EE 219 PROJECT 4

CLUSTERING

TEAM MEMBERS
Omkar Patil : - 904760474
Shikhar Malhotra :- 504741656
Twinkle Gupta :- 804740325

Clustering is the task of grouping a set of objects in such a way that objects in the same cluster are more similar to each other than to those in other clusters. Clustering is an unsupervised learning task since the training labels are not known before hand. In this project, we use K-means clustering algorithm. 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, we intend to find the most optimal representation of data for K-means and evaluate the performance of the algorithm using various purity measures.

We implemented the project in Python. For performing K-means, we used the sci-kit learn library.

DATASET USED

In this project, we use the 20 Newsgroup dataset. The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.
The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale/ soc.religion.christian). In order to define the clustering task, we assume the data to be unsupervised, i.e. we assume that class labels are not available to us. We then aim to find document clusters, where documents in each group are more similar to each other than to those in other group. We then use class labels as ground truth to evaluate the performance of the clustering task.

PART 1 : CONSTRUCTING TF X IDF MATRIX

To preprocess the data, we first remove punctuations, common stop words and finding which words share the same stem so that they can be counted together while finding their TF-IDF.
In order to do the latter we used a SnowBall stemmer from the nltk Python library.

Once the data has been pre-processed, the next step is to find the TFIDF of each term. TFIDF (term frequency-inverse document frequency) is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. For this we convert the document into a set of numerical features.

For this problem,

- The documents in the data set are turned into numerical feature vectors.
- The words here are tokenized and unwanted tokens like punctuation marks, stop words, stemming words etc are removed for preprocessing the data.
- After this a TFxIDF vector representation is created and its terms are reported as

Total number of terms are 54433

PART 2 K-MEANS CLUSTERING

In this problem, k-means clustering was applied to the data points with k = 2.
To find the approximate ground truth cluster labels, we assigned the cluster value 0 to labels starting with 'com.' if most data points with belonging to these labels were clustered in cluster 0. And similarly assignments were done for cluster 1. The algorithm was run 10 times and error percentage, homogeneity score, completeness score, adjusted rand score, the adjusted mutual info score were found and the following results were obtained-

S.NO	ERROR PERCENTAGE	HOMOGENEITY SCORE	COMPLETENESS SCORE	V MEASURE	ADJUSTED RAND INDEX	ADJUSTED MUTUAL INFO SCORE
1	9.974640744	0.573764518	0.594947515	0.292524908	0.157788235	0.573699506
2	10.92561285	0.55144346	0.576115765	0.281936775	0.149652414	0.551375044
3	11.26373626	0.544824275	0.571499802	0.279081389	0.147022834	0.54475485
4	10.31276416	0.567192503	0.591055866	0.290015443	0.154919112	0.567126489
5	10.94674556	0.549115928	0.573092584	0.280589176	0.1496322	0.549047157
6	11.07354184	0.545328906	0.571657049	0.279781218	0.148870733	0.545259557
7	8.664412511	0.618149816	0.631310091	0.313030026	0.173969349	0.618091575
8	8.114961961	0.618149816	0.631310091	0.313030026	0.173969349	0.618091575
9	10.81994928	0.555986875	0.581981277	0.284780244	0.150790407	0.555919152
10	9.129332206	0.591580161	0.610618925	0.301611671	0.165528265	0.591517867
AVG	**10.12256974**	**0.571553626**	**0.593358897**	**0.291638088**	**0.15721429**	**0.571488277**

On average, an error of 10.12 % was obtained and homogeneity score of 0.5716 was obtained. The adjusted rand score obtained was 0.1572, which is very low.

Since the K-means algorithm chooses random initial centers for the clusters, changing permutations of rows also does not help much. The error values obtained for different permutations were different, but close. There were no permutations found for which the confusion matrix was most diagonal for all 10 iterations of the k-means algorithm. It is the placement of the centroids that impacts the performance, and not the permutation of rows.

PART 3- DIMENSIONALITY REDUCTION

It was observed that high dimensional sparse TF-IDF vectors do not yield a good clustering Performance, especially in terms of Adjusted Rand Score. In this part we try to find a representation of data that improves the performance of clustering algorithm. For this., we reduce the dimensions of the TF-IDF matrix using dimensionality reduction.
We use Latent Semantic Indexing(LSI) and Non-negative Matrix Factorization(NMF).

a) For dimensionality reduction, Singular value decomposition was applied. The values of the number of components was varied from 2 to 20 and the best estimate was found by observing the sum of homogeneity and adjusted rand score measures. The dimension with highest sum was chosen.

For SVD, the optimal dimension was found to be 15.

The various purity measures obtained at number of components = 15 were :
Homogeneity: 0.574988387963
Adjusted Rand-Index: 0.645483468296
Completeness: 0.586864049464
V-measure: 0.580865526266
Mutual Information: %0.3f 0.574923563421
Explained variance of the SVD step with 15 components: 5%
Confusion Matrix :
[[1924 419]
[46 2343]]

b) Similarly for NMF, the values of number of components was varied from 2 to 20 and the **best estimate was found to be 2.**
The various purity measures obtained are :
Homogeneity: 0.507815489576
Adjusted Rand-Index: 0.573561228146
Completeness: 0.522826833076
V-measure: 0.515211840939
Mutual Information: 0.507740419148
Explained variance of the NMF step with 2 components: 6%
Confusion Matrix :

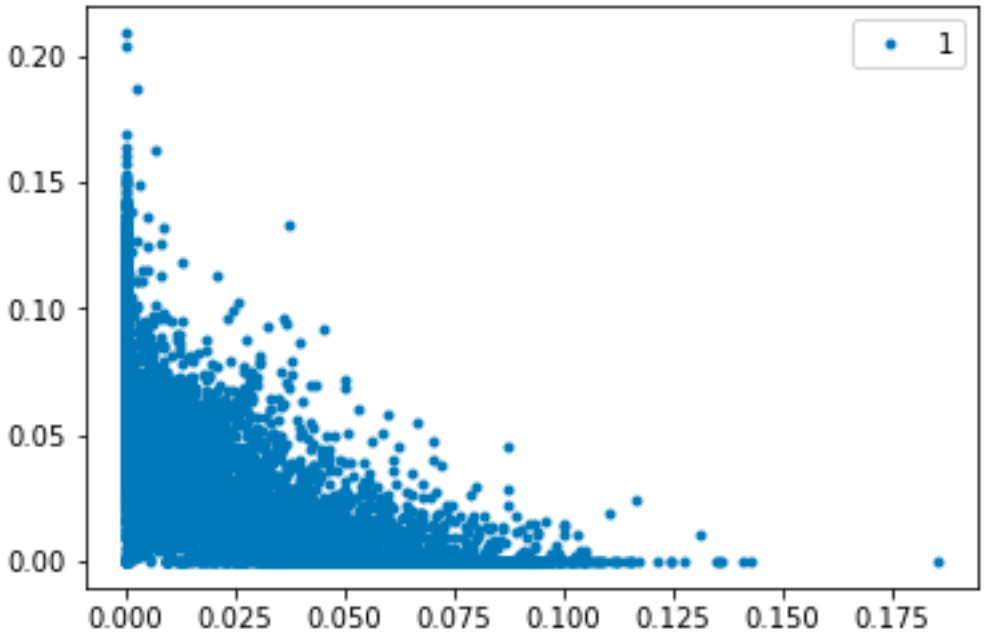
[[1832 511]
[63 2326]]

c) We also applied normalization, and similarly varied the number of dimensions for 2 to 20 and deduced best estimate for dimension by observing the purity measures.

The best dimension was found to be 3.

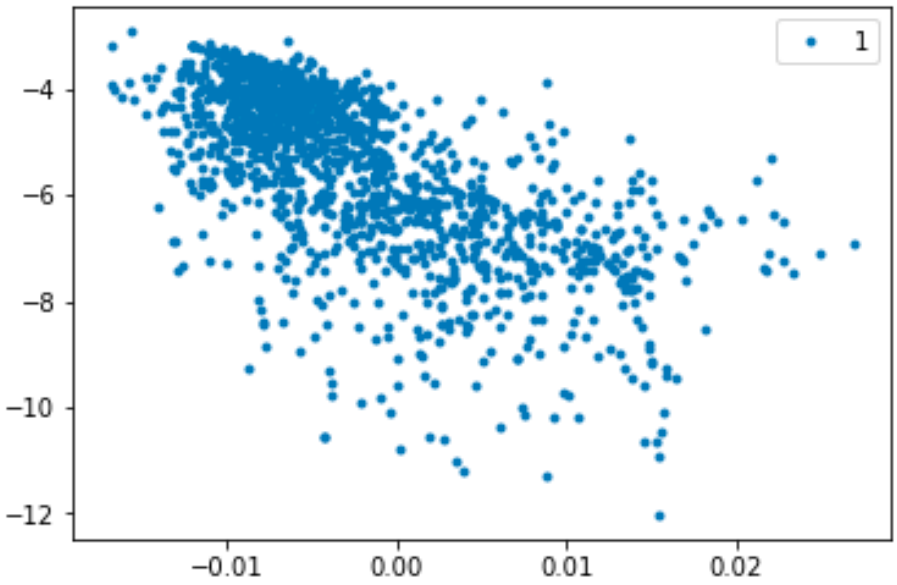
The various purity measures obtained are :
Homogeneity: 0.640058826879
Adjusted Rand-Index: 0.741902719283
Completeness: 0.640408140555
V-measure: 0.64023343607
Mutual Information: 0.640003927801
Explained variance of the NMF step with 3 components: 6%
Confusion Matrix :
[[2225 118]
[210 2179]]
We observed no significant increase in the purity measures after normalization.

d) To observe the distribution of data points, the dimension of the TF-IDF matrix was reduced to 2 using NMF and the points were plotted in 2D space. The following plot was obtained.

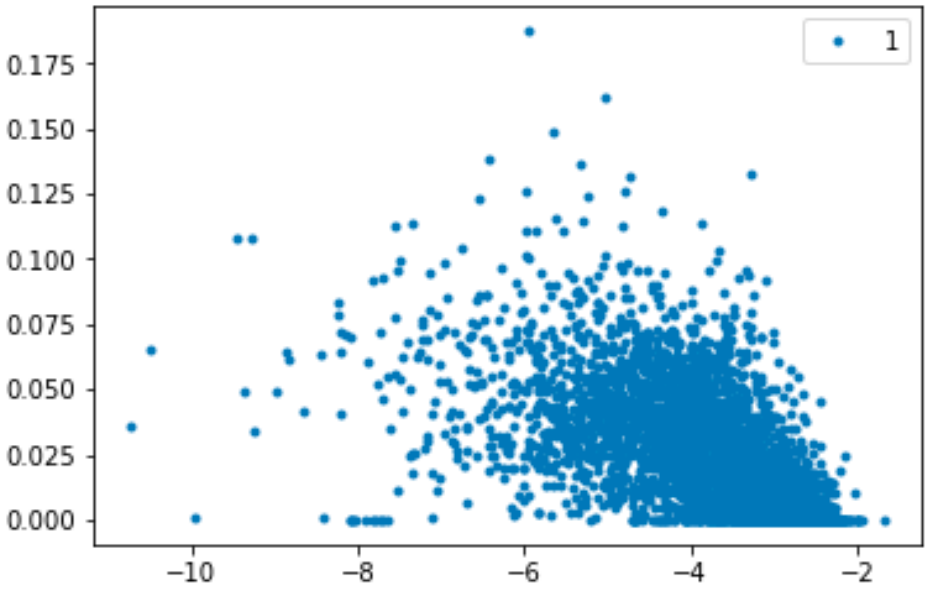


It was observed that most of the data points were concentrated near the origin. However, for K-means, we need a more globular spread of data for higher accuracies and purity measures. Therefore, a non-linear transformation of data makes sense.

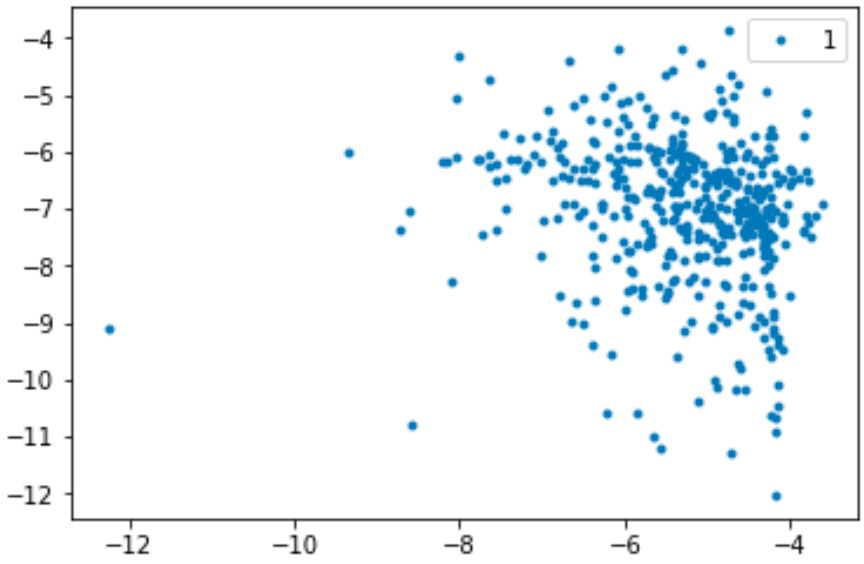
To justify this, the data in one dimension (here the vertical dimension, or the y- dimension) was transformed using log and the points were plotted again. The following plot was obtained-



Similarly, if log transformation was done on the x-dimension, the following plot was obtained.



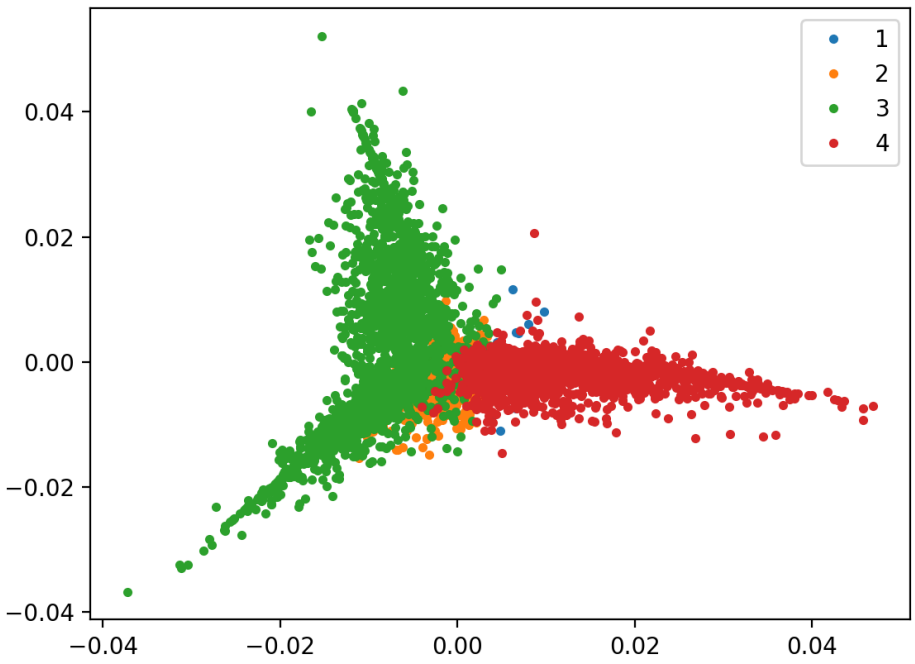
Similarly, when log transformation was applied on both the dimensions, the following plot was obtained.



It was observed with non-linear logarithmic transformation; the data points were spread in the 2D space.

PART 4 : PLOTTING LABELS

Log transformation was applied to the reduced data and fed into k-means algorithm.
To project data points into 2-dimensional space, PCA (Principal Component Analysis) was applied, and the resulting reduced dimension values were fed into the k-means algorithm.
True and false positives and negatives were plotted with different colors to visually evaluate the performance of the algorithm.
The following results were obtained.
Confusion Matrix :
[[1666 677]
 [33 2356]]


1
2
3
4
0.04
0.02
0.00
−0.02
−0.04
−0.04
−0.02
0.00
0.02
0.04