EE219 Project 4

Clustering
Winter 2017

Anshita Mehrotra - 904743371
Swati Arora – 404758379
Shubham Mittal - 104774903

Introduction:

In this project, we have done data analysis on the 20 Newsgroups dataset which comprises of 20,000 newsgroup documents that are partitioned evenly across 20 newsgroups categories. The dataset is analyzed by using Clustering. Clustering algorithms are unsupervised methods for finding groups of data point that have similar representations in a proper space. As such, K-means clustering iteratively groups data points into regions characterized by a set of cluster centroids. Each data point is then assigned to the cluster with the nearest cluster centroid. In this project the goal is to find proper representations of the data and evaluate the performance of clustering algorithms.

Part 1: Transform the documents into TF-IDF vectors

Since there are lot of common words in each document, the data needs to be preprocessed. For this, first the punctuations are removed, followed by the common stop words. We then find which words share the same stem so that they can be counted together while finding their TF-IDF vectors.
To do the latter, we used a SnowBall stemmer (nlkt) to achieve this.
Once the data has been pre-processed, we move on to finding the TF-IDF vector for each term. For this we convert the document into a set of numerical features. This is done using CountVectorizer. Next we get the TFxIDF Representations using a Transformer. A matrix is obtained with the number of documents (records) as the row and the number of terms obtained as the number of columns.

Number of documents	7882
Number of terms	92773

Part 2: Apply K-means clustering

Once we get a good representation of the data using TF-IDF vectors. We assigned the ground truth value of 1 to “rec” class and 0 to “com” class.
We then apply k-means clustering on the given data where k has been defined as 2. Based on the ground truth values assigned previously, we compare the clustering results.

The best confusion matrix, evaluating how well our clusters match the ground truth labels was achieved to be:

3900	3
2355	1624

In K-means clustering as we define the k centroids for each cluster at the start randomly. Changing the permutation of the rows for the data did not help improve the different performance measures like homogeneity, completeness etc. The values that were obtained were very close. But there was no significant difference in the values. **The permutation of the rows did not make the confusion matrix look almost diagonal.**
It is the placement of the centroids that should impact the performance and not the permutation of the different rows in the data. We lopped over 100 iterations, each time randomly picking the cluster center but didn’t observe significant changes in the confusion matrix.

The other different measures of purity in the partitioning of the data points with respect to the ground truth were obtained as follows:

Measure	Values
Homogeneity Score	0.2256
Completeness Score	0.3144
Adjusted rand score	0.1476
Adjusted mutual info score	0.225

Fig: Clustering performance measure without dimensionality reduction

As we can observe, the clustering performance here is not very good. In the next part, we will try improving upon the performance for the clustering algorithm.

Part 3: Dimensionality Reduction

To improve upon the performance of the clustering algorithm, we need to reduce the dimensions of the data properly. For this we used Latent Semantic Indexing(LSI)/ Truncated SVD and Non-Negative Matrix Factorization(NMF).

a) We first use **SVD** to perform dimensionality reduction. To get an initial guess on the approximate dimensions we check the top singular values of the TF-IDF matrix. As per our observation, the initial guess is around 6.
Then starting with the number of dimensions as 2, we go up to 20. To find the dimensionality which yields better results in terms of the clustering purity values.

Based on the adjusted_rand_score and adjusted_mutual_info, we could conclude that the **best clustering was performed when the number of dimensions are 14**. Note :
1) this was after normalization of the data.
2) even when the **number of dimensions were 4, we got comparable results.**

The following results were obtained:

Measure	Values
Homogeneity Score	0.7865
Completeness Score	0.7869
Adjusted rand score	0.8676
Adjusted mutual info score	0.7865

Fig: Clustering performance with dimensionality reduction using SVD and normalization
No of dimensions: 14

Confusion Matrix	
175	3728
3884	95

Fig: Confusion matrix [dimensionality reduction using SVD, with normalization of data]
Number of dimensions: 14

Measure	Values
Homogeneity Score	0.7341

Completeness Score	0.7359
Adjusted rand score	0.8201
Adjusted mutual info score	0.7340

Fig: Clustering performance with dimensionality reduction using SVD and normalization No of dimensions: 4

Confusion Matrix	
285	3618
3892	87

Fig: Confusion matrix [dimensionality reduction using SVD, with normalization of data] Number of dimensions: 4)

Without the normalization of data, the performance improvement was not significant where the average values were as given below:

Measure	Values
Homogeneity Score	0.2197
Completeness Score	0.3092
Adjusted rand score	0.1420
Adjusted mutual info score	0.2196

Fig: Clustering performance with dimensionality reduction using SVD and without normalization of data

Confusion Matrix	
3899	4
2451	1528

Fig: Confusion matrix [dimensionality reduction using SVD, without normalization of data]

As we can see, there was improvement in the adjusted mutual score. However, the other performance parameters are almost the same.

b) We then used **NMF** to perform dimensionality reduction. Starting with the number of dimensions as 2, we go up to 20. To find the dimensionality which yields better results in terms of the clustering purity values.

Based on the adjusted_rand_score and adjusted_mutual_info, we could conclude that the **best clustering was performed when the number of dimensions are 3**. Note this was after normalization of the data.
The following results were obtained:

3) Measure	Values
Homogeneity Score	0.7713
Completeness Score	0.772
Adjusted rand score	0.854
Adjusted mutual info score	0.771

Fig: Clustering performance with dimensionality reduction using NMF and normalization. No of Dimensions: 3

Confusion Matrix	
3702	201
96	3883

Fig: Confusion matrix [dimensionality reduction using SVD, with normalization of data] No of dimensions: 3

Without the normalization of data, the performance of the clustering algorithm was not very good, where the average values were as given below:

Measure	Values
Homogeneity Score	0.0802
Completeness Score	0.2033

Adjusted rand score	0.0267
Adjusted mutual info score	0.0801

Fig: Clustering performance with dimensionality reduction using NMF and without normalization of data

Confusion Matrix	
611	3292
3	3976

Fig: Confusion matrix [dimensionality reduction using NMF, without normalization of data]

c) We then plot the points, to get a visual sense on the embedding of data and to choose the appropriate non-linear transformation.

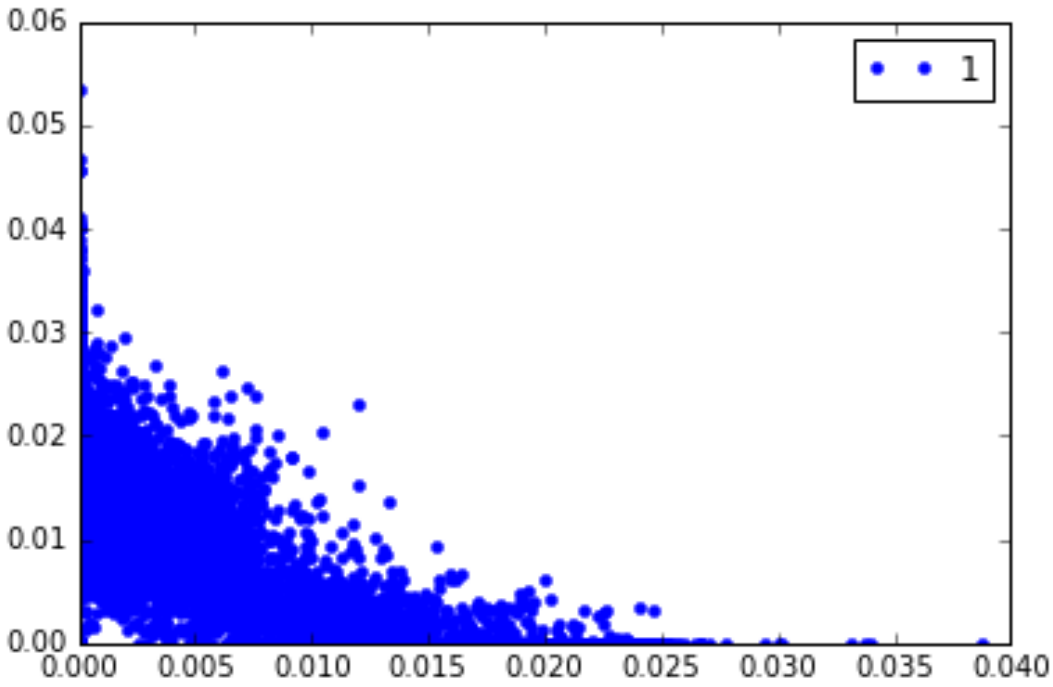
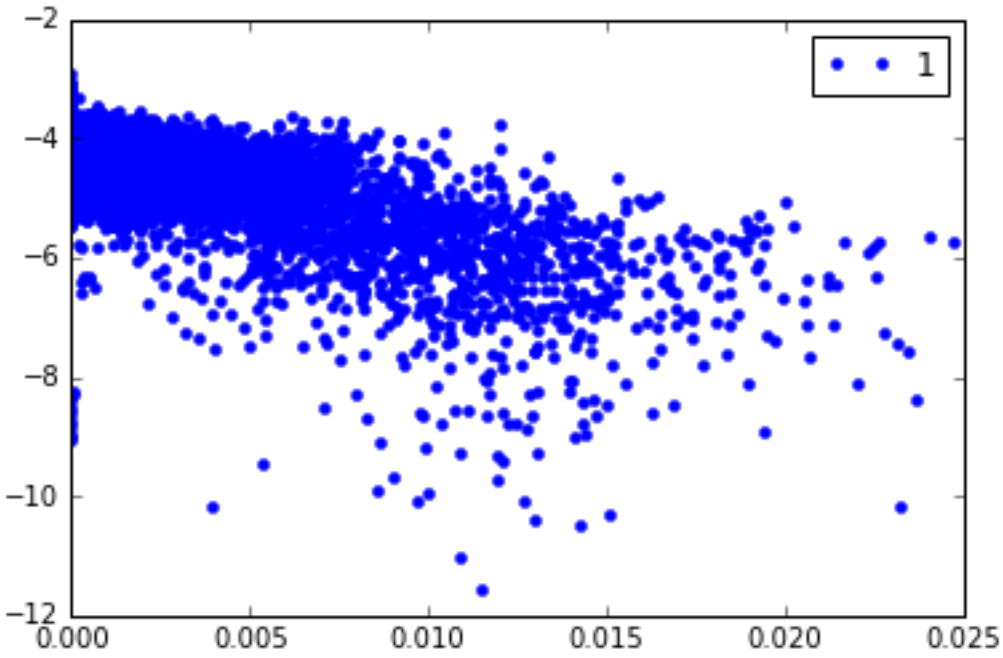
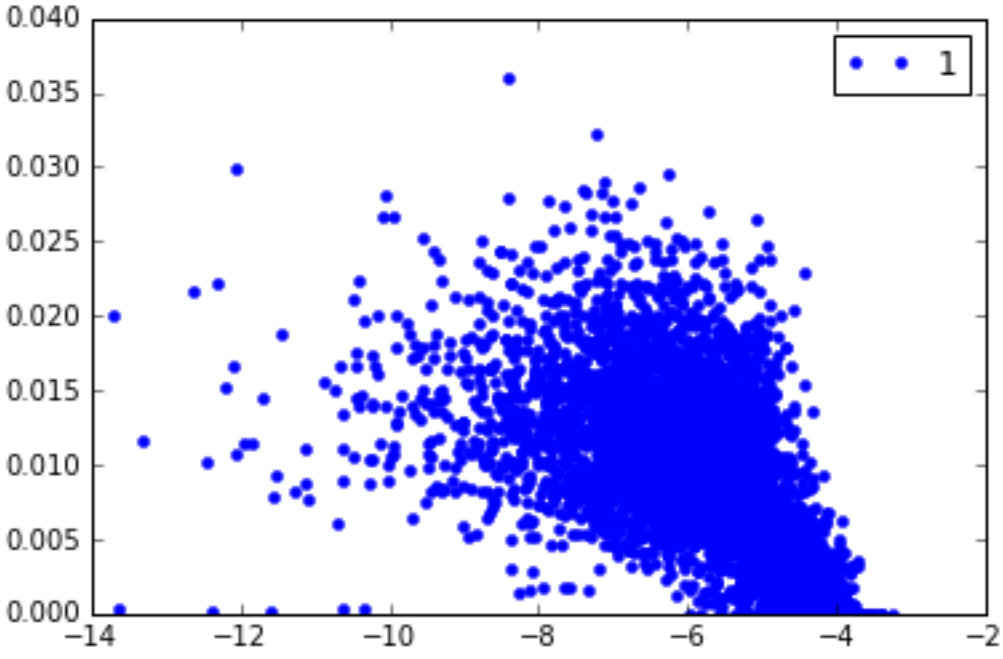


Fig: Plot for the resulting data points.

As we can observe, many points are cramped near the origin. But k-means need a globular scattering of the points.

So, we tried applying a nonlinear transformation on our TF-IDF vectors, to help overcome this problem. As, TFIDF is proportional to the frequency of the terms. We use a logarithmic function. The plots obtained were as follows:





Part 4: Performance visualization for Clustering

To help visualize the performance of your clustering we perform the following steps.

- Perform Dimensionality reduction using NMF. Where the number of dimensions are 5.
- We then use PCA (Principal Component Analysis) to project the results onto two vectors.
- This output was then fed in to K means algorithm for clustering.
- Note we use a nonlinear transformation on our TF-IDF vectors to ensure globular scattering of the points. We use a logarithm function, as TFIDF is proportional to the frequency of the terms.
- We also plotted the confusion matrix, wherein we labeled the different cells with different colors i.e. TP, FP, TN, FN with different labels.

  The following plot was obtaining on plotting the results:

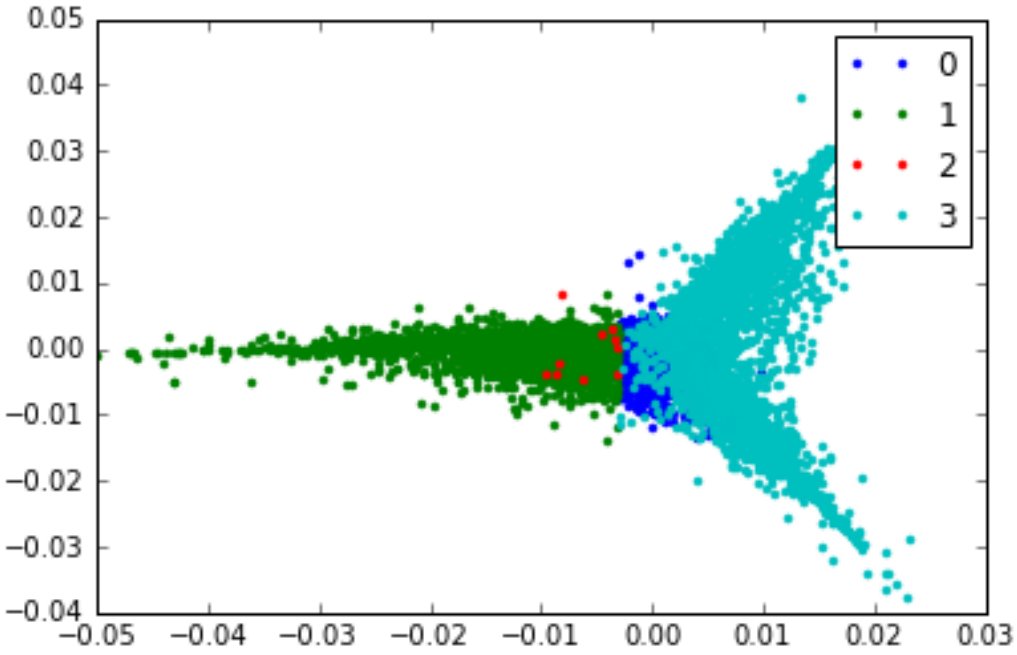


Fig: Plot showing the confusion matrix for the clustering

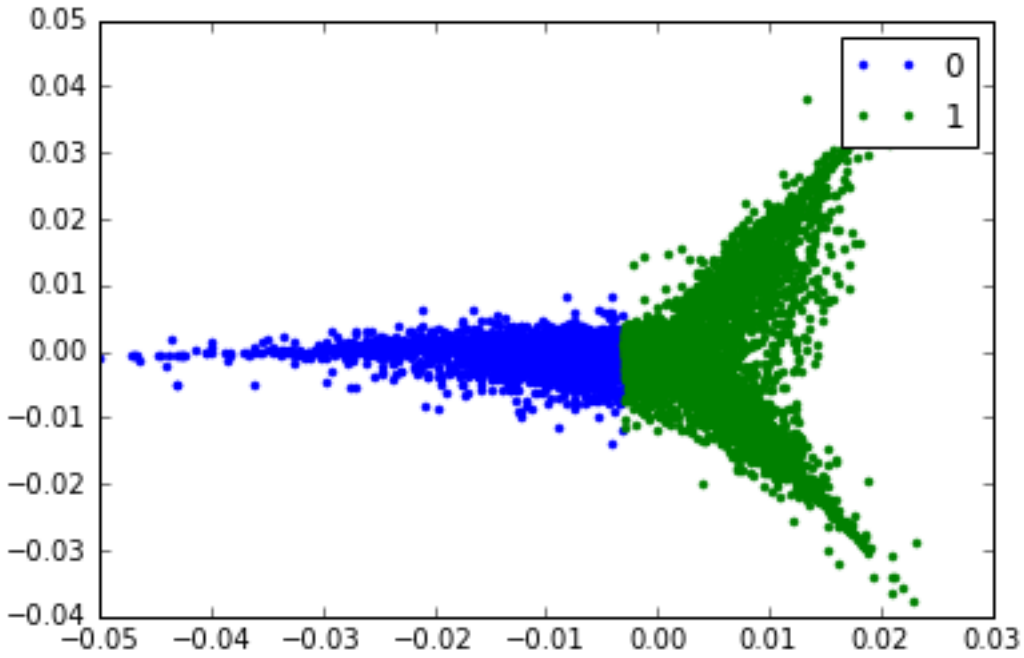


Fig: Plot showing the clustering of the two classes

Confusion Matrix	
2594	1309
10	3969

Fig: Confusion matrix for the plots above

- We can observe from the first plot, that most of the misclassification is at boundary of the two classes.

Part 5: Clustering for all 20 original Sub Classes

In this part we performed clustering for all the original 20 sub classes. To improve upon the performance of the clustering algorithm, we need to reduce the dimensions of the data properly. For this we used Latent Semantic Indexing(LSI)/ Truncated SVD and Non-Negative Matrix Factorization(NMF).
We get the proper representation of data using a TF-IDF vector. Where the rows represent the documents and the columns are the terms.

Number of documents	18846

Number of terms	151006

a) We first use **SVD** to perform dimensionality reduction. Starting with the number of dimensions as 2, we go up to 20. To find the dimensionality which yields better results in terms of the clustering purity values.

Based on the adjusted_rand_score and Homogenity_score, we could conclude that the **best clustering was performed when the number of dimensions are 17**. Note that the data that we used has been normalized.

Results for different dimensions using SVD is as follows:

Dimension	**Homogeneity Score**	**Adjusted Rand Score**
2	0.2378	0.07025
3	0.3215	0.1327
4	0.3716	0.1733
5	0.3996	0.1993
6	0.3868	0.1914
7	0.3830	0.1926
8	0.4036	0.2199
9	0.4242	0.2314
10	0.4358	0. 2570
11	0.4279	0.2412
12	0.4207	0.2470
13	0.4318	0.2502
14	0.4310	0.2587
15	0.4312	0.2628
16	0.4348	0.2581
17	0.4353	0.2672
18	0.4218	0.2431
19	0.4227	0.2473
20	0.4268	0.2601

Fig: Clustering Performance for different dimensions using SVD(with normalization of data)

Measure	**Values**
Homogeneity Score	0.4262
Completeness Score	0.4538

Adjusted rand score	0.2572
Adjusted mutual info score	0.4243

Fig: Clustering Performance for SVD when the number of dimensions are 17 (with normalization of data)

b) We then used **NMF** to perform dimensionality reduction. Starting with the number of dimensions as 2, we go up to 20. To find the dimensionality which yields better results in terms of the clustering purity values.

Based on the Adjusted_rand_score and Homogenity_score, we could conclude that the **best clustering was performed when the number of dimensions are 18**. Note this was after normalization of the data.
Results for different dimensions using SVD is as follows:

Dimension	Homogeneity Score	Adjusted rand score
2	0.1897	0.0713
3	0.2631	0.1230
4	0.2840	0.1182
5	0.3394	0.1542
6	0.3668	0.1904
7	0.3385	0.1584
8	0.3568	0.1772
9	0.3719	0.2026
10	0.3569	0.1838
11	0.3799	0.2050
12	0.3630	0.1915
13	0.4149	0.2312

14	0.3680	0.2006
15	0.3703	0.1919
16	0.4021	0.2277
17	0.3952	0.2117
18	0.4157	0.2558
19	0.4108	0.2506
20	0.4081	0.2546

Fig: Clustering Performance for different dimensions using NMF(with normalization of data)

Measure	Values
Homogeneity Score	0.4161
Completeness Score	0.4408
Adjusted rand score	0.2565
Adjusted mutual info score	0.4143

Fig: Clustering Performance for NMF when the number of dimensions are 18 (with normalization of data)

c) We then tried to find an appropriate value for the parameter k to use in the kMeans clustering algorithm to find the pure clusters with respect to the class labels.
We then varied the value of k from 2 to 100 and plotted the homogeneity score and completeness score to get an idea about the performance as we vary the clusters.
The plot for values of k from 2 to 100 is as follows:

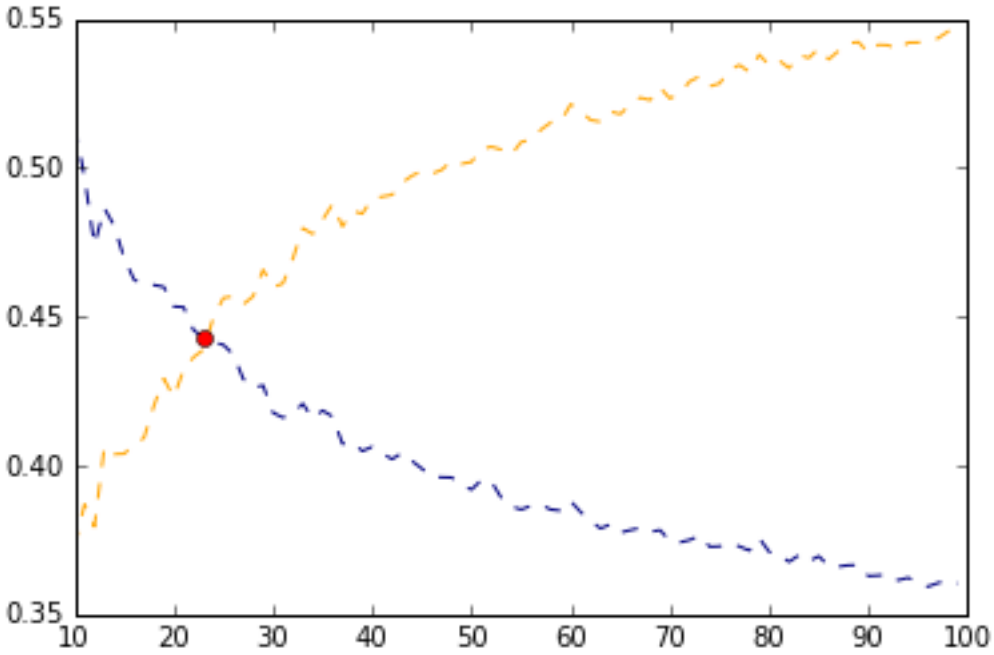


Fig: Plot for the Completeness score (in navy) and Homogeneity score(in orange) as we vary the number of clusters from 2 to 100.

As the optimal value lies between 20 and 30 for both the completeness score and homogeneity score, we then varied the value for k from 20 to 30 and plotted the graph.

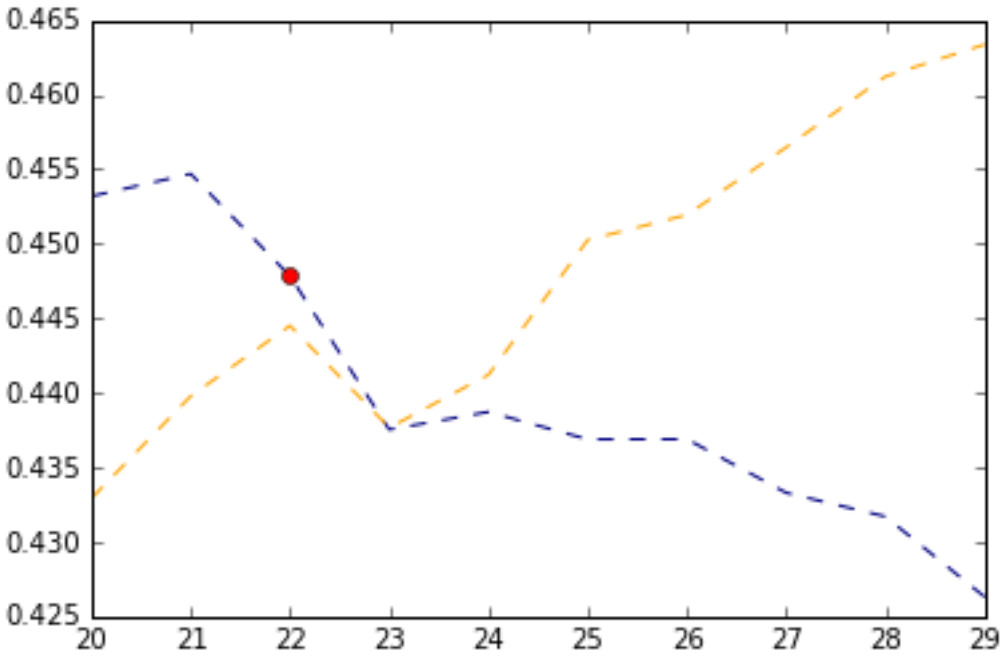


Fig: Plot for the Completeness score (in navy) and Homogeneity score (in orange) as we vary the number of clusters from 20 to 30.

As we can observe from the graph above the **optimal value for k is 22**. Where both, the completeness score and the homogeneity score is high. Therefore, we select k = 22 as the value for pure clustering. The clustering performance measures for k =22 are as follows:

Measure	Values
Homogeneity Score	0.4378
Completeness Score	0.4464
Adjusted rand score	0.2617
Adjusted mutual info score	0.4357

Fig: Clustering Performance when k = 22

Part 6: Clustering Performance when retrieving the topic-wise classes

In this part, we evaluated the performance of clustering in retrieving the topic-wise classes reduction and feature transformation. We first find a proper representation of our data using TF-IDF vectors. The number of clusters in the kMeans clustering algorithm is fixed at 6.

We first use **SVD** to perform dimensionality reduction. Starting with the number of dimensions as 2, we go up to 20. To find the dimensionality which yields better results in terms of clustering all the documents into 6 clusters as per 6 major classes.

Based on the adjusted_rand_score and Homogenity_score, we could conclude that the **best clustering was performed when the number of dimensions are 7**. Note that the data that we used has been normalized.

The plot for values of components from 2 to 100 is as follows:

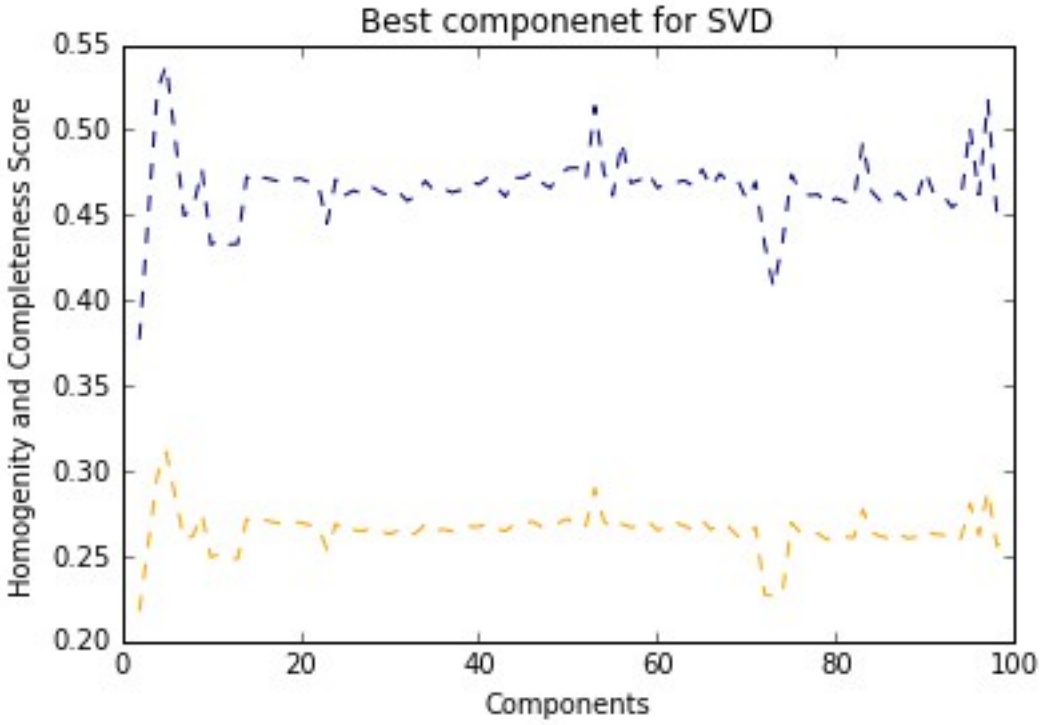


Fig: Plot for the Completeness score (in navy) and Homogeneity score(in orange) as we vary the number of components from 2 to 100.

The plot for values of components from 2 to 10 is as follows:

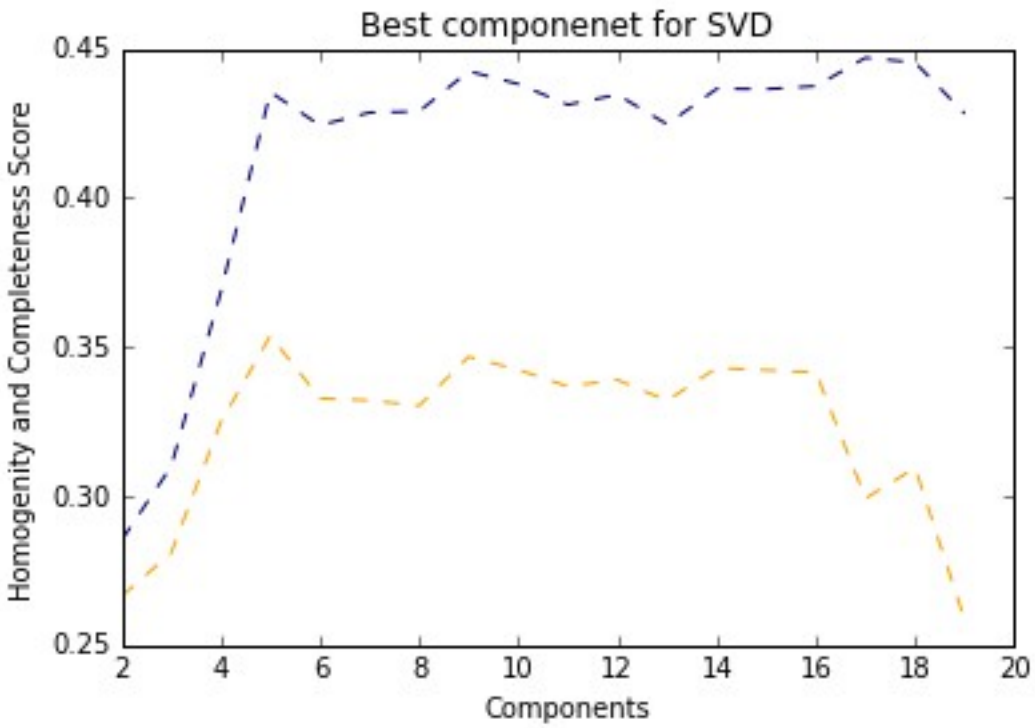


Fig: Plot for the Completeness score (in navy) and Homogeneity score(in orange) as we vary the number of components from 2 to 10.

As we can observe from the graph above the **optimal value for component is 7**. Where both, the completeness score and the homogeneity score is high. Therefore, we select component = 7 as the value for clustering. The clustering performance measures for k =6 are as follows:

Measure	Values
Homogeneity Score	0.3321
Completeness Score	0.4286
Adjusted rand score	0.1526
Adjusted mutual info score	0.3318

Confusion Matrix					
1246	0	34	31	2668	0
1	1211	9	130	1072	1
6	1	2970	2	1905	7
1	17	0	1334	1268	5
2	7	318	115	2955	555
21	0	0	0	225	729

Fig: Confusion matrix for the clustering above