Project 2 report draft

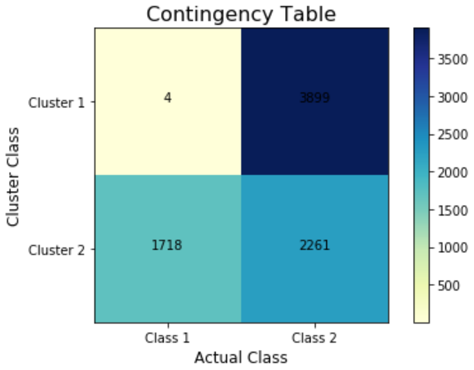
Mahmoud Essalat, Sherry He, Weitang Sun, Siqi Huang
ID: 005034839, 805040110, 904946260, 504490530

Question 1.

Before we perform the clustering, we need to transform the documents into TF-IDF vectors. We extract features by using CountVectorizer with min_df = 3 and stopwords = 'english'. The dimension of our TF-IDF matrix is (7882, 27768).

Question 2.

Then we apply K-means clustering with *k* = 2 using the TF-IDF data, with all the specs indicated by the project requirement. From the contingency table, we can see that K-means clustering performs poorly, especially cluster 2 has similar amounts of data points from class 1 and class 2. The reason is the so called curse of dimensionality. Here the feature space dimension is 27768 which is so big and all points are far from each other and have approximately the same distance which is against the K-means clustering concept of closeness.



Question 3.

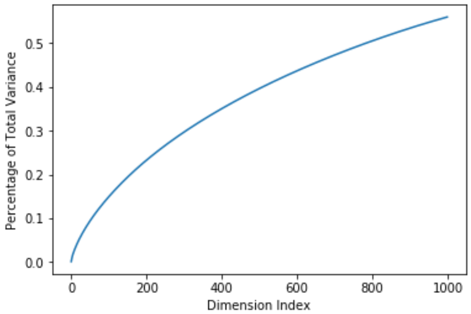
We also use measures homogeneity score, completeness score, V-measure, adjusted Rand score and adjusted mutual info score to evaluate clustering results.

Measures	Scores
Homogeneity Score	0.25
Completeness Score	0.33
V-measure	0.29
Adjusted Rand Score	0.18
Adjusted Mutual Info Score	0.25

The homogeneity score is 0.25. It means that our clusters are not homogenous and contain samples of the other class as well. The completeness score is 0.33, means that the clusters perform poorly on covering up whole data samples of one class. As homogeneity score and completeness score are less ideal, V-measure score is pretty bad as well. Adjusted rand index is similar to accuracy measure and adjusted mutual information score measures the mutual information between the cluster label distribution and the ground truth label distribution. They are both lower than 0.5. All five measures indicate that K-means clustering doesn't perform well. One of the reasons might be that K-means clustering doesn't perform well in a high-dimensional space, as the distances between data points become approximately the same.

Question 4.

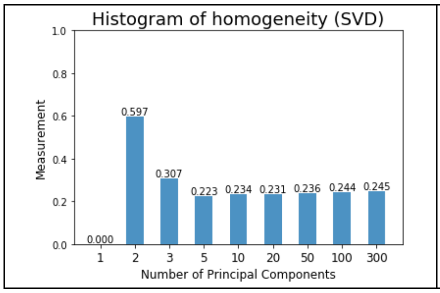
We report the plot of the percent of variance the top *r* principle components can retain from *r* = 1 to 1000. It is easy to see that, the more number of vectors we included, the more information we use and the more variances can be explained by the truncated SVD representation. When we use 1000 vectors (3.6% of the original vectors), they can explain more than 50% of the variance.

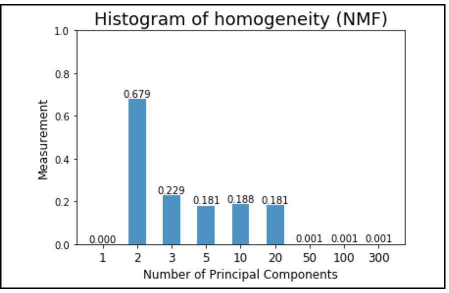


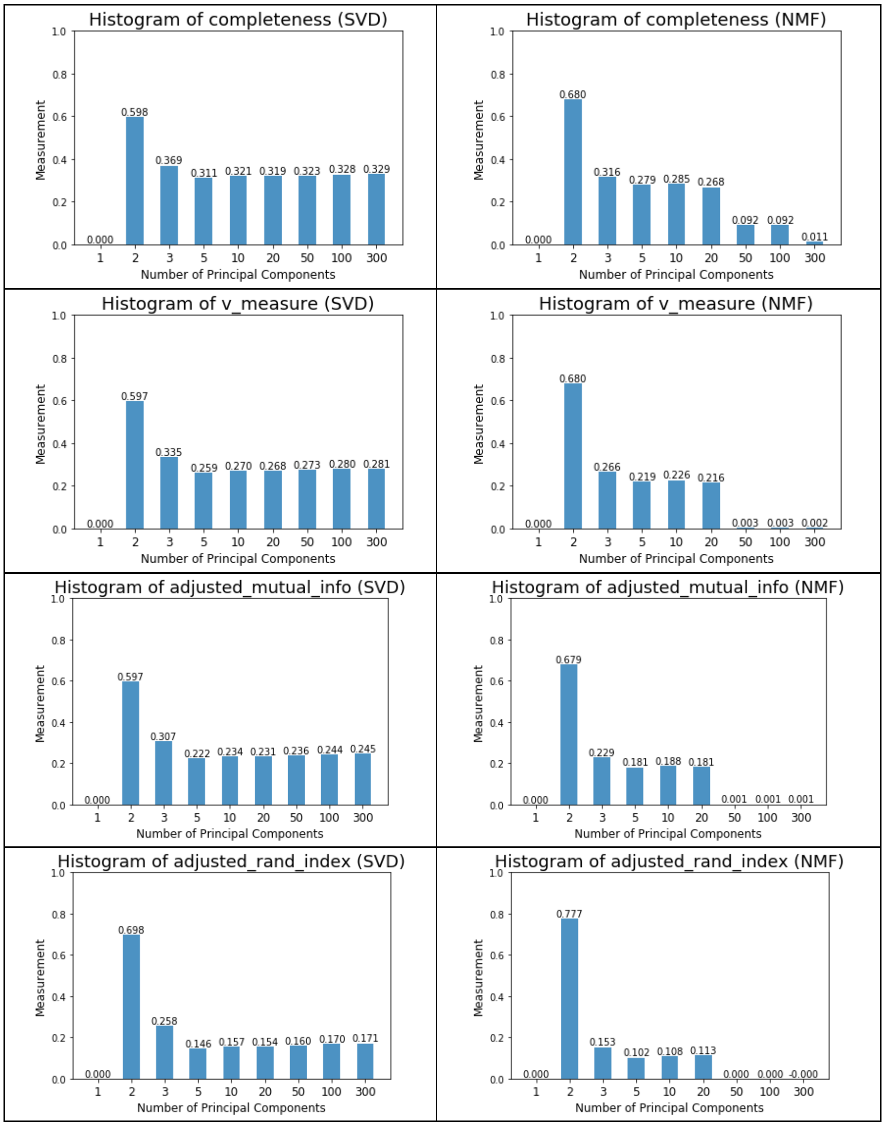
Question 5.

As K-means clustering doesn't perform well in high-dimensional space, we use both SVD and NMF to reduce the data dimension and test out the K-means clustering performance. We try *r* = 1, 2, 3, 5, 10, 20, 50, 100, 300 and plot the five measures performance in the following.

The best *r* is 2, as the five measures consistently show that when *r* = 2, the score is the highest. However, if the measures return inconsistent measure, it depends on the goal of the clustering to determine the best measure. If we care about our clusters to be homogenous, which means that we want to be pretty sure that if we select any two documents from the same cluster they end up having the same topic, we will value homogeneity more, however the clusters can end up not covering all the documents that have the same topic. Whereas if we want to cover all the documents having the same topic we should value the completeness more however each cluster might end up having different classes. Hence, we can design a new metric rather than the V-measure which is basically the harmonic average, to a weighted average. The adjusted rand score which is the accuracy score isn't a very good measure because it just takes the accuracy into account and not the accuracy on each cluster, whereas it might be the case that we are interested in accuracy of one particular class more than the rest. Also, mutual information score doesn't have all the information in homogeneity and completeness.

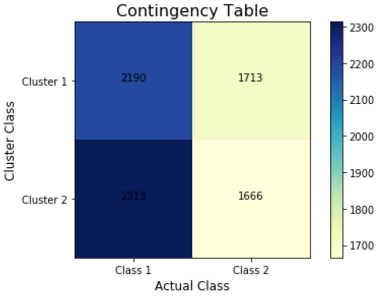




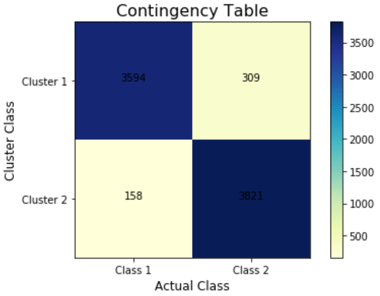


The Contingency Tables of *r* = 1, 2, 3, 5, 10, 20, 50, 100, 300 are reported in the following:
Note that in the case SVD r=2 the cluster label 1 or 2 is just a preference and one can switch that. So it shows a very good clustering result.

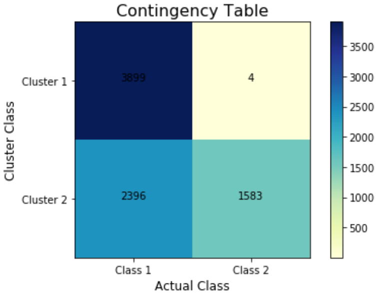
NMF, $r = 1$



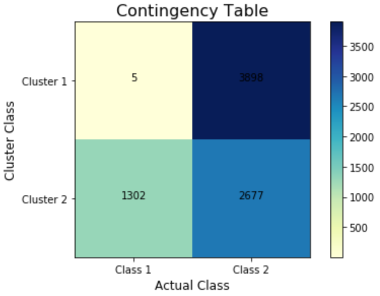
NMF, $r = 2$



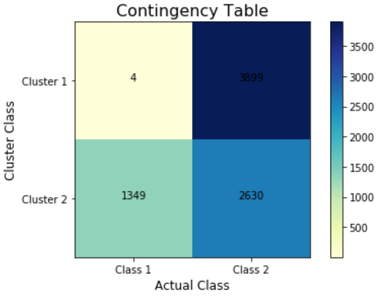
NMF, $r = 3$



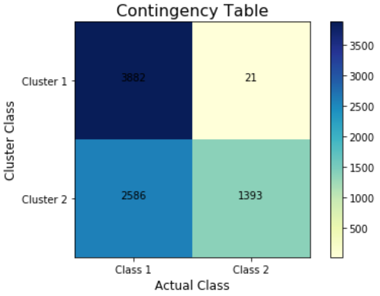
NMF, $r = 5$



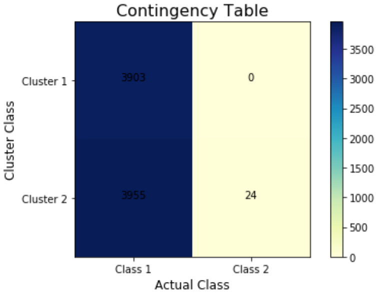
NMF, $r = 10$



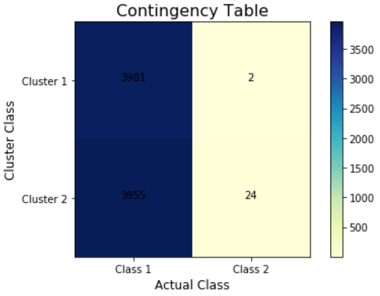
NMF, $r = 20$



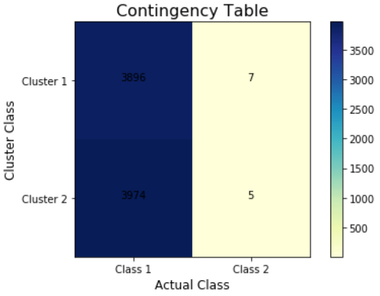
NMF, $r = 50$



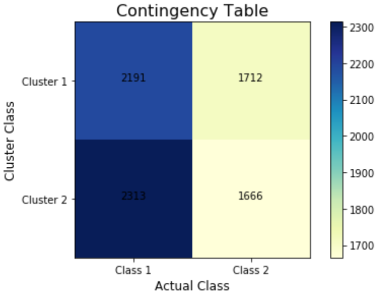
NMF, $r = 100$



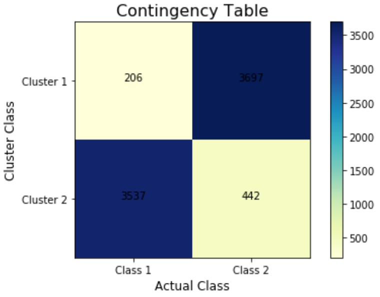
NMF, $r = 300$



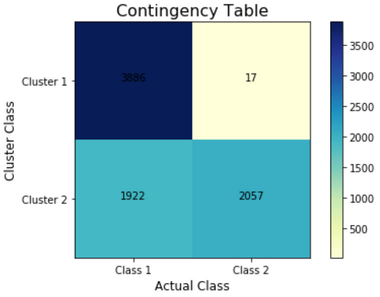
SVD, $r = 1$



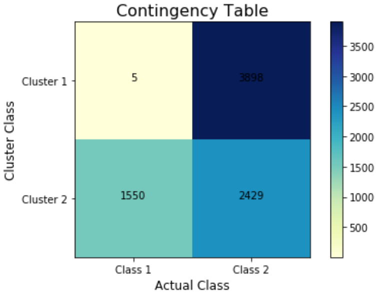
SVD, $r = 2$



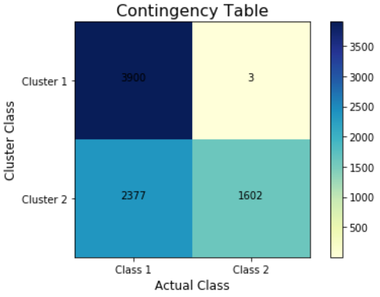
SVD, $r = 3$



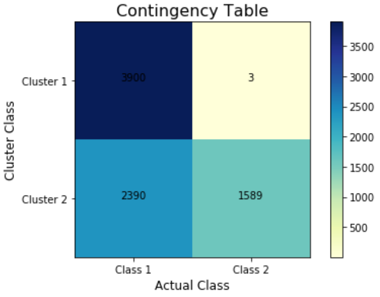
SVD, $r = 5$



SVD, $r = 10$



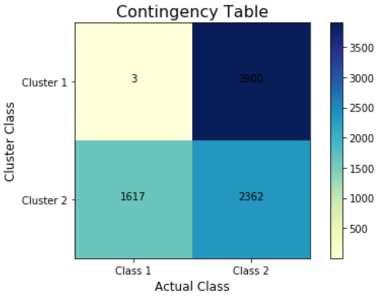
SVD, $r = 20$

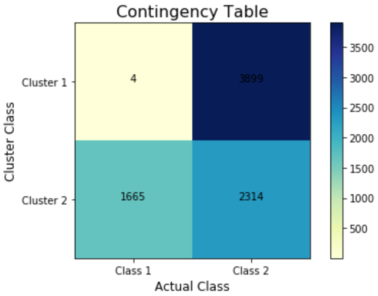


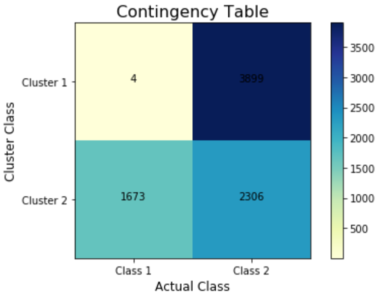
SVD, $r = 50$

SVD, $r = 100$

SVD, $r = 300$





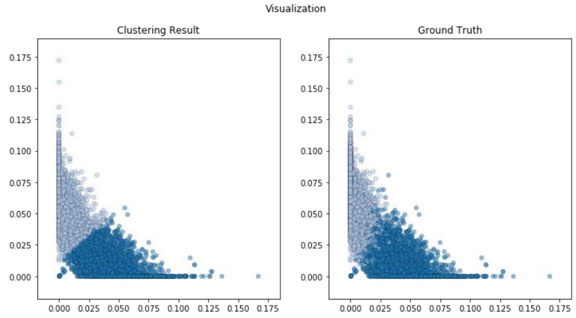
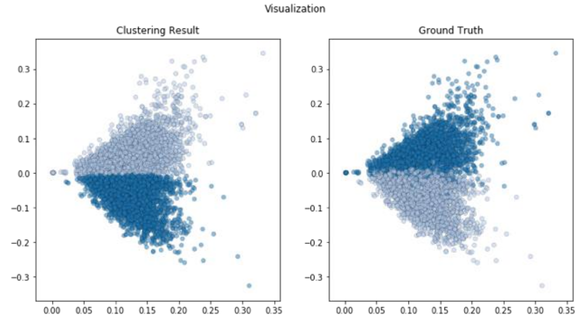


Question 6.

When r = 1, it is not surprising that the performance is unsatisfactory, as the matrix has limited information (Please note that the value is not exactly zero but it is very low). When r increases, we observe that performance enhances, as there is more information in the matrix helping with clustering. When r is too large, the performance drops as it runs into the high-dimensional problem. Euclidean distance is not a good metric for high dimensional matrix because the distances between data points tend to be the same. When we introduce more dimensions or features, we might make the data nosier and thus it leads to inaccuracy. However, the more features there are, the more information the matrix contains helping with the clustering. Combining these forces, we find out that the best *r* is not the smallest number and is not the largest number. The measures display a non-monotonic behavior. Compared with SVD, the results of NMF seems to be more volatile. The reason might be that NMF returns local optima. So that after a certain dimension it cannot find a good reduction of the data and the curse of dimensionality drops the performance drastically. Whereas, in SVD the performance remains somehow the same as long as the dimension is not so high (which we tested till 300), because it finds the best k-rank approximation and for higher dimensions there is more information which leads to slightly better performance (somehow saturated).

Question 7.

As Question 5 indicates, the best *r* is 2. We visualize the clustering results for SVD and NMF with *r* = 2. From visualization of ground truth plots, we can see that data points from each class are not well separated and not in round shape. In NMF it is dense around axes. These patterns might cause difficulties for K means clustering.

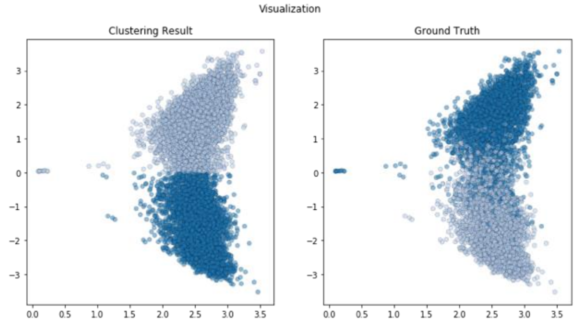
NMF	SVD
	

Question 8.

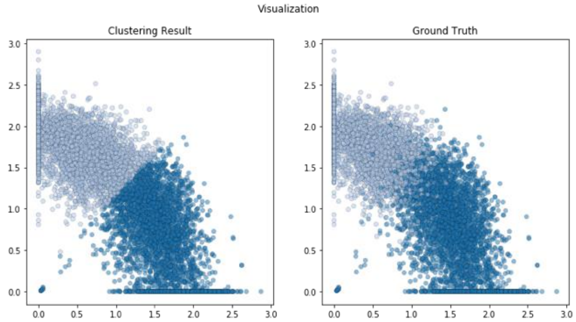
The reason for performing the scaling transformation is that we guess it will help the data to become more round shape, and as a result k-means which is based on L2 norm will perform better on round shaped data. The reason of using logarithm transform is described in Q9.

We report the plots and measures in the following. Some clustering results are visually bad, e.g. Scaled SVD and Nonlinear of scaled NMF. To see what could be the best transforming method, it is possible to plot the ground truth table and see whether data points can be easily separable by a circular curve. The reason for performing the scaling transformation is that we guess it will help the data to become more round shape, and as a result k-means which is based on L2 norm will perform better on round shaped data.

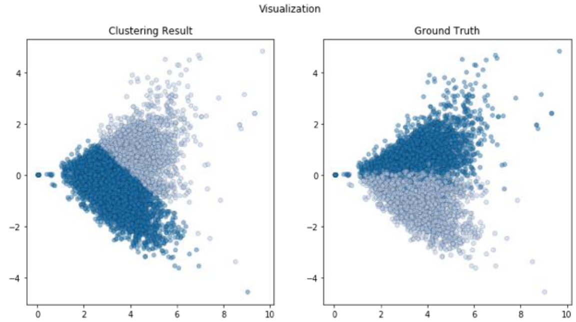
Nonlinear SVD



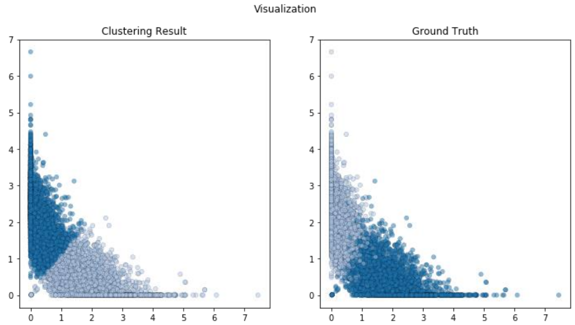
Nonlinear NMF



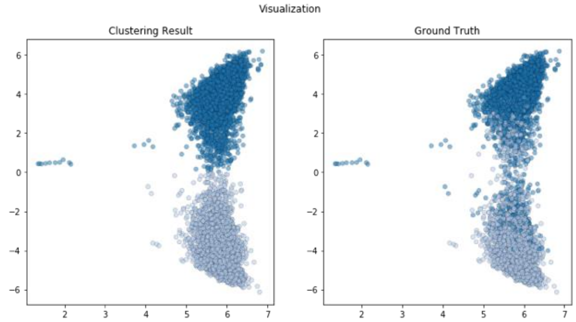
Scaled SVD



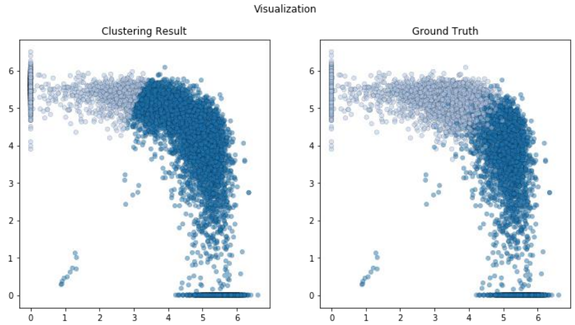
Scaled NMF



Nonlinear of scaled SVD

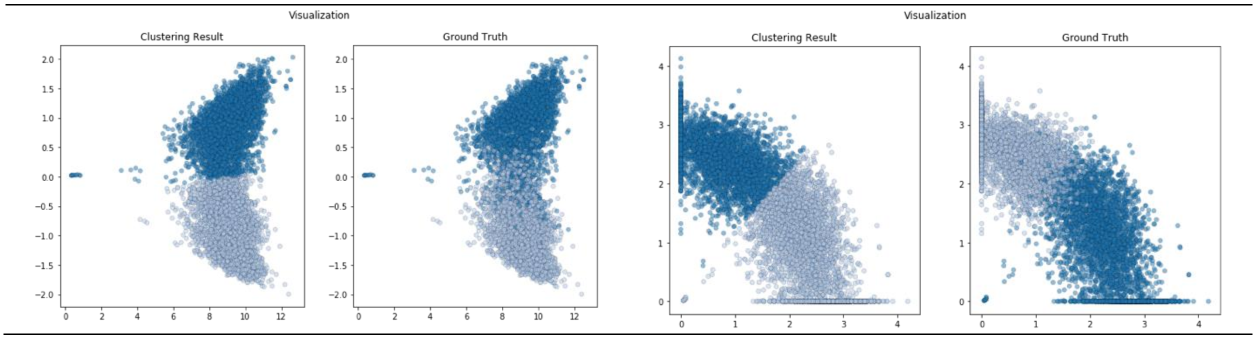


Nonlinear of scaled NMF



Scaled Nonlinear SVD

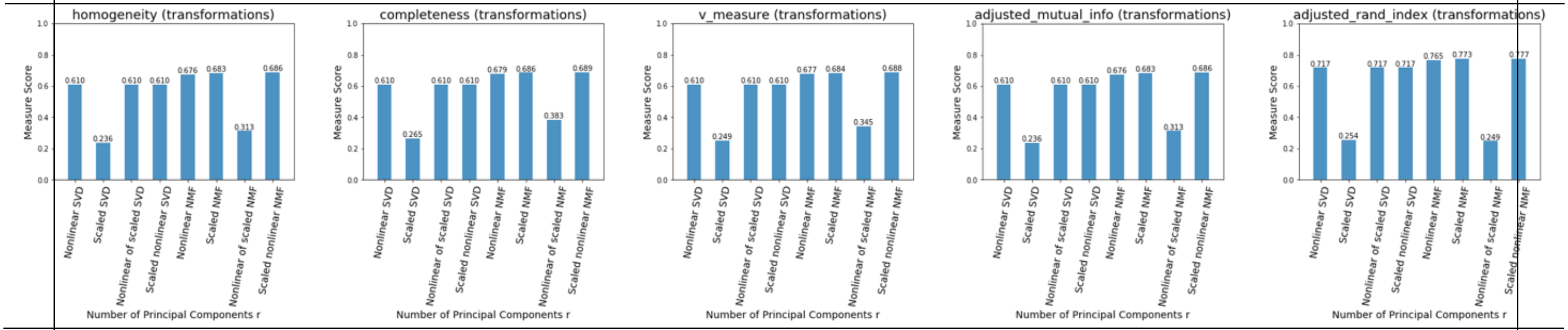
Scaled Nonlinear NMF



Question 9.
The plots show that data points can be dense near axes and the origin. Therefore, we want a method to transform the data to a more separable way for K Means method. As log function is steep near zero and monotonic, logarithm transformation can make the data points near zero separable, while still preserving the order of the data points. The logarithm domain is the positive numbers that's why there is the absolute value in the formula, the sign would be then preserved by the multiplication. Also to avoid the log of zero which is undefined the parameter C is in the formula. The reason to use scaling is described in Q8.

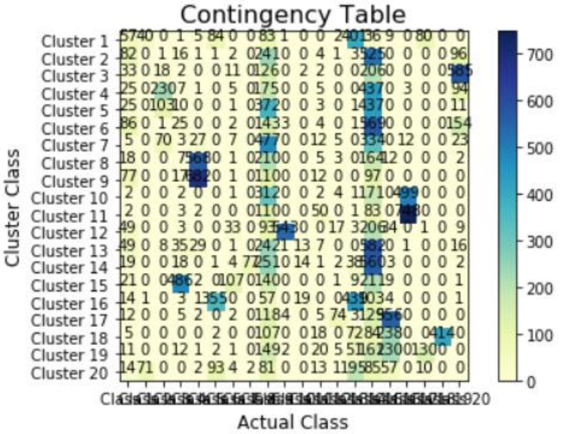
Question 10.
We can see that measures results indicate that Scaled nonlinear NMF performs the best across measures. Scaled SVD and Nonlinear of Scaled NMF perform the worst consistently.

The reason for bad performance of Nonlinear Scaled NMF is obvious from the visualization of data in Q8. The shape of the transformed data is not round at all. Hence the k-means cannot perform well on that. In the scaled SVD the shape is more round, but the reason that from the visualization in Q8 can be seen is that probably the initialization of the clustering was not good and the k-means has stuck in a local optima. But the random seed must be constant and we are not allowed to change it. Hence it emphasized the importance of initialization in k-means algorithm. Scaling implies changing the relative importance of features. As SVD picks up the most important features by variance, scaling SVD matrix will significantly decrease the clustering performance. However, if we apply the log transformation and then scaling method, we separate the data points and utilize the features in an efficient way.



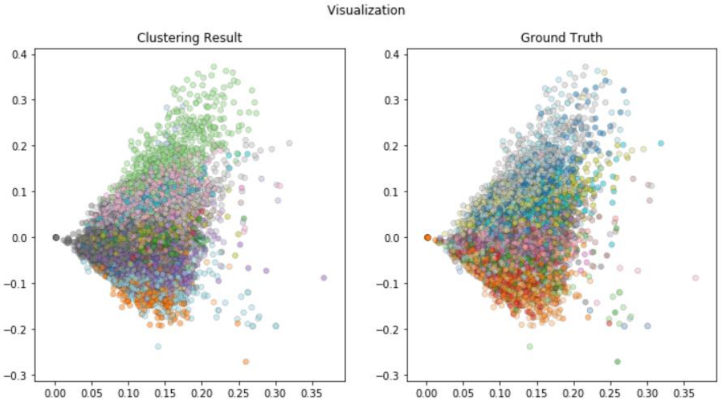
Question 11.

We extract features by using CountVectorizer with min_df = 3 and stopwords = ‘english’. We import all 20 categories. <u>The dimension of our TF-IDF matrix is (18846, 52295)</u>. This shows that the clustering cannot perform well because the feature space is very high dimension and the curse of dimensionality causes the distances to be approximately the same which is against the basic notion of k-means clustering which works based on proximity in feature space.



Measures	Scores
Homogeneity Score	0.35
Completeness Score	0.40
V-measure	0.37
Adjusted Rand Score	0.12
Adjusted Mutual Info Score	0.35

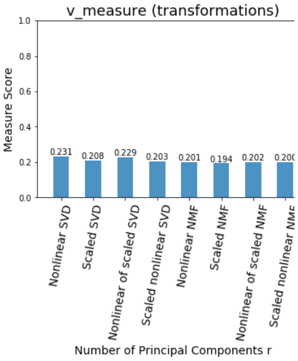
Below is the result of k-means clustering which is shown on the first two principle components of the data. As you can see the data is not well separated however same topics can be close in feature space.



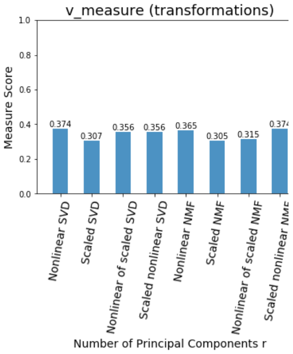
Question 12.

We could plot 8 transformed methods * 8 *r* choices (r = 2, 10, 20, 50, 80, 200, 300, 500) with the plots of five measures corresponding to each *r* choice. However, the report would be too long if we report all the plots. After eyeballing the five measures plots, we choose to present the V measure scores for each r choices, because V measure contains the information of homogeneity and completeness. Five measures also positively correlated with each other. It is not necessary to report all of them. Then we discuss what might be the best *r* and report the eight transformed methods. Later we comment on this 20 categories clustering analysis. V measures are reported in the following:

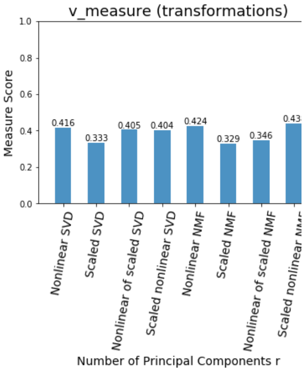
r = 2



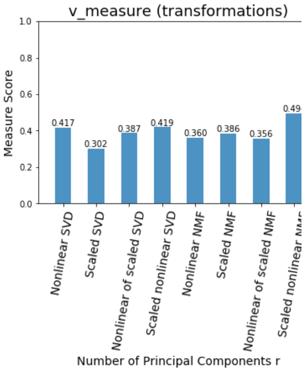
r = 10



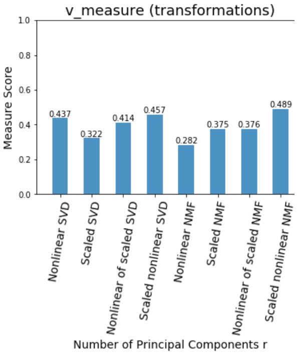
r = 20



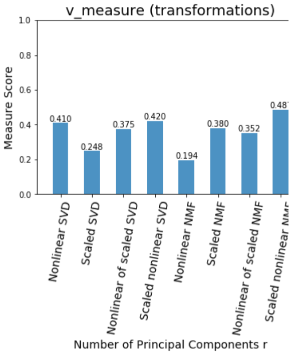
r = 50



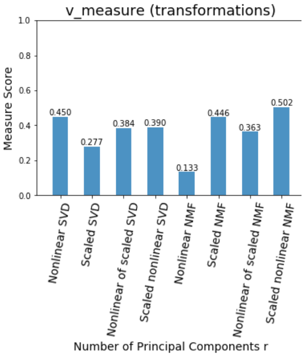
r = 80



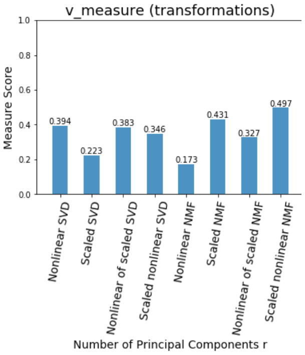
r = 200



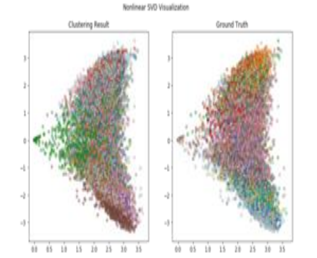
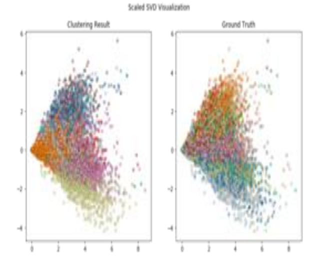
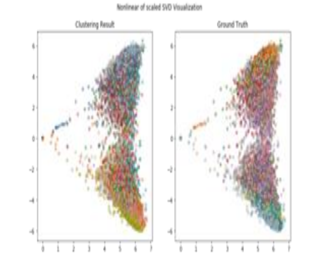
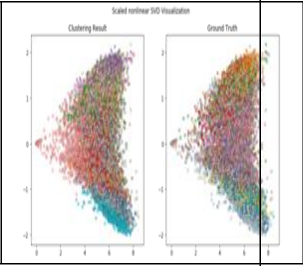
r = 300

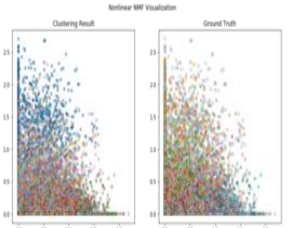
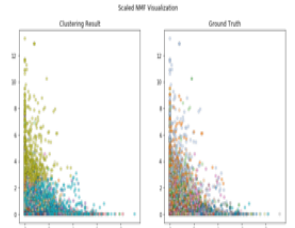
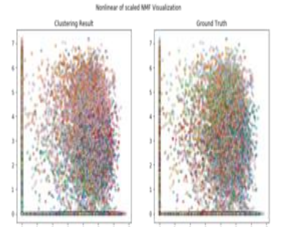
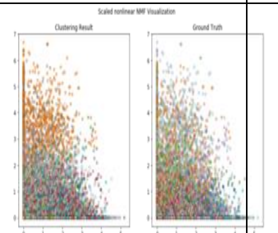
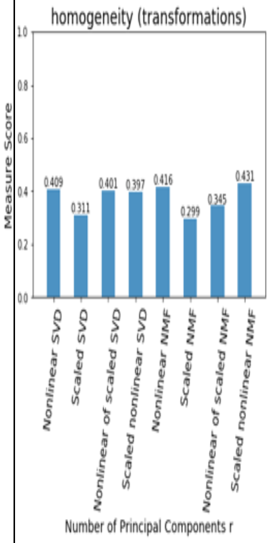
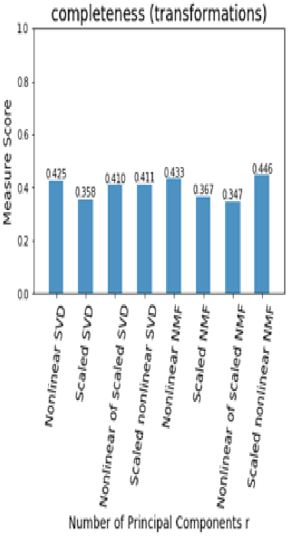
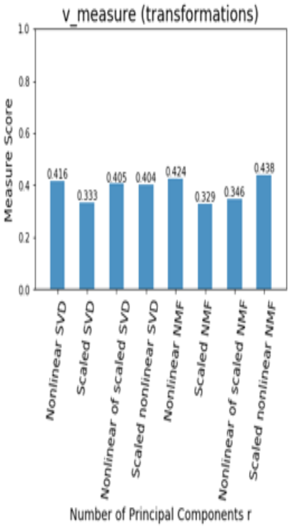
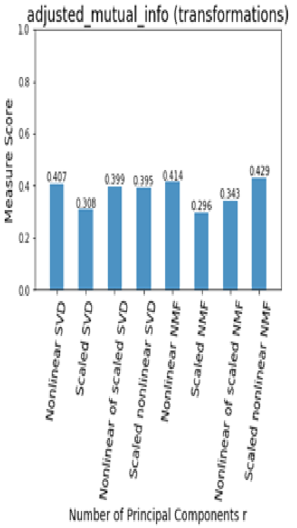
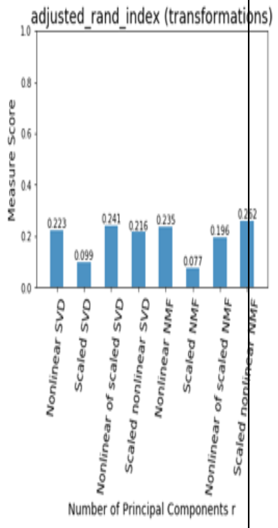


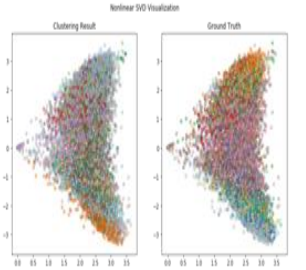
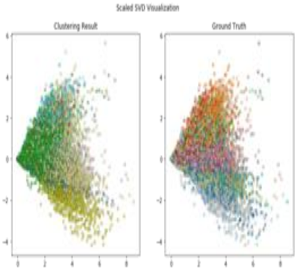
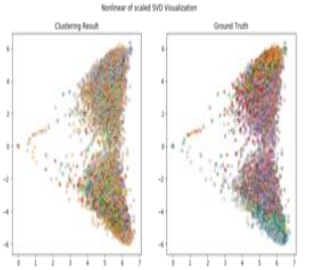
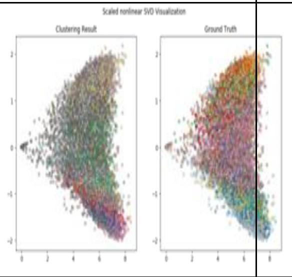
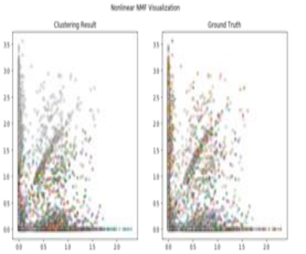
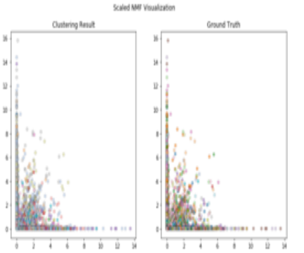
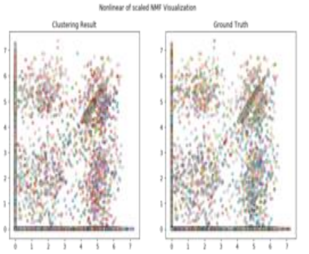
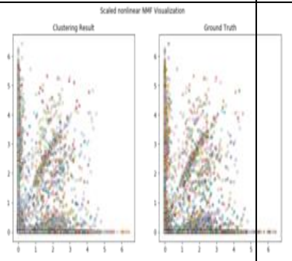
r = 500

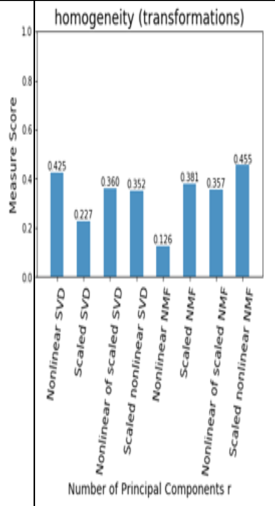


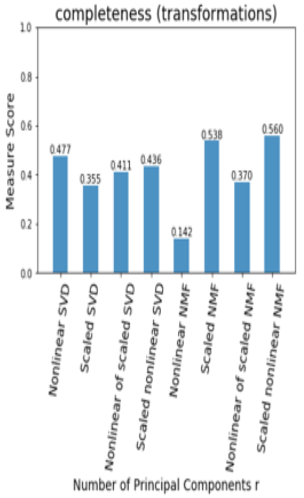
We can see that when *r* increases, the performance of clustering method increases first, and then increases for some methods and decreases for other methods. Therefore, we report *r* = 20 as an example of good and stable clustering and *r* = 300 as an example of good but volatile example.

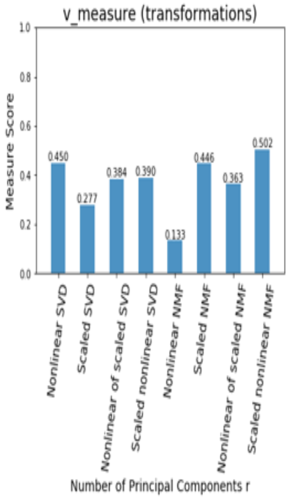
r = 2				

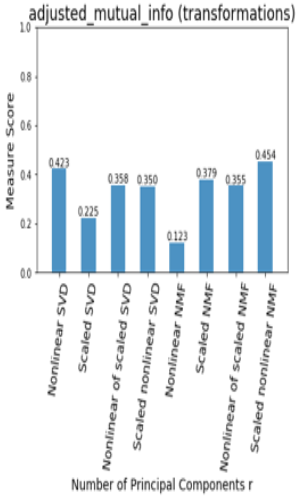
	Nonlinear NMF Visualization Clustering Result Ground Truth	Scaled NMF Visualization Clustering Result Ground Truth	Nonlinear of scaled NMF Visualization Clustering Result Ground Truth	Scaled nonlinear NMF Visualization Clustering Result Ground Truth
homogeneity (transformations) Measure Score 0.409 0.311 0.401 0.397 0.416 0.299 0.345 0.431 Nonlinear SVD Scaled SVD Nonlinear of scaled SVD Scaled nonlinear SVD Nonlinear NMF Scaled NMF Nonlinear of scaled NMF Scaled nonlinear NMF Number of Principal Components r	completeness (transformations) Measure Score 0.425 0.358 0.410 0.411 0.433 0.367 0.347 0.446 Nonlinear SVD Scaled SVD Nonlinear of scaled SVD Scaled nonlinear SVD Nonlinear NMF Scaled NMF Nonlinear of scaled NMF Scaled nonlinear NMF Number of Principal Components r	v_measure (transformations) Measure Score 0.416 0.333 0.405 0.404 0.424 0.329 0.346 0.438 Nonlinear SVD Scaled SVD Nonlinear of scaled SVD Scaled nonlinear SVD Nonlinear NMF Scaled NMF Nonlinear of scaled NMF Scaled nonlinear NMF Number of Principal Components r	adjusted_mutual_info (transformations) Measure Score 0.407 0.308 0.399 0.395 0.414 0.296 0.343 0.429 Nonlinear SVD Scaled SVD Nonlinear of scaled SVD Scaled nonlinear SVD Nonlinear NMF Scaled NMF Nonlinear of scaled NMF Scaled nonlinear NMF Number of Principal Components r	adjusted_rand_index (transformations) Measure Score 0.223 0.099 0.241 0.216 0.235 0.077 0.196 0.262 Nonlinear SVD Scaled SVD Nonlinear of scaled SVD Scaled nonlinear SVD Nonlinear NMF Scaled NMF Nonlinear of scaled NMF Scaled nonlinear NMF Number of Principal Components r

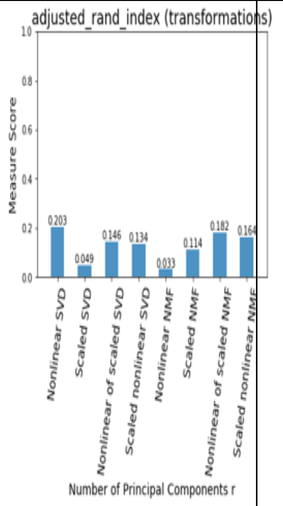
r = 300	Nonlinear SVD Visualization Clustering Result Ground Truth	Scaled SVD Visualization Clustering Result Ground Truth	Nonlinear of scaled SVD Visualization Clustering Result Ground Truth	Scaled nonlinear SVD Visualization Clustering Result Ground Truth
	Nonlinear NMF Visualization Clustering Result Ground Truth	Scaled NMF Visualization Clustering Result Ground Truth	Nonlinear of scaled NMF Visualization Clustering Result Ground Truth	Scaled nonlinear NMF Visualization Clustering Result Ground Truth











When *r* = 20, scaled nonlinear NMF works consistently best across five measures. When *r* = 300, Scaled nonlinear NMF works the best in four out of five measures. Combining all the information above, we conclude that Scaled nonlinear NMF when *r* = 300 has the best performance. The rational is stated in in more details Question 10, but to briefly mention it again: The nonlinear function tries to separate the data that are located near the axes and origin by signifying their differences by a logarithm function. Then the scaled method tries to make the data as round shape as possible so that the k-means algorithm that is based on L2 norm (which is round-shape distance measurement) would perform the best. When r is so low (e.g. less than 15) the dimension of feature space is so low that clustering cannot separate the clusters very well. By increasing the dimension, the result becomes better, but we know that in very high dimensions the curse of dimensionality can degrade the results. Therefore, r = 2 and r = ~initial size of feature space performs poorly but a choice of r = 300 or r = 20 performs the best among our clustering results.