EE219 Project 2

Clustering

Winter 2020

01/30/2020

Chenyu Wang **UID 805436446**

Gaofang Sun **UID 104853165**

Gongjie Qi **UID 805429380**

Hao Xu **UID 305429561**

Introduction

Clustering algorithms are unsupervised methods for finding groups of data points that have similar representations in a feature space. Clustering differs from classification in that no a priori labeling (grouping) of the data points is available. In this project, we learn to use clustering to find proper representation of the data. Then we performed K-means clustering on the dataset and evaluated the result. We also tried different preprocessing methods which may increase the performance of the clustering.

I. Clustering of Text Data

In this project, we work with the 20 newsgroups dataset. It comprises around 20,000 newsgroups posts on 20 topics. Size balance in each category of dataset is of great importance to an effective learning algorithm. In case of imbalance in the relative sizes of the datasets with different categories, we would need to modify the penalty function or down-sample the majority classes. Checking if they are balanced is very important.

Class 1 Computer Technology : 'comp.sys.ibm.pc.hardware', 'comp.graphics', 'comp.sys.mac.hardware', 'comp.os.ms-windows.misc',
Class 2 Recreational Activity : 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey'

1. Building the TF-IDF matrix.

Question 1: Dimensions of the TF-IDF

We removed stopwords and symbols from the text before vectorization. The shape of the TF-IDF data is as following:

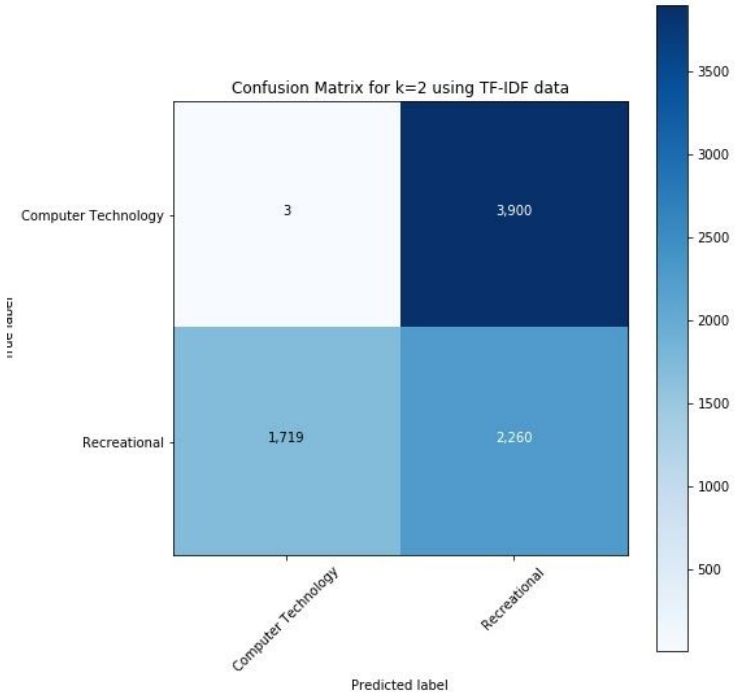
(7882, 28776)
meaning 7882 documents and 28776 terms per document.

2. Apply K-means clustering

In this part, we applied K-means clustering with k = 2 using the TF-IDF data. We got the contingency table, and evaluated the clustering results using various measures for a given partition of the data points with respect to the ground truth. The measures include homogeneity score, completeness score, V-measure, adjusted Rand score and adjusted mutual info score.

Question 2:

Contingency table of our clustering result:



Question 3:
5 measures for the K-means clustering results we got:
Homogeneity: 0.2548
Completeness: 0.3364
V-measure: 0.2900
Adjusted Rand-Index: 0.1812
Adjusted Mutual Info: 0.2548

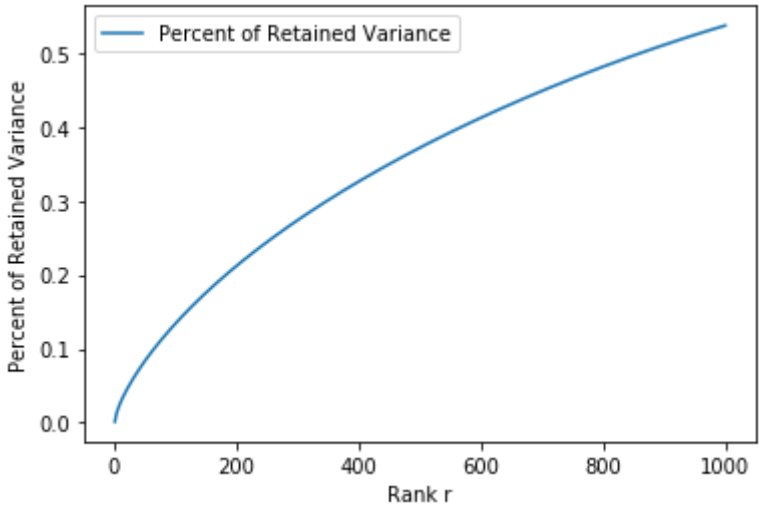
3. Dimensionality reduction

High dimensional sparse TF-IDF vectors do not yield a good clustering result. K-means clustering may fail to identify the clusters properly in some situations, because these limitations, in this part we try to find a "better" representation tailored to the way that K-means clustering algorithm works, by reducing the dimension of our data before clustering. We used Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) for dimensionality reduction.

(a) To find the effective dimension, we examine the ratio of the variance of the original data. Then we calculate the variance of the top 1000 principal components and plot the percent of variance the top r principal components can retain v.s.r, for r = 1 to 1000.

Question 4:

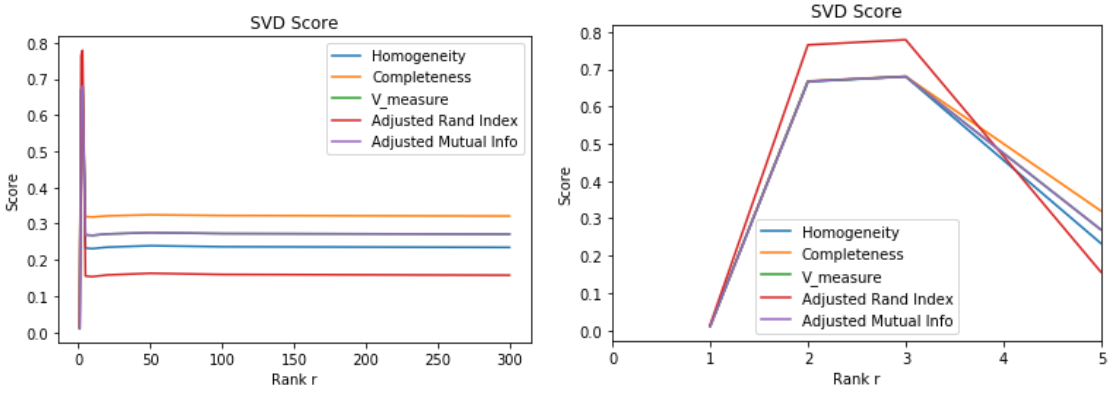
We plotted the percent of variance the top r principle components can retain v.s.r, from r = 1 to 1000, which is shown below. The graph shows that the percent of retained variance increases with the increasing of rank r, this is an intuitive relationship between the amount of information and the number of terms kept.



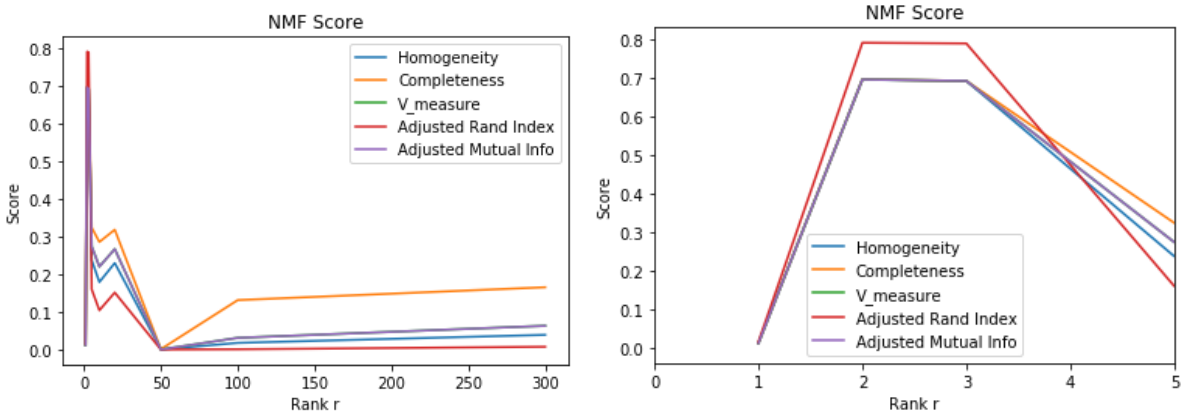
(b) we used the two methods: truncated SVD / PCA and NMF to reduce the dimension of the data. Then we swept over the dimension parameters for each method, and selected the better results in terms of clustering purity metrics. We choose r from 1, 2, 3, 5, 10, 20, 50, 100, 300 and plot the 5 measure scores v.s. r for both SVD and NMF.

Question 5:

LSI method:



NMF method:



From the above graphs, it is clear that the best r-value for SVD is r=3 and NMF is r=2. Because when r is 3 for SVD and r is 2 for NMF, the matrix reaches a balance point. In the plot, all of the 5 measures are relatively the highest using our best r and the trend of all measures are approximately the same. This is because our best r arrived at a trade-off point between the sparse problem and variance problem of matrix corresponding to low rank(small r) and high rank(large r). The sparse problem will decrease the clustering performance while the variance problem will increase the incompleteness of the information. Since all measures arrive at the best values with our best r, there is no doubt for us to make this conclusion.

Question 6:

The reason for non-monotonic is because with a smaller r, less information could be retained so that the clustering result is not good. Although with r increases, more information could be retained, when r becomes large enough, the data space will become high-dimensional. In this case, the Euclidean distance we used in the K-means algorithm becomes worse and leads to lower value of the 5 measures. Therefore, in K-means algorithm based on Euclidean distance, a proper dimension is needed to get an ideal result.

4. Visualization and Transformation

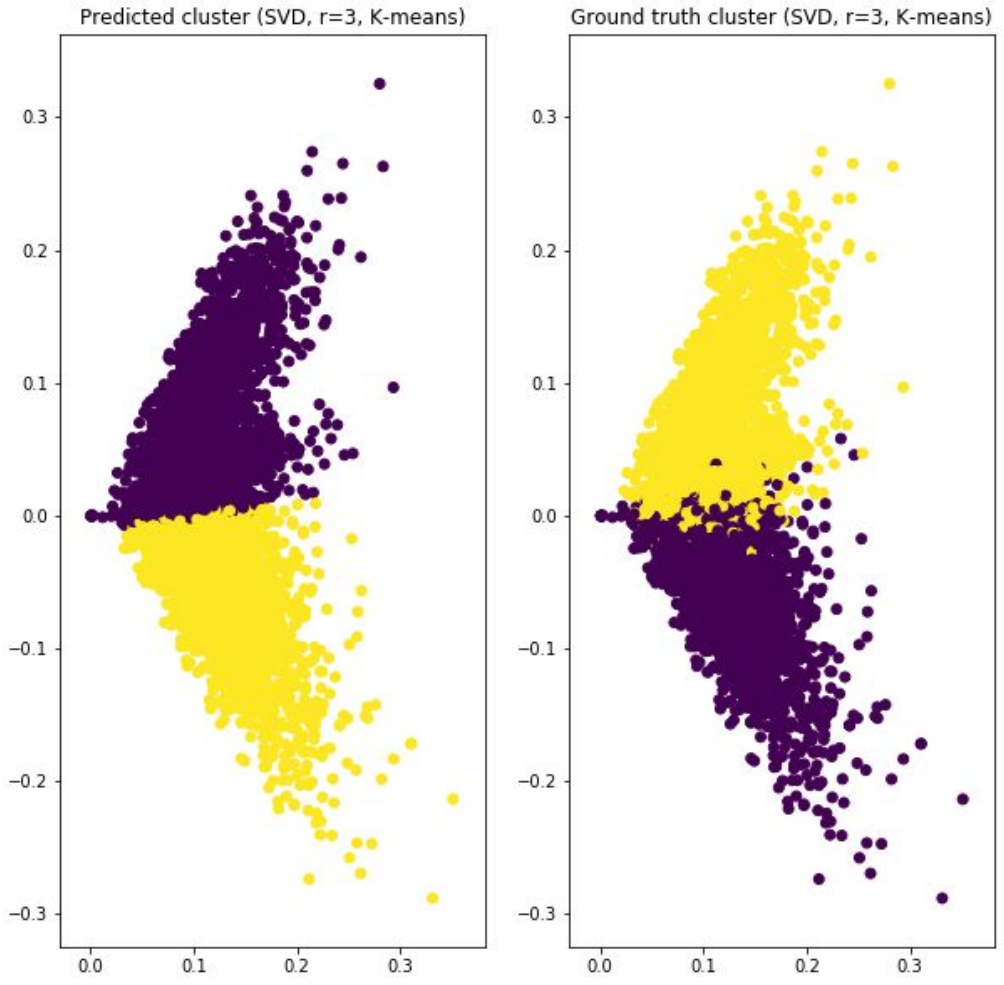
In this part, we visualize the clustering results given by the previous part by SVD and NMF method.

(a) Visualization

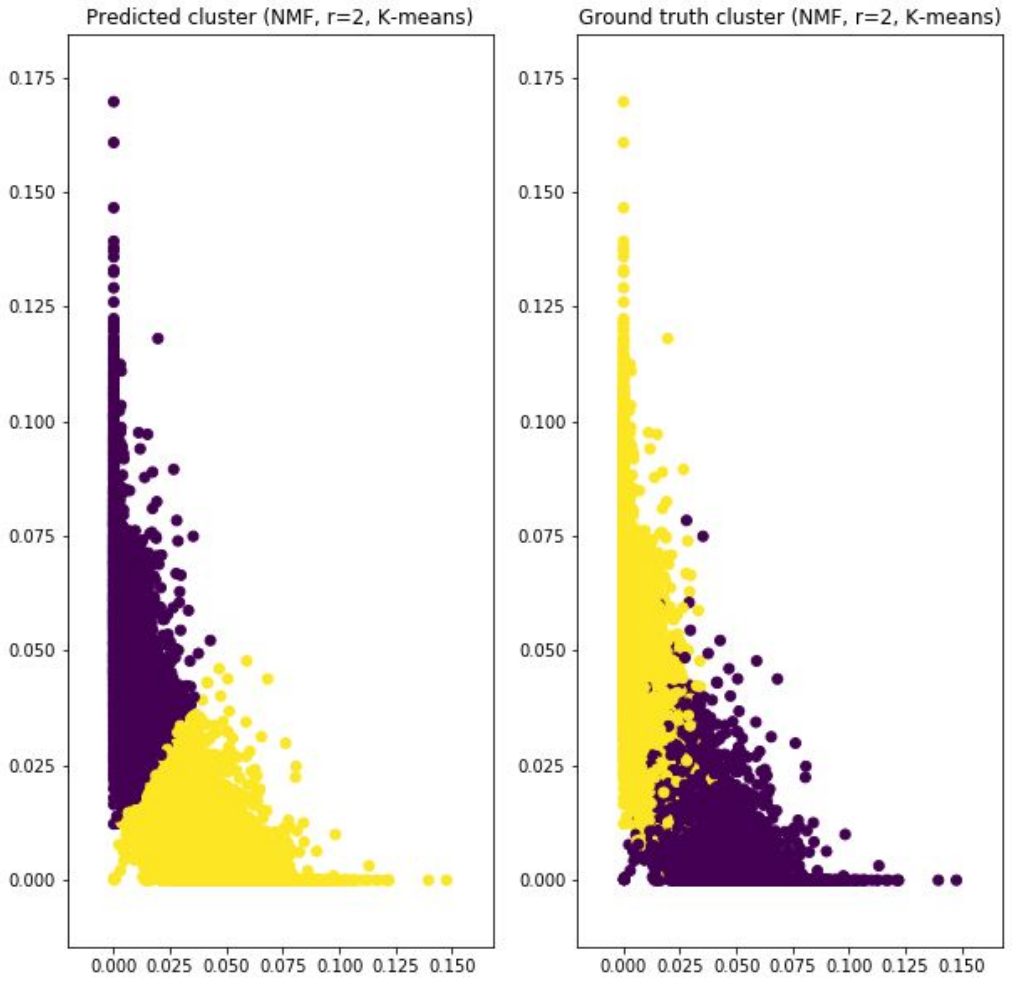
We use r=3 for SVD and r=2 for NMF to perform the dimensionality reduction and visualize the results by projecting the output data matrix onto the 2-D plane and assign colors to the data points according to the clustering and the ground truth class labels.

Question 7:

1. SVD with r=3:



2. NMF with r=2:

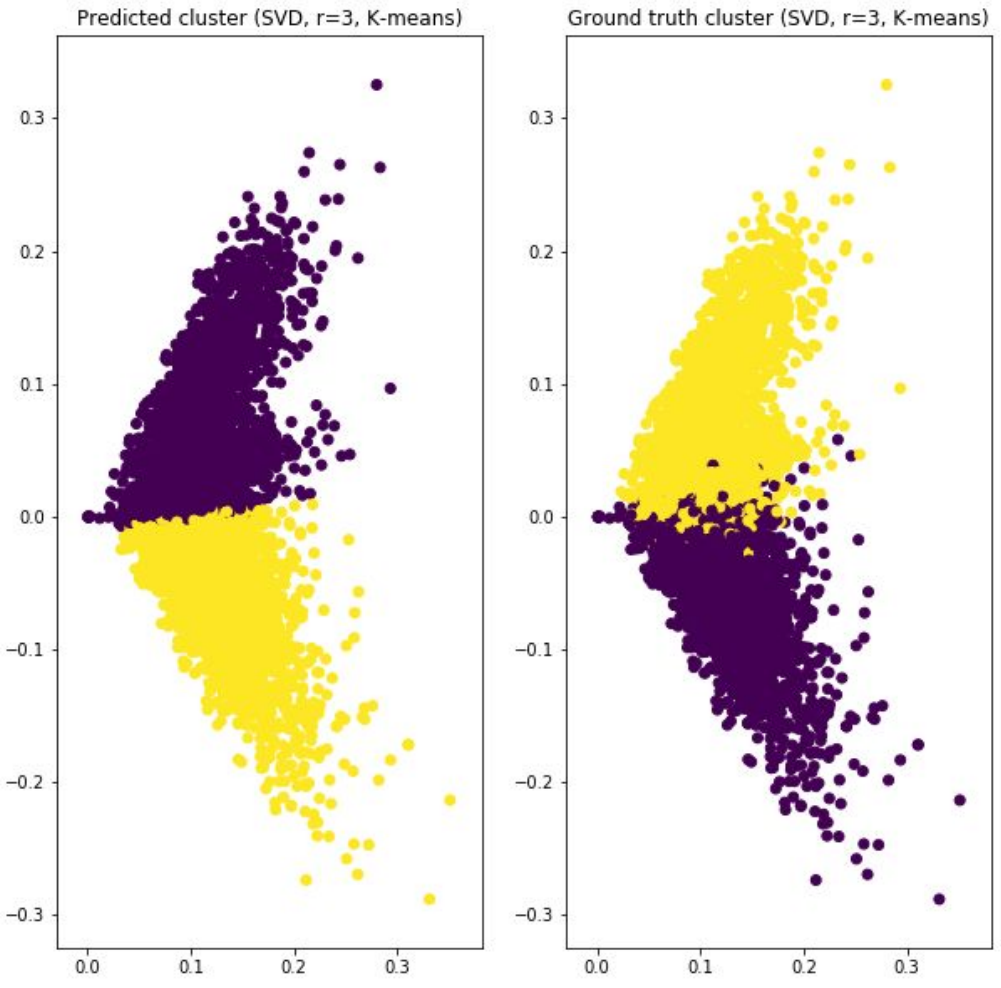


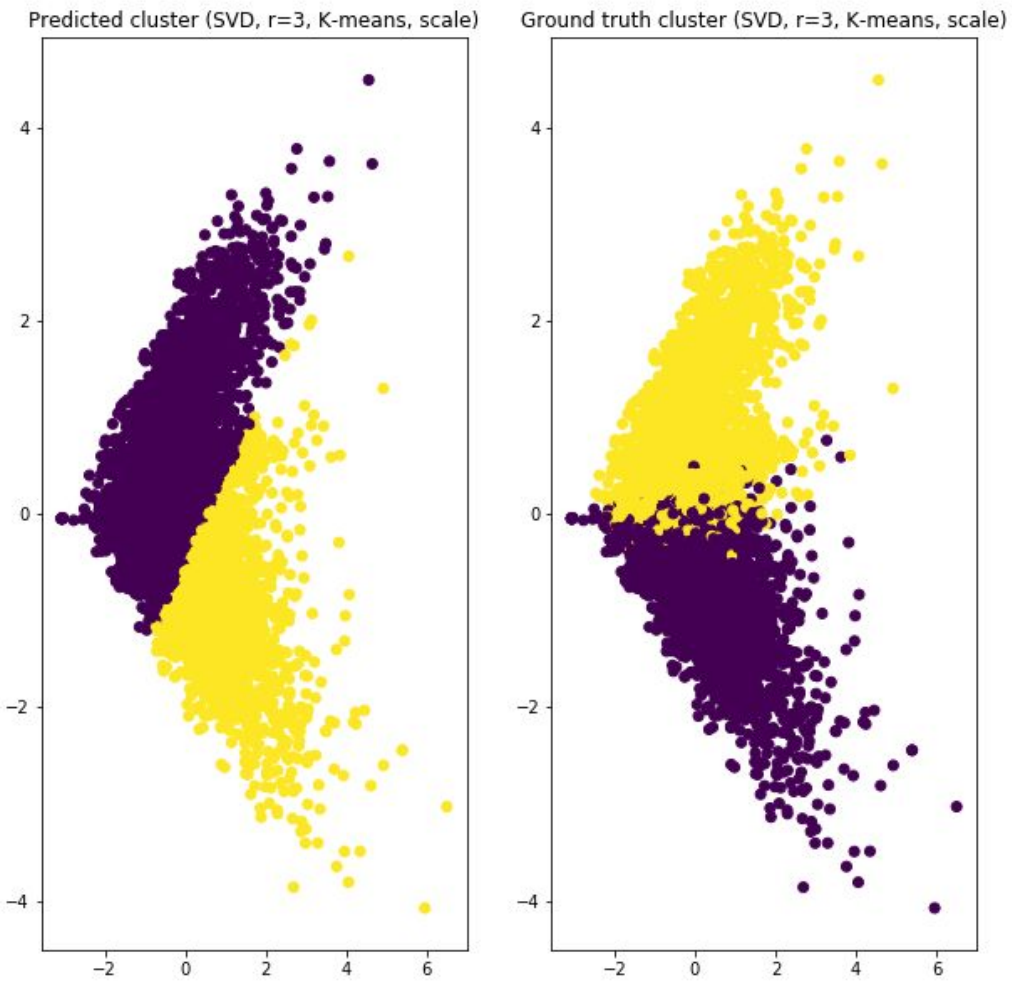
(b) Transformation

In this section, we do the transformation on the data by three different methods: scaling features, using a non-linear transformation(logarithm) and combining both of them in order to improve the K-means results.

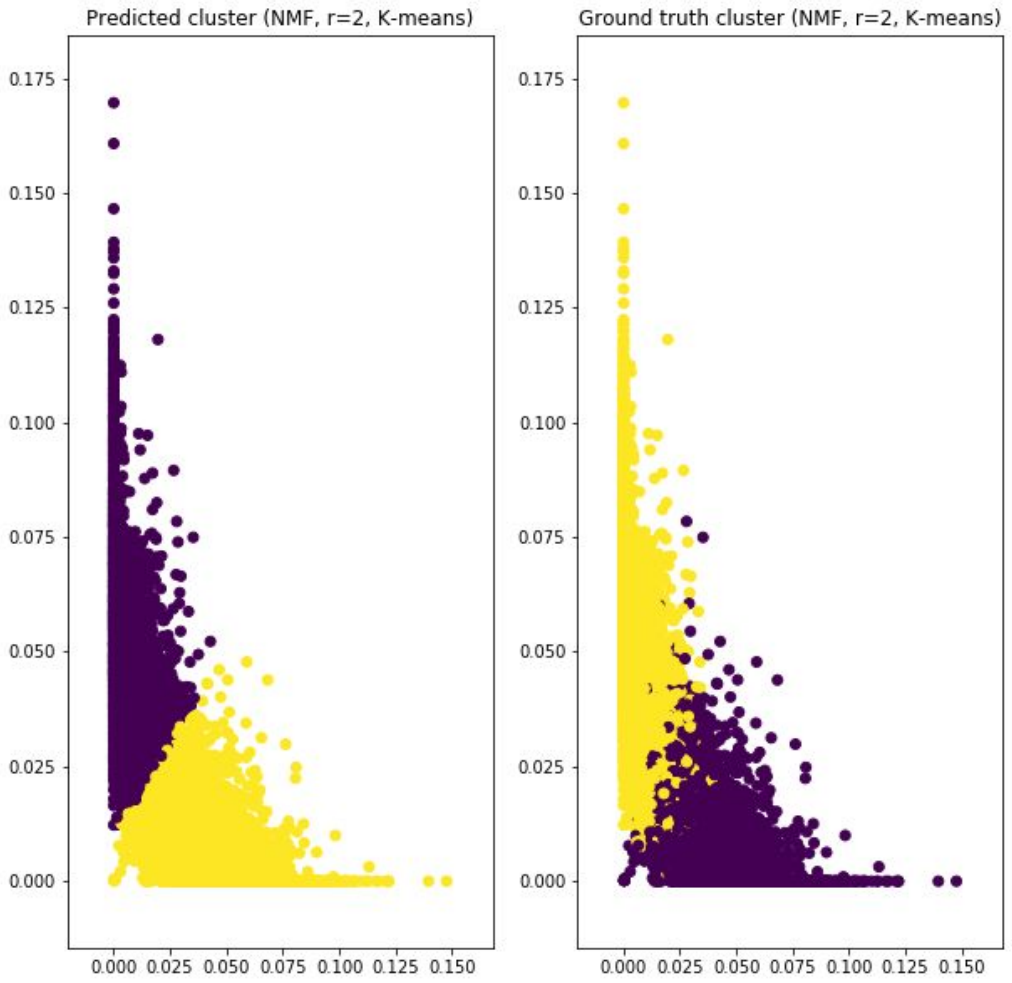
Question 8:

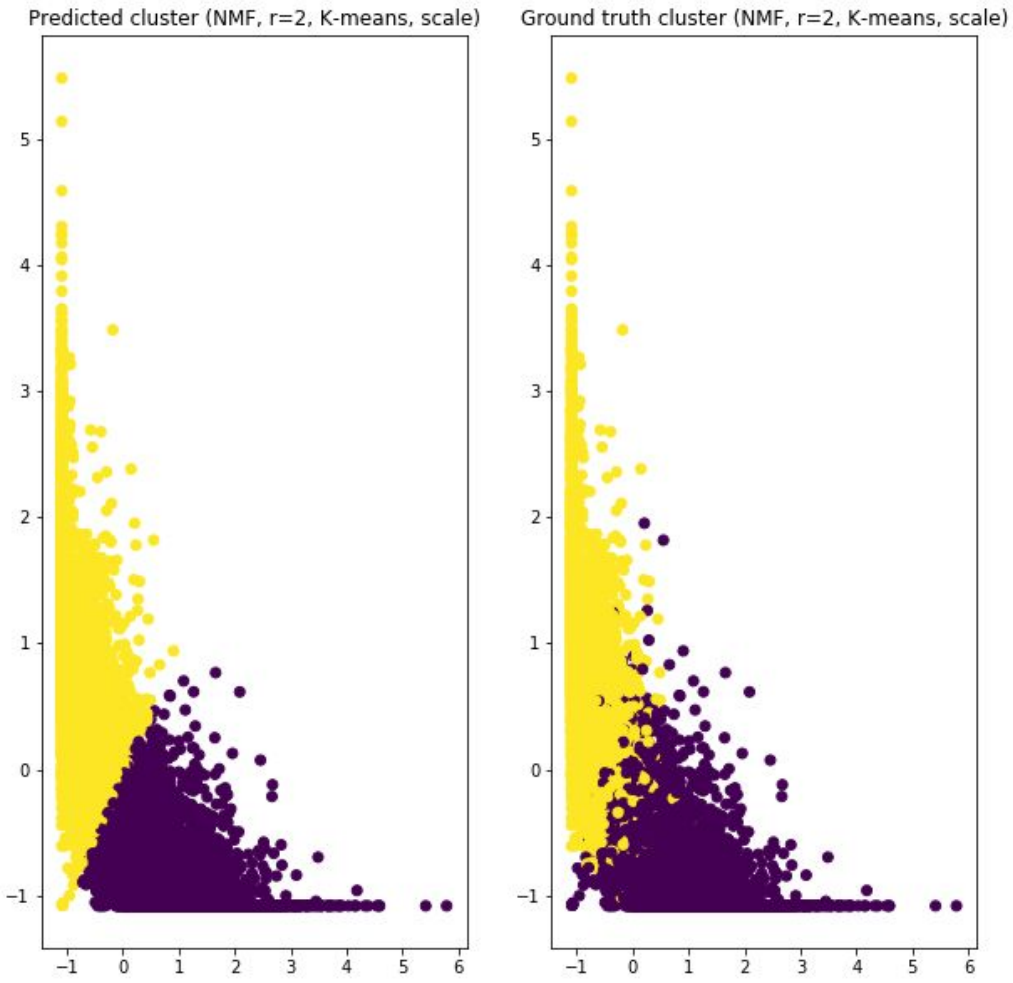
1. SVD with r=3 (2 possibilities):

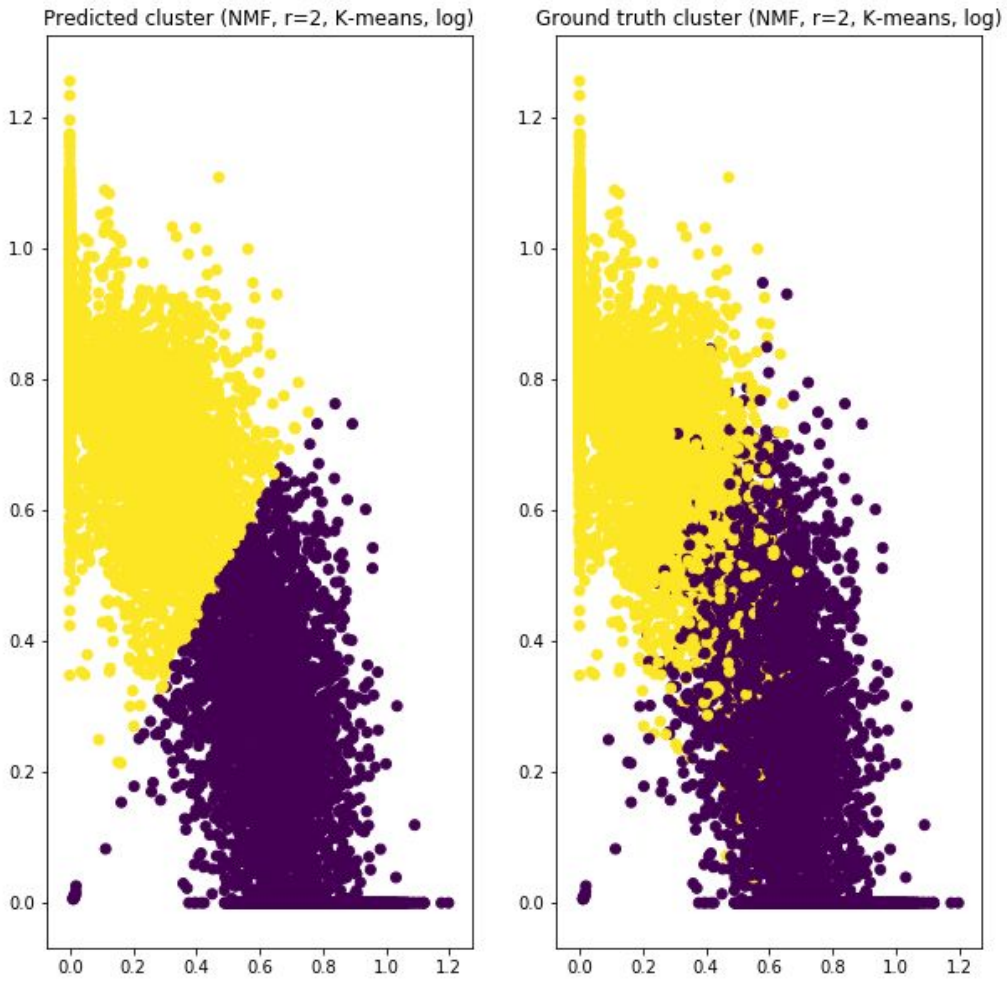

Predicted cluster (SVD, r=3, K-means)
Ground truth cluster (SVD, r=3, K-means)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.1
0.2
0.3

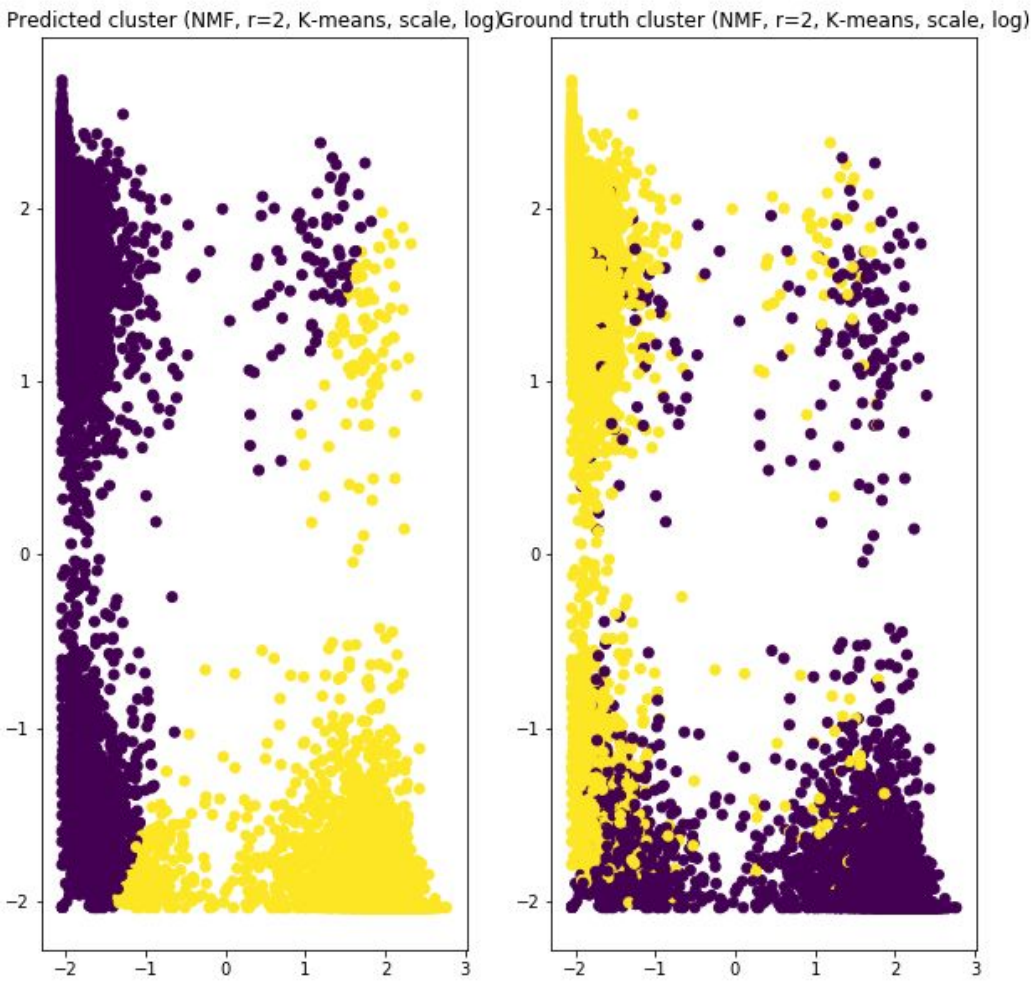


2. NMF with r=2 (4 possibilities):


Predicted cluster (NMF, r=2, K-means)
Ground truth cluster (NMF, r=2, K-means)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.000 0.025 0.050 0.075 0.100 0.125 0.150


Predicted cluster (NMF, r=2, K-means, scale)
Ground truth cluster (NMF, r=2, K-means, scale)
5
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6


Predicted cluster (NMF, r=2, K-means, log)
Ground truth cluster (NMF, r=2, K-means, log)
0.0
0.2
0.4
0.6
0.8
1.0
1.2



Question 9:

The improvement is brought by the unique property of the logarithm function. Since the values of TF-IDF vectors are ranging between 0 and 1, the logarithm function log(x) decreases rapidly as x is approaching 0. When logarithm transformation is applied to dimension reduction, centroids would spread far from each other, which increases the symmetry of the features and makes measure easier to find the distance between the data and centroid.

Question 10:

SVD with r=3	No Trans	Scaling
Homogeneity	0.6774	0.2696
Completeness	0.6782	0.3051
V-measure	0.6778	0.2863
Adjusted Rand-Index	0.7766	0.2806
Adjusted Mutual Info	0.6778	0.2862

NMF with r=2	No Trans	Scaling	Log	Scaling + Log
Homogeneity	0.6963	0.7110	0.7038	0.6468
Completeness	0.6966	0.7115	0.7044	0.6525
V-measure	0.6965	0.7113	0.7041	0.6497
Adjusted Rand-Index	0.7919	0.8064	0.8000	0.7312
Adjusted Mutual Info	0.6965	0.7112	0.7040	0.6496

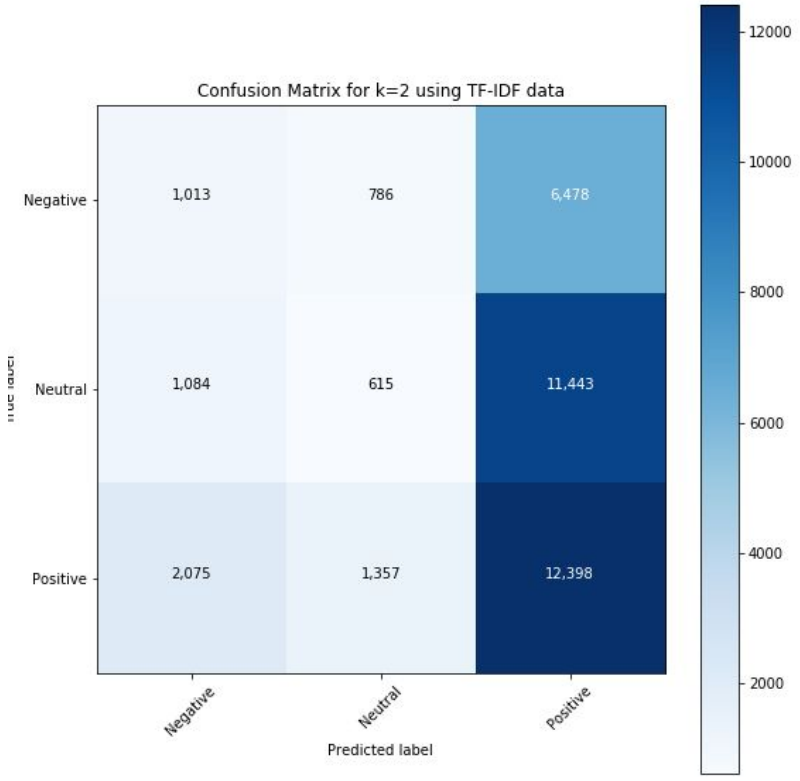
From the plot we can read that the data without scaling works best for the SVD truncated data, performing scaling operation on the data even makes the 5 measures decrease incredibly. As for NMF truncated data, the scaling transformation works best, the logarithm operation also improves the performance of the clustering using NMF truncated data. But all

of them are just slightly improved compared to the result without any transformation. The results of NMF are better than the result of SVD.

II. Your Own Dataset

In this part we downloaded a dataset for sentiment analysis from kaggle (https://www.kaggle.com/cosmos98/twitter-and-reddit-sentimental-analysis-dataset), of 3 sentiments: negative, neutral and positive. It contains comments scrapped from reddit. We extract the comments and tone from the csv files and form the training dataset where x_train denotes the comment sentence, and y_train denotes one of the 3 sentiments. The size of the dataset is 37249.

We then lemmatize the comments data and vectorize them with TF-IDF. The TF-IDF dataset is of size (37249, 18346). Then we train the dataset with different SVD and NMF k-means models to find the best r for SVD and NMF. The contingency matrix we got is as below.



The best r for SVD is 1; and the best r for NMF is also 1. Then we try transformation before K-means to get a visualization of the clustering (using r=2) and found the measurements of:

Homogeneity: 0.0498

Completeness: 0.0580

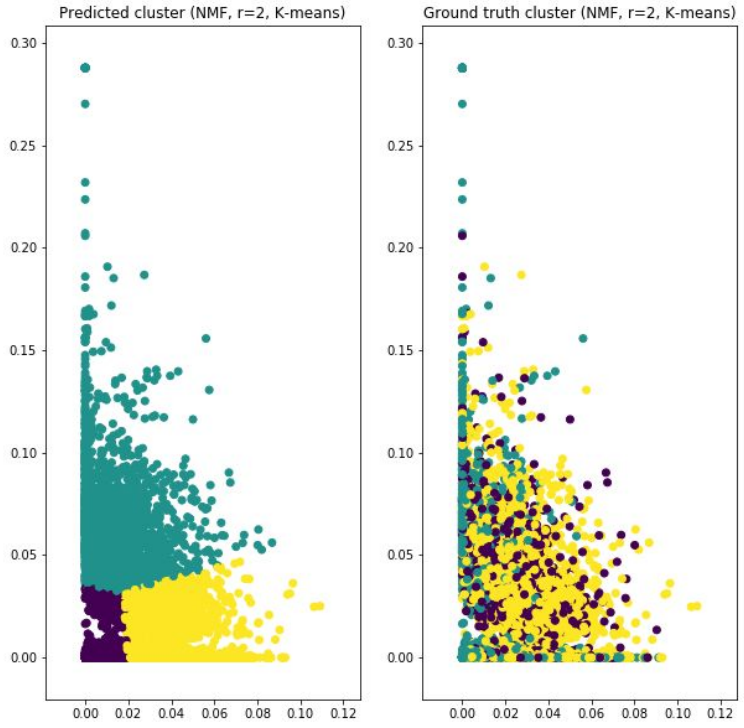
V-measure: 0.0536

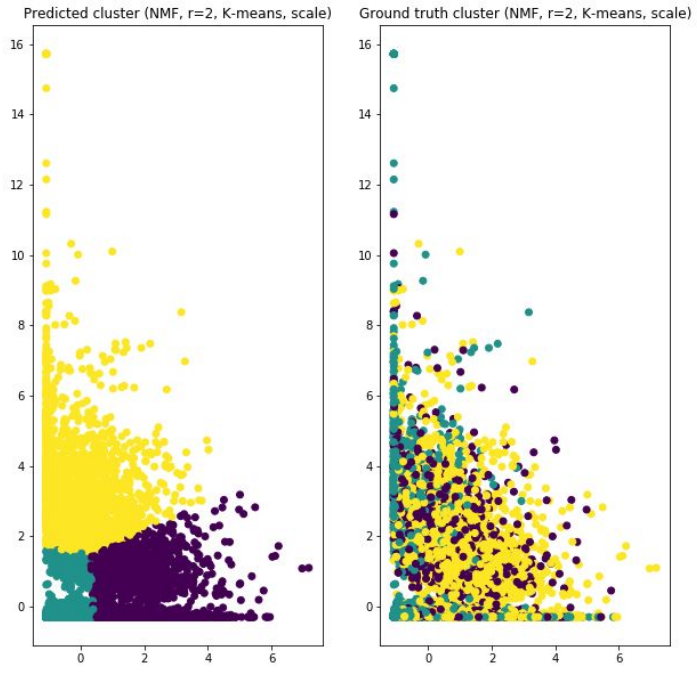
Adjusted Rand-Index: 0.0622

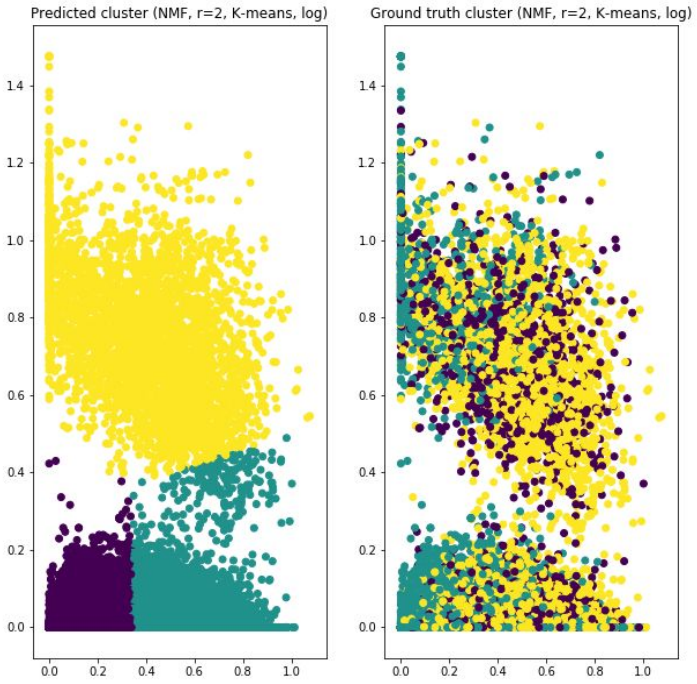
Adjusted Mutual Info: 0.0498

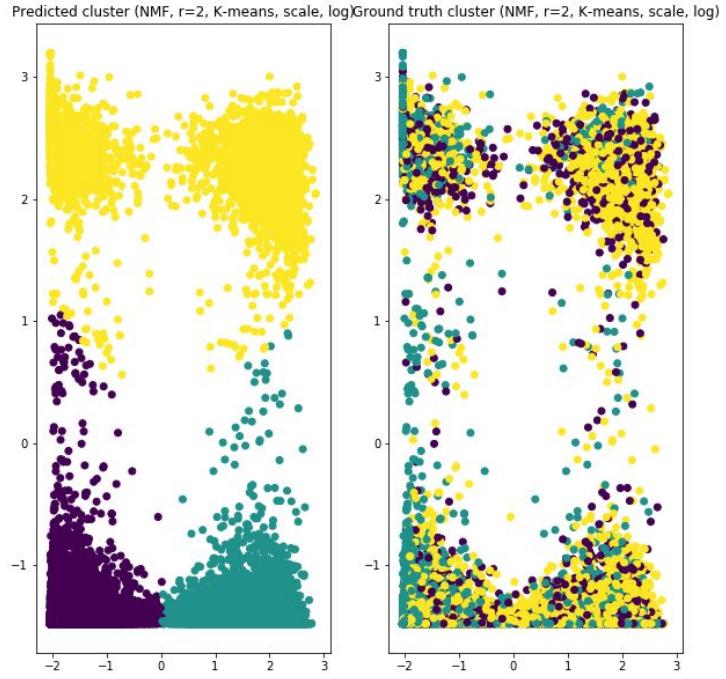
As observed from the results, the measurements are not ideal, very low actually. This is because the sentiment is a relatively subtle criteria and needs more complex language processing. And basic methods like lammetizing and vectorization cannot fully extract the feature of the sentiment tone of the text data.

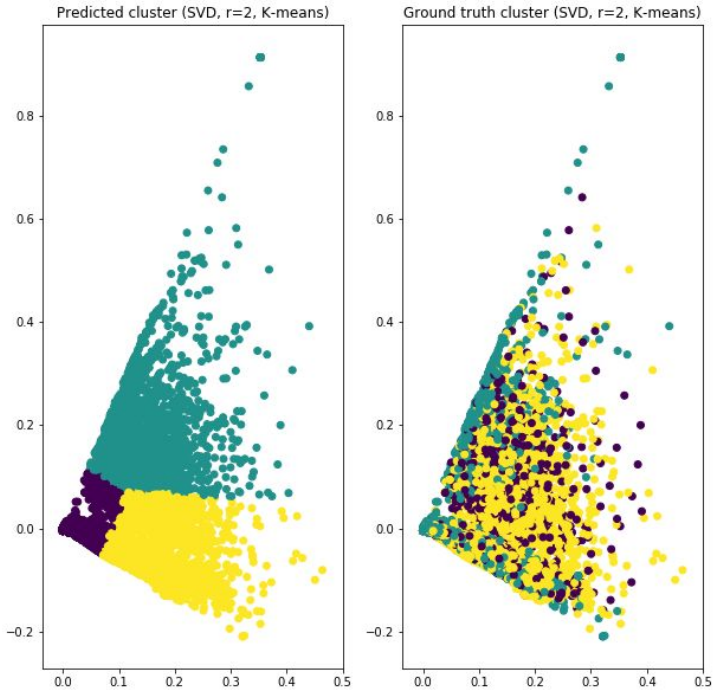
The visualized data are shown below.

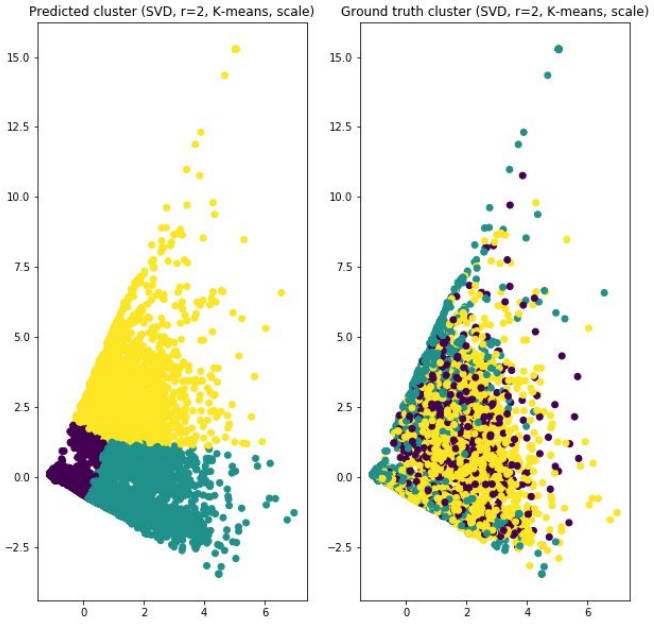



Predicted cluster (NMF, r=2, K-means, scale)
Ground truth cluster (NMF, r=2, K-means, scale)

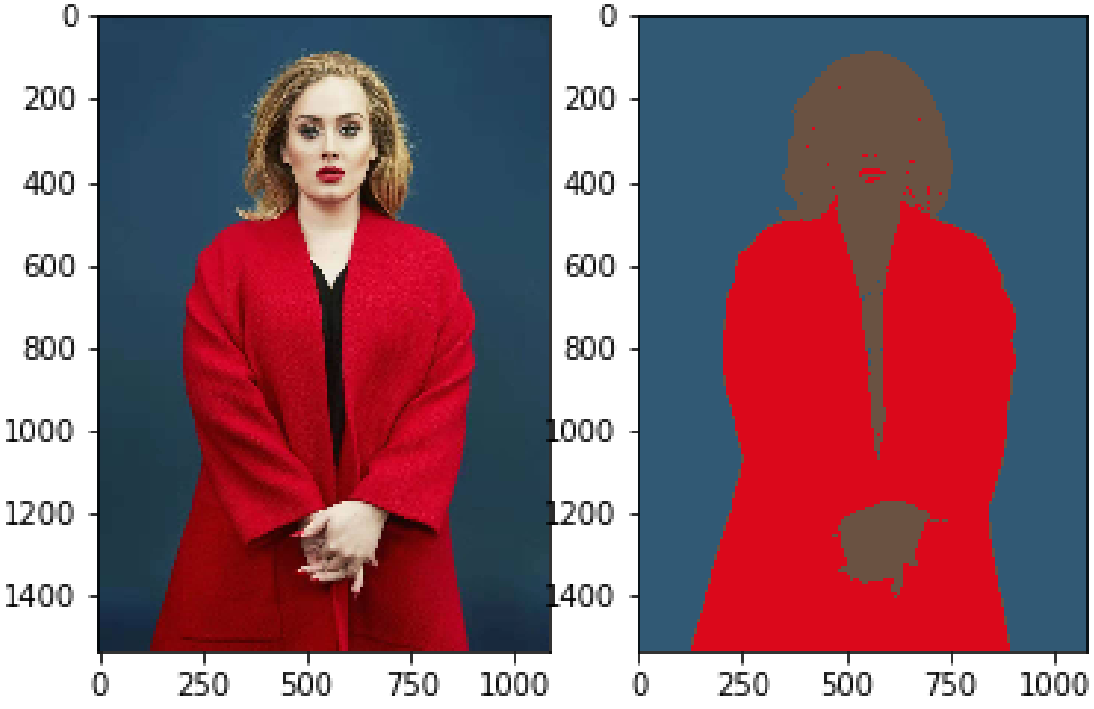

Predicted cluster (NMF, r=2, K-means, log)
Ground truth cluster (NMF, r=2, K-means, log)


Predicted cluster (NMF, r=2, K-means, scale, log)
Ground truth cluster (NMF, r=2, K-means, scale, log)


Predicted cluster (SVD, r=2, K-means)
Ground truth cluster (SVD, r=2, K-means)



III. Color Clustering



In this part, we reshaped the image into 3 vectors corresponding to 3 color channels and normalized the RGB values to perform the K-means clustering.

From the plot above, we can intuitively figure out that the features in the original image have been well segmented using K-means clustering algorithm with k=3. For example the mouth, the hand and the clothes of the celebrity has been segmented into red and brown color. The background can also be well distinguished as the color of blue.

Conclusion

In the end, we finished the project and solved the problems above. We learned and practiced different clustering measures, found best r for different dimensionality reduction, and compared two transformation methods with visualization.

Besides, we chose our own dataset of reddit comment sentiments to perform the clustering analysis, and by using K-means clustering. We also finished the color clustering with a picture of a celebrity's face.