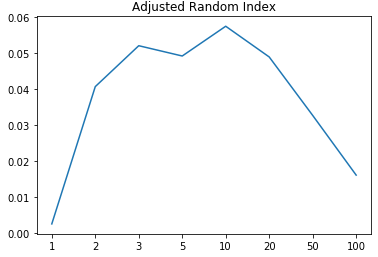
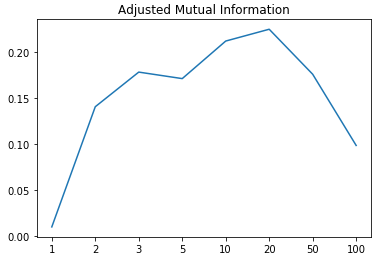










As we can observe, the best r value is at r = 20. i.e, only 20 components/features are more than enough to best cluster the documents in two classes with maximum accuracy.

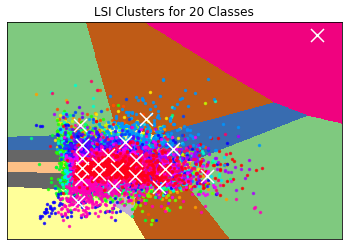
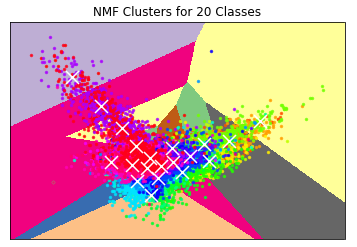
The non-monotonic behavior can be explained by the following reasoning. As we increase the dimension of representation of the data, the data points begin to get close to each other. On the other hand, increasing the dimension of data representation provides intricate details which helps in distinguishing each data point. Hence, a proper balance needs to be found to represent data in best dimension space. Here, we observed that the clustering improved performance as we increase dimension but started performing worse for higher dimensions. Hence, it exhibits the non-monotonic behavior.

1. Visualize the performance:
   1. We found that,

r=5 works best in case of LSI, and

r=2 works best in case of NMF.

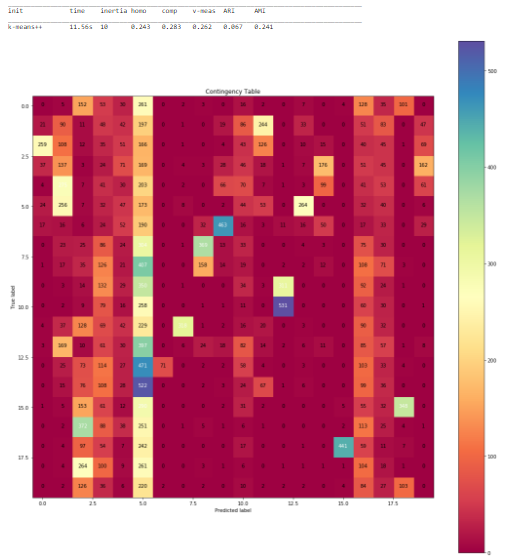
Hence, following are the clusters with their decision boundary for the two methods:

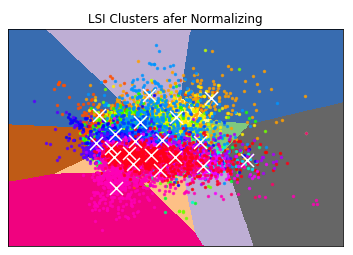
 

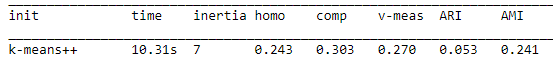
* 1. 3 Methods:
     1. Normalizing:

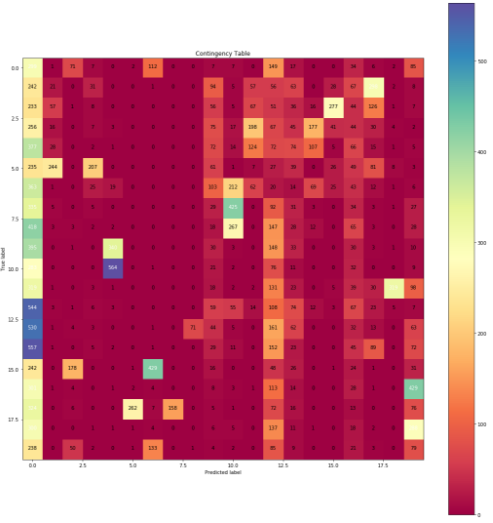
We performed scaling using sklearn “scale” to get unit variance and then normalized the data. Following are the results for LSI and NMF:

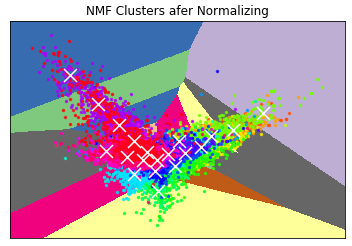




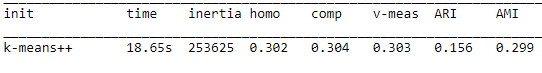


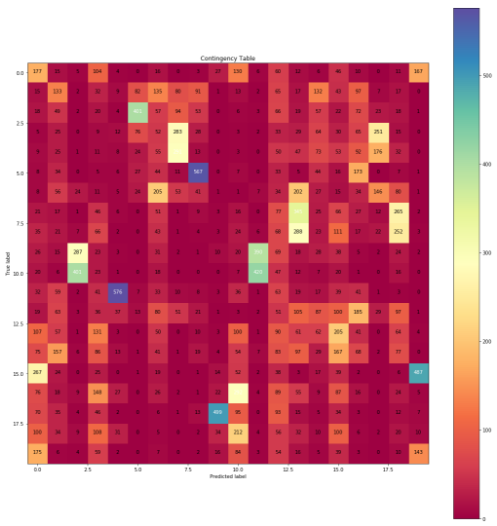


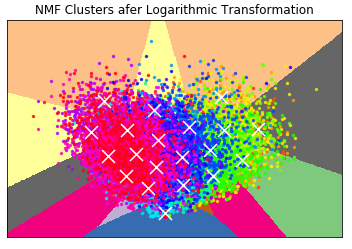




* + 1. Logarithmic Transformation:

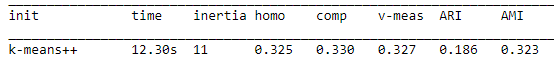


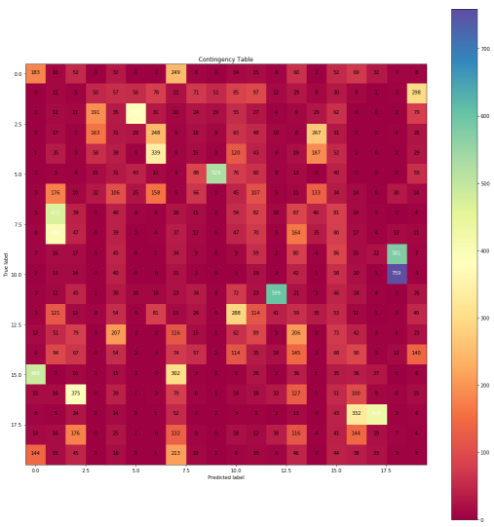


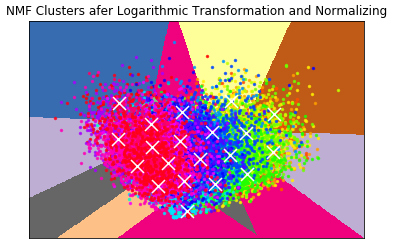


* + 1. Combination
       1. Normalization and then Log Transformation

We perform normalization and then apply log transformation on NMF data to get the following results:







* + - 1. Log Transformation and then Normalization

We apply log transformation on NMF data and then perform normalization to get the following results:

