EE 219 Project 2 Report: Clustering

Jessica Fu (805034901) & Jasmine Moreno (705035581)
Winter 2018

Introduction

In this project, we use k-means to cluster classes together. Clustering algorithms are unsupervised methods for finding groups of data points that have similar representations in a proper space. Clustering differs from classification in that no a prior labeling (grouping) of the data points is available. This means we do not use training data.

We experimented with different dimension reductions, transformation methods, linear and nonlinear transformation to determine the best clustering results. Most of this project only used two well-balanced classes to cluster. Then the last part involves all 20 sub-classes.

Part 1

In this part, we built an TF-IDF matrix using our data from the “20 Newsgroups” dataset. For this part we only worked with two classes, computer technology and recreational activity. In the table below, we can see the subclasses of the two:

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

Two well-separated classes

When transforming the data into TF-IDF vectors, we use min_df=3 and exclude the stopwords.

Results

When we transformed the dataset to TF-IDF, we got the dimensions (7882, 27768). This means we had 7882 samples and 27768 features.

Part 2

In this part, we will apply K-means clustering with k=2 to our TF-IDF vectors. Making k equal to 2 tells the function that we want to separate the data into two clusters. For this part, we use the K-means to cluster the vectors. We can inspect the clustering results by making a contingency matrix. We will determine how pure the clustering result is with respect to the ground truth by using five different measures:

- The **homogeneity** score represents the purity of a clustering results. The range of the homogeneity measure is from 0.0 to 1.0. If all the clusters only contains the same data points of the same class, then the score will be 1. Therefore, we aim to get as close to one when clustering.

- The **completeness** score ranges from 0.0 to 1.0. Unlike homogeneity, a clustering result satisfy completeness if all the data points from a class are the part of the same cluster. This perfectly complete labeling would receive a score of 1.0. Again, we try to aim for one in this score.

- The **V-measure** is the harmonic mean between the homogeneity score and the completeness score. The equation is :

  $$V = 2 \times \frac{homogeneity \times completeness}{homogeneity + completeness}$$

  Just like the previous score, V-measure has the range of 0.0 to 1.0 with 1.0 being in a perfectly complete labeling.

- The **adjusted rand** score computes the similarity measure between the clustering results and the ground truth labels. It counts all the pair of points that fall in the same or different classes which means it penalizes for both false positive and false negative decisions. The score follows the scheme:

  $$ARI = \frac{RI - Expected\ RI}{max(RI) - Expected\ RI},$$

  where RI stands for Rand Index, ARI stands for adjusted rand score. The adjusted rand score has an output range of -1 to 1 with 1 meaning the clusters are a perfect match.

- The **adjusted mutual information** score measure the mutual information between the clustering result labels and the ground truth labels. This is to account for chance, which means the random variation are centered around a mean score of 0.0. This module returns a 1 when the dataset is perfectly matched.

Results

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3899	4
True Recreational Activity	2262	1717

Clustering Result Contingency Matrix

Measurements:

- Homogeneity: 0.253
- Completeness: 0.335
- V-measure: 0.288
- Adjusted Rand-Index: 0.181
- Adjusted Mutual info score: 0.253

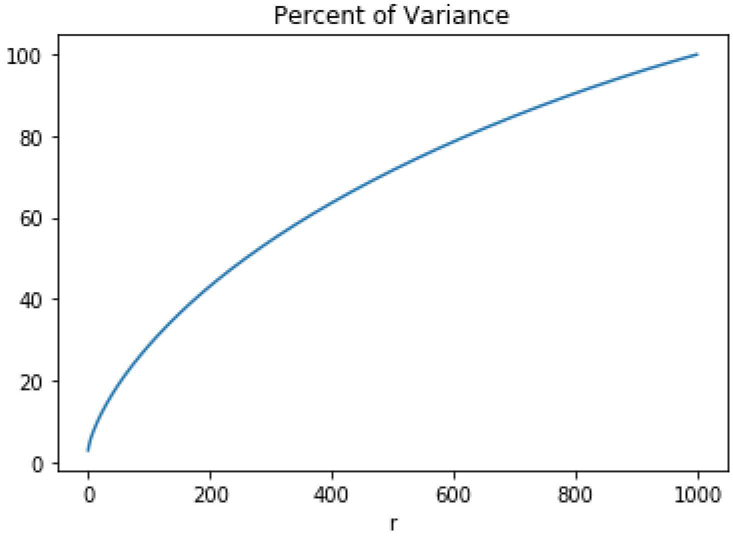
As we can see, the measurement values are really low which means our cluster are not pure. We can see from the contingency matrix that more than half of the recreational activity is not properly labeled. 2262 points are labeled as computer technology but truly should be labeled recreational activity.

Part 3

As we saw in Part 2, high dimensional sparse TF-IDF do not give a good clustering results. Since K-means goal is to minimize the sum of the distance, then we are safe to assume that clusters are isotropically shaped. If the initial cluster are not round, then K-means may also fail since the clusters have unequal variances.

In this section, we will see how the variance changes as the dimension increases when computing the k-means. We use two methods to reduce the dimensions of the data and sweep through different dimensions to find a better representation of the data set. Finding a better representation will help us improve our k-means clustering results.

Results for a - i



Percent of Variance vs r

In this data, we can see the percent of variance versus different principle component (dimension r). The dimensions we tested ranged from 1 to 1000. As we can see the data, the percent of variance increases as the dimension increases. This gives us a slight hint that for this data, it will be probably be best to reduce the data's dimension to a really low number.

Results for a-ii (SVD)

For these results, we tested reducing our dimensions to r = 1, 2, 3, 5, 10, 20, 50, 100, 300 using the SVD method. Then we calculated the contingency matrix for each r and plotted the five measurements on separate plots. Below are the results.

Contingency Matrices

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	2187	1716
True Recreational Activity	2307	1672

Clustering Result Contingency Matrix when r = 1

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3694	209
True Recreational Activity	441	3538

Clustering Result Contingency Matrix when r = 2

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	35	3868
True Recreational Activity	2562	1417

Clustering Result Contingency Matrix when r = 3

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	5	3898
True Recreational Activity	1540	2439

Clustering Result Contingency Matrix when r = 5

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3	3900
True Recreational Activity	1617	2362

Clustering Result Contingency Matrix when r = 10

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3900	3
True Recreational Activity	2369	1610

Clustering Result Contingency Matrix when r = 20

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3900	3
True Recreational Activity	2332	2324

Clustering Result Contingency Matrix when r = 50

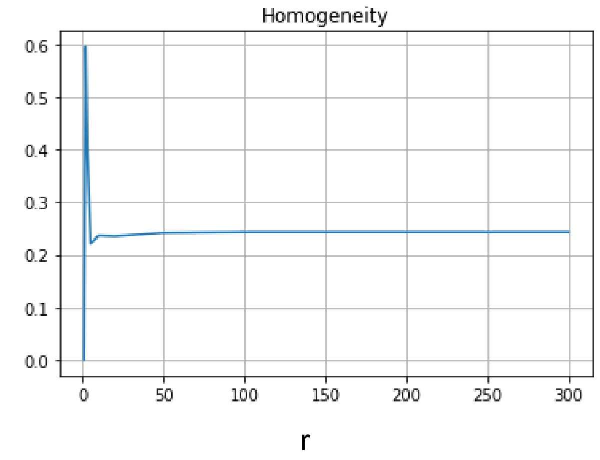
	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3	3900
True Recreational Activity	1655	2324

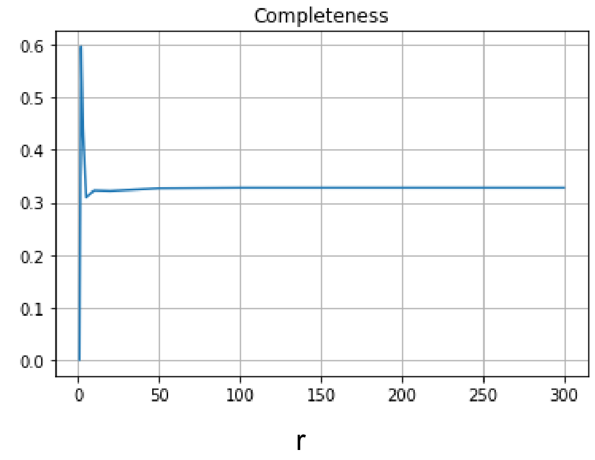
Clustering Result Contingency Matrix when r = 100

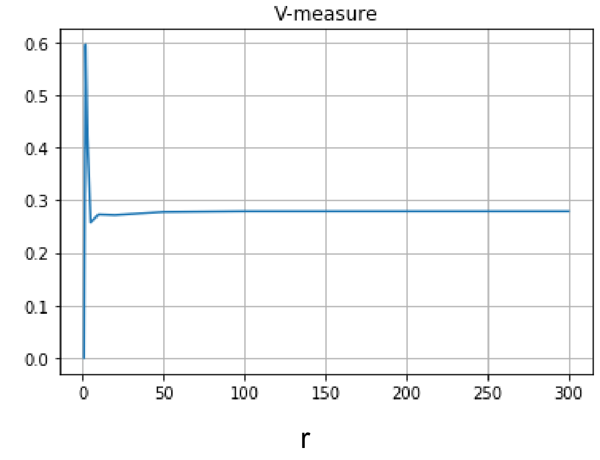
	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3	3900
True Recreational Activity	1655	2324

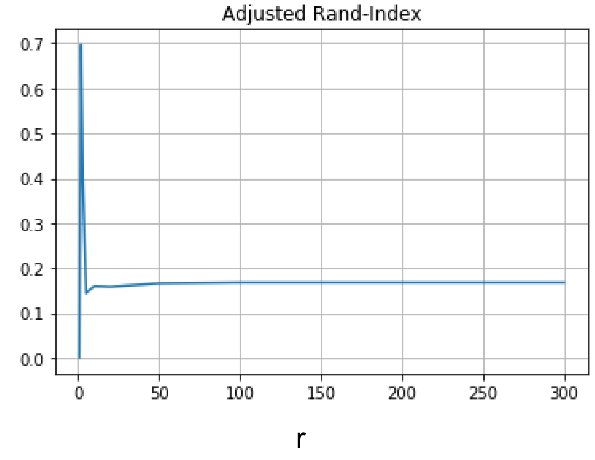
Clustering Result Contingency Matrix when r = 300

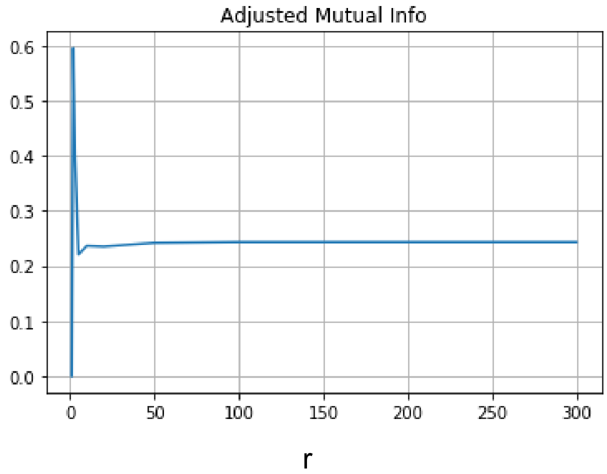
Measurement Plots











Just looking at the contingency matrix does not help us determine which dimension had the best value. Therefore, we needed to look at both contingency matrix and the measure value plots. We want to pick the r that is the closest to 1 for all the plots above. As we can see in the above plots and the contingency matrix, that the best results happens when r = 2.

Results for a-ii (NMF)

For these results, we tested reducing our dimensions to r = 1, 2, 3, 5, 10, 20, 50, 100, 300 using the NMF method. Then we calculated the contingency matrix for each r and plotted the five measurements on separate plots. Below are the results.

Contingency Matrices

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	2200	1703
True Recreational Activity	2323	1656

Clustering Result Contingency Matrix when r = 1

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3594	309
True Recreational Activity	158	3821

Clustering Result Contingency Matrix when r = 2

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	4	3899
True Recreational Activity	1583	2396

Clustering Result Contingency Matrix when r = 3

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	5	3898
True Recreational Activity	1302	2677

Clustering Result Contingency Matrix when r = 5

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	4	3899
True Recreational Activity	1361	2618

Clustering Result Contingency Matrix when r = 10

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3881	22
True Recreational Activity	2582	1397

Clustering Result Contingency Matrix when r = 20

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	10	3893
True Recreational Activity	0	3979

Clustering Result Contingency Matrix when r = 50

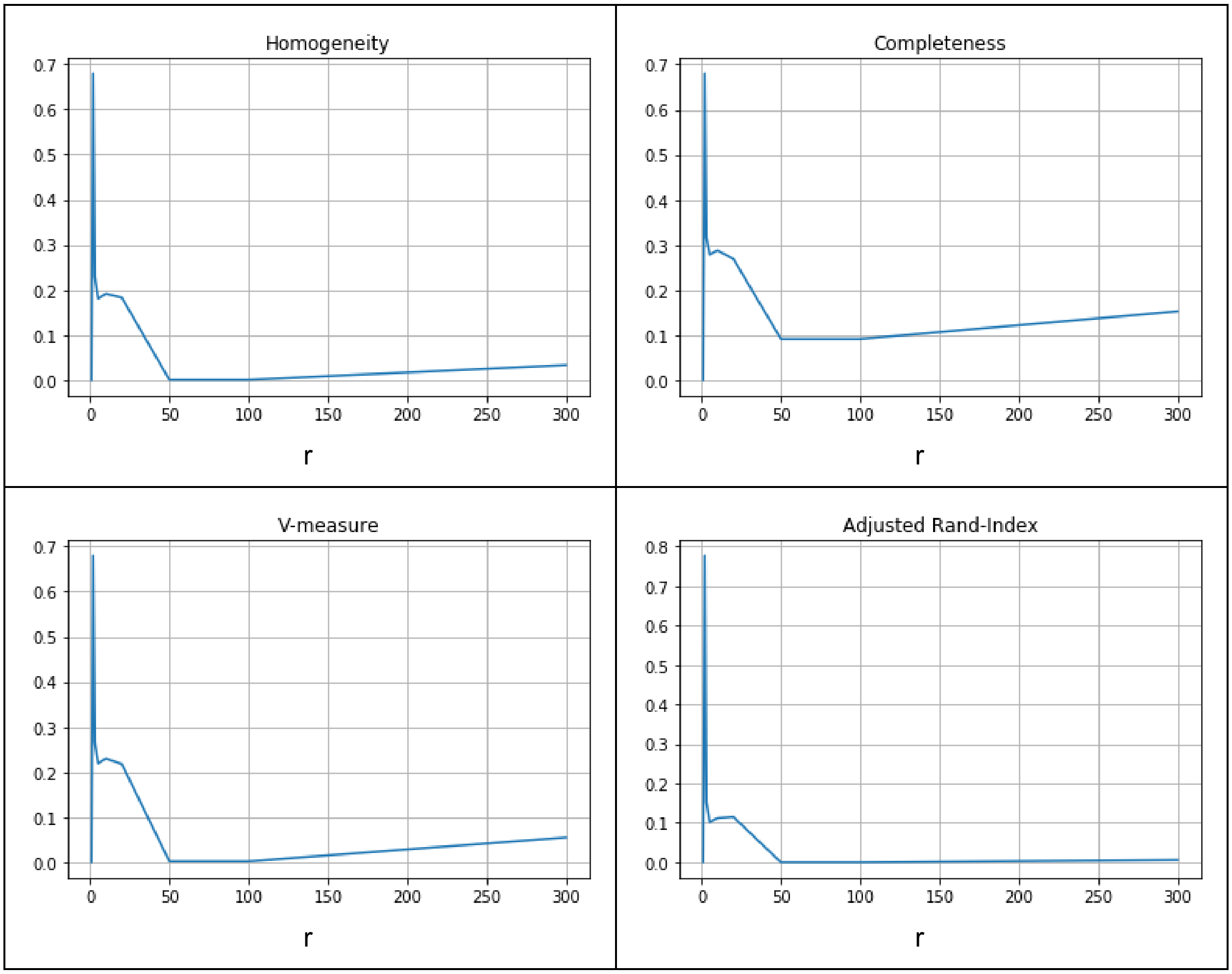
	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	10	3893
True Recreational Activity	0	3979

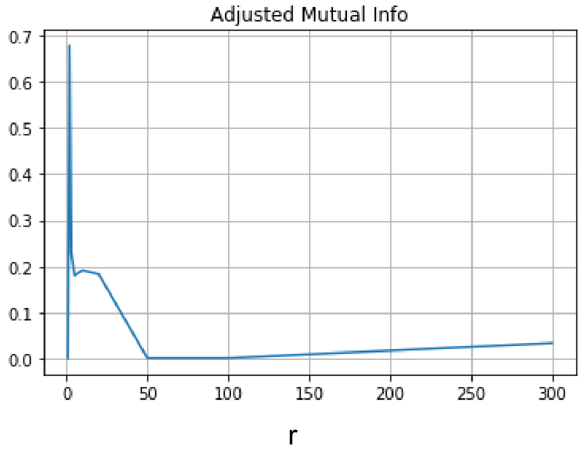
Clustering Result Contingency Matrix when r = 100

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	275	3628
True Recreational Activity	3	3976

Clustering Result Contingency Matrix when r = 300

Measurements Plots





Again, we need to look at both contingency matrix and the measure value plots to determine the best performance. As we can see in the above plots and the contingency matrix, that the best results happens when r = 2.

Discussion

How do you explain the non-monotonic behavior of the measures as r increases?
The non-monotonic behavior is expected because we can see, from our percent of variance plot, that the variance increases as the dimension increases. This is because in higher dimensions, the points have more space and do not tend to cluster together. Since we are using k-means clustering to determine the classes, we need the variance to be small and the points closer together so they can form a cluster and give better results.

Part 4

In this part, we visualize our result on a 2 dimensional plan and color coding the classes. In addition, we try different methods like normalizing and taking the logarithm of our dataset.

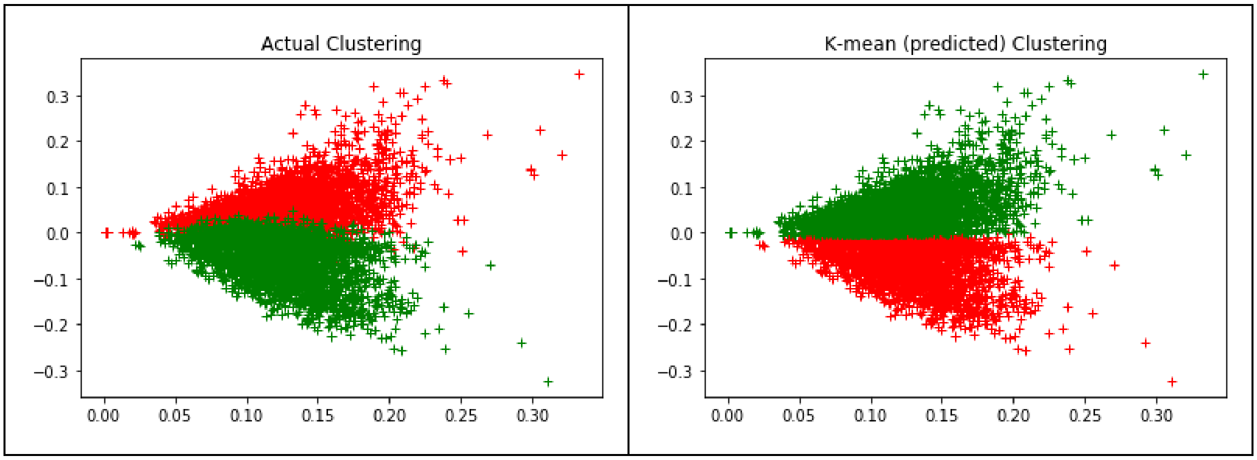
Results for a (SVD)

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3663	240
True Recreational Activity	372	3607

Clustering Result Contingency Matrix for best dimension r = 2

As discussed in the previous section, our best results happened when our dimension was equal to two. The exact values for the five measurements are:

- Homogeneity: 0.608
- Completeness: 0.609
- V-measure: 0.609
- Adjusted Rand-Index: 0.713
- Adjusted Mutual info score: 0.608



In the charts above, we can see how the SVD method mapped the dataset into a 2D space. The figure on the left represents the ground truth labeling and the figure on the right shows our clustering results. As we can see the division of the cluster is linear. By mapping our dataset into a 2D plane, we got better results in the measurement values than the part 2.

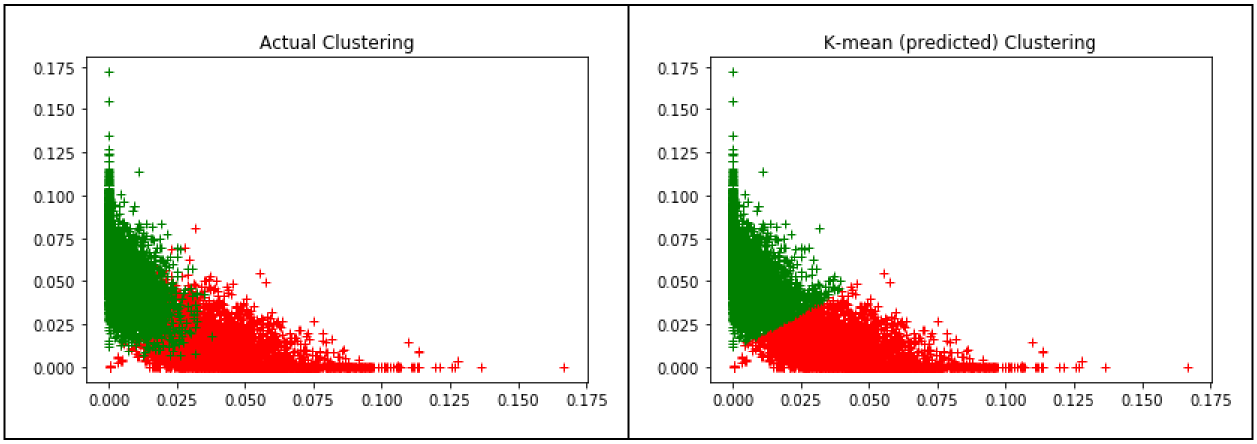
Results for a (NMF)

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3594	309
True Recreational Activity	158	3821

Clustering Result Contingency Matrix for best dimension r = 2

As discussed in the previous section, our best results happened when our dimension was equal to two. The exact values for the five measurements are:

- Homogeneity: 0.679
- Completeness: 0.680
- V-measure: 0.680
- Adjusted Rand-Index: 0.777
- Adjusted Mutual info score: 0.679



NMF maps only positive values, which gave us slightly better measured value results than the SVD mapping. In the figures above, we can compare the the ground truth clustering (left figure) and the k-mean clustering (right figure). Again, we can see that the k-mean clustering is linear.

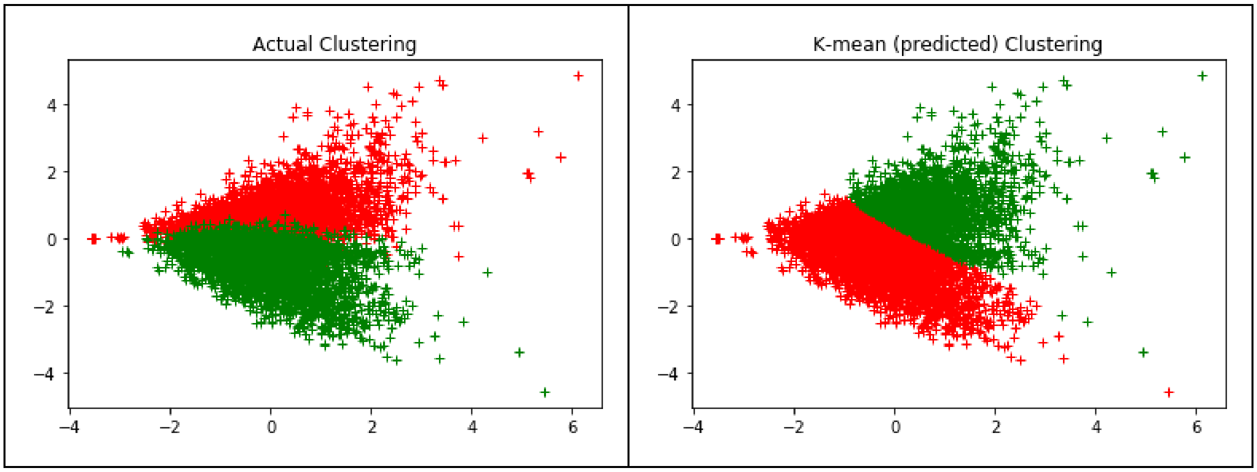
Results for b: Normalizing Feature (SVD)

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	1709	2194
True Recreational Activity	3735	244

Clustering Result Contingency Matrix for best dimension r = 2

As discussed in the previous section, our best results happened when our dimension was equal to two. The exact values for the five measurements are:

- Homogeneity: 0.235
- Completeness: 0.263
- V-measure: 0.248
- Adjusted Rand-Index: 0.254
- Adjusted Mutual info score: 0.235



We can see that the clustering results (right figure) worsen when normalizing the dataset after reducing the dimensions using SVD. From the figure, we can see that when normalizing, the method to predict the label becomes a line with a negative slope (unlike the previous results which was a zero-sloped line).

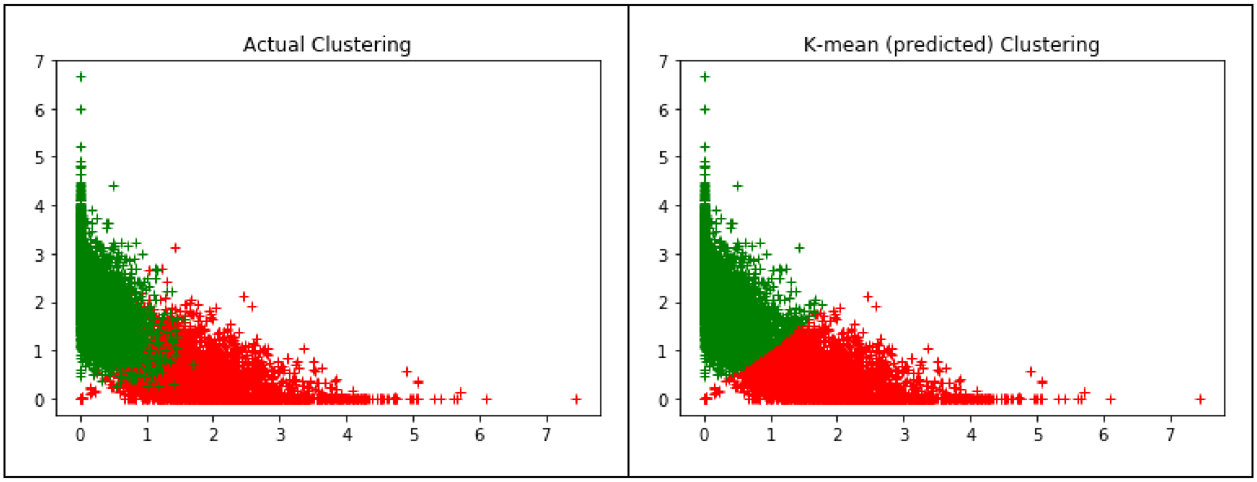
Results for b: Normalizing Feature (NMF)

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	1709	2194
True Recreational Activity	3735	244

Clustering Result Contingency Matrix for best dimension r = 2

As discussed in the previous section, our best results happened when our dimension was equal to two. The exact values for the five measurements are:

- Homogeneity: 0.683
- Completeness: 0.686
- V-measure: 0.684
- Adjusted Rand-Index: 0.773
- Adjusted Mutual info score: 0.683



From the results above, we can see that normalizing the data after using NMF reduction gave a slightly better results. The division is a positive sloped line just like the previous results. From these results, it seems that normalizing the data set is best when you need a non-zero slope line.

Results for b: Non-Linear Transformation (NMF)

In this section, we try to non-linearly transform our data only after reducing it using the NMF method. To non-linearly transform the data, we take the logarithm of the whole dataset. We did not attempt to use the SVD because it has negative values which cannot be logged.

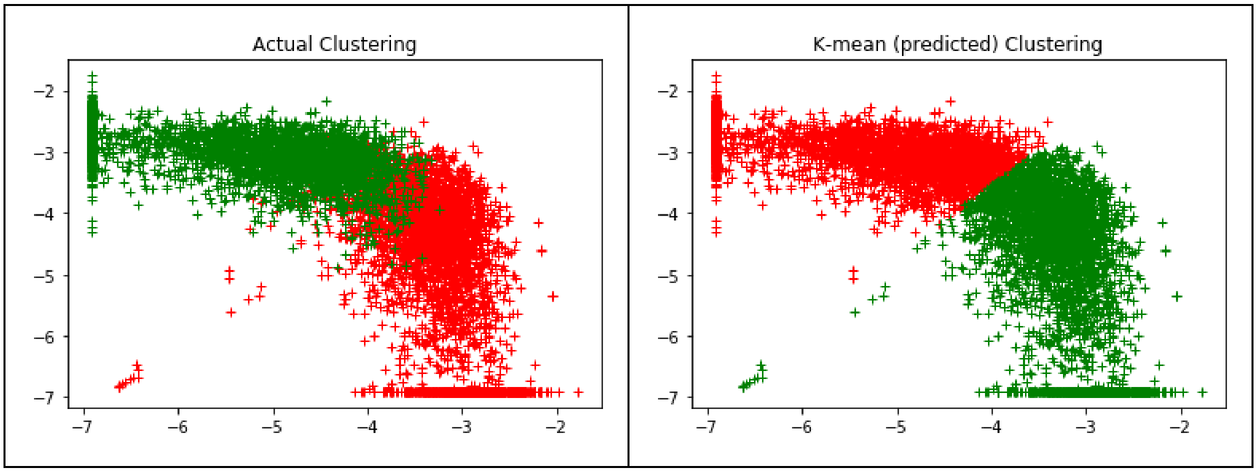
In this section, we had to add a constant to the class whose value is zero since we cannot take the log of zero. Our first thought was to add a really small number to the data set so it would not impact the solution. However, we found that adding the constant 0.001 gave the best clustering solution.

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3691	212
True Recreational Activity	183	3796

Clustering Result Contingency Matrix for best dimension r = 2

As discussed in the previous section, our best results happened when our dimension was equal to two. The exact values for the five measurements are:

- Homogeneity: 0.713
- Completeness: 0.713
- V-measure: 0.713
- Adjusted Rand-Index: 0.810
- Adjusted Mutual info score: 0.713



As we can see that the non-linear transformation improved our results.

Can you justify why logarithm transformation may increase the clustering results?
All the data points are independent of each other and do not have an initial relationship with each other. Therefore, that already makes our dataset non-linear. Using a logarithm transformation increases our result because it will transform our non-linear dataset to be solved by our linear problem (k-means).

Results for b: Normalize then Non-Linear Transformation (NMF)

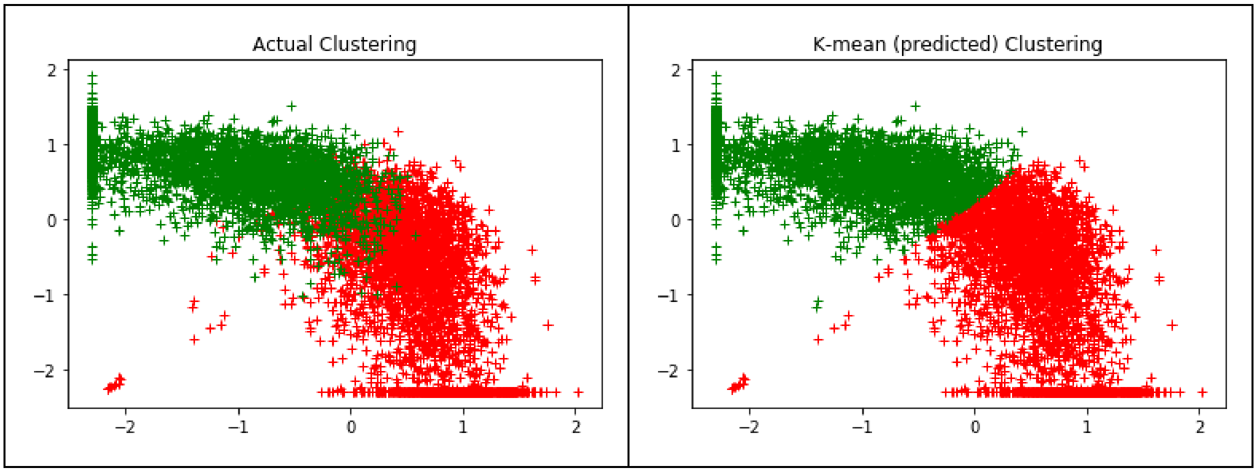
In this section, we try the combination of normalizing our dataset then non-linearly transforming it. Similarly to the last section, we needed to add a constant to our data set since we cannot take the log of zero. We found that adding the constant 0.1 gave us the best clustering results.

	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3661	242
True Recreational Activity	162	3817

Clustering Result Contingency Matrix for best dimension r = 2

As discussed in the previous section, our best results happened when our dimension was equal to two. The exact values for the five measurements are:

- Homogeneity: 0.709
- Completeness: 0.710
- V-measure: 0.710
- Adjusted Rand-Index: 0.805
- Adjusted Mutual info score: 0.709



As we can see, combining the transformation gave us similar results to just a logarithm transformation.

Results for b: Non-Linear Transformation then Normalize (NMF)

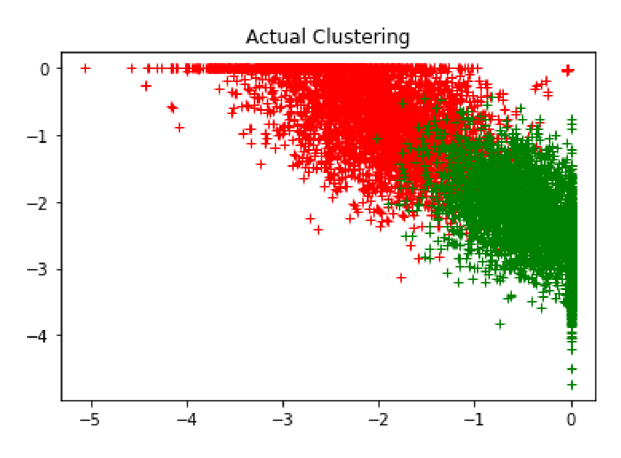
In this section, we try the combination of normalizing our dataset then non-linearly transforming it. Similarly to the last section, we needed to add a constant to our data set since we cannot take the log of zero. We found that adding the constant 0.1 gave us the best clustering results.

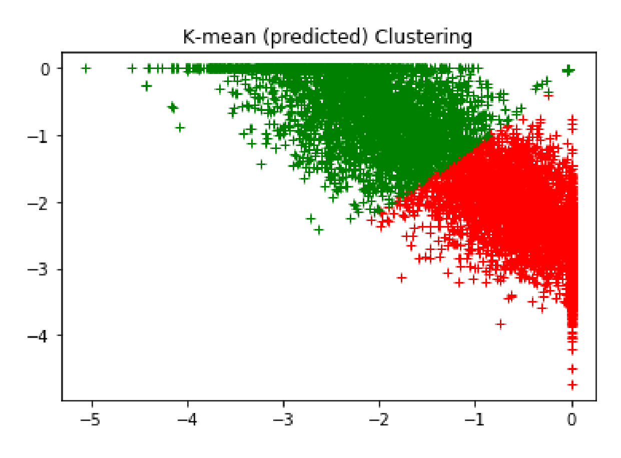
	Predicted Computer Technology	Predicted Recreational Activity
True Computer Technology	3517	386
True Recreational Activity	102	3877

Clustering Result Contingency Matrix for best dimension r = 2

As discussed in the previous section, our best results happened when our dimension was equal to two. The exact values for the five measurements are:

- Homogeneity: 0.678
- Completeness: 0.681
- V-measure: 0.680
- Adjusted Rand-Index: 0.768
- Adjusted Mutual info score: 0.678





The results of this combination where not as good as the previous sections.

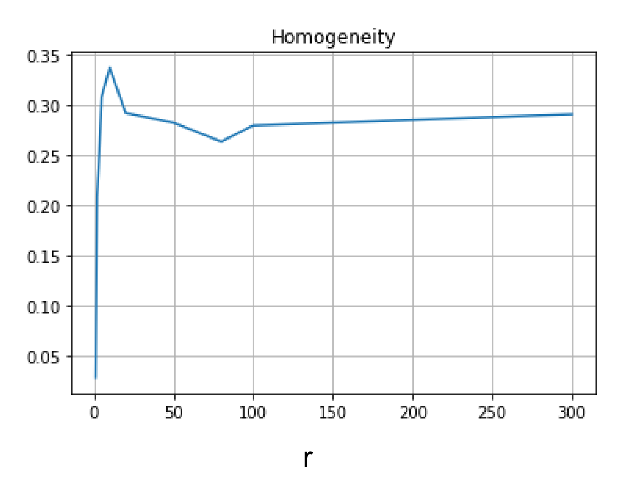
Part 5

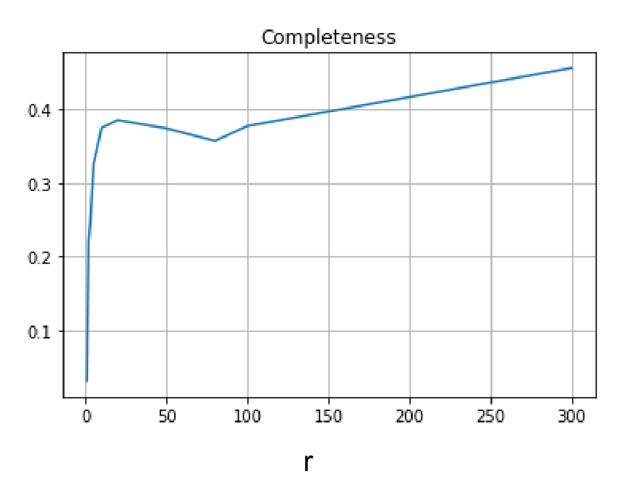
In this part, we want to evaluate the purity of our results for 20 sub-classes vs using 2 classes. We needed to test out different dimensions and report the best one. We still use the same parameters as in part 1.

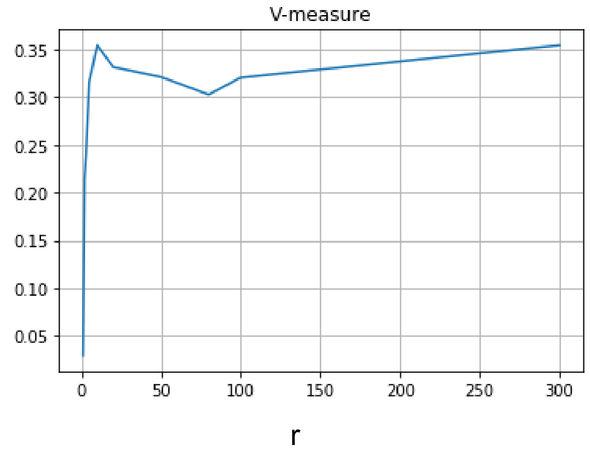
Multi-class Clustering (SVD)

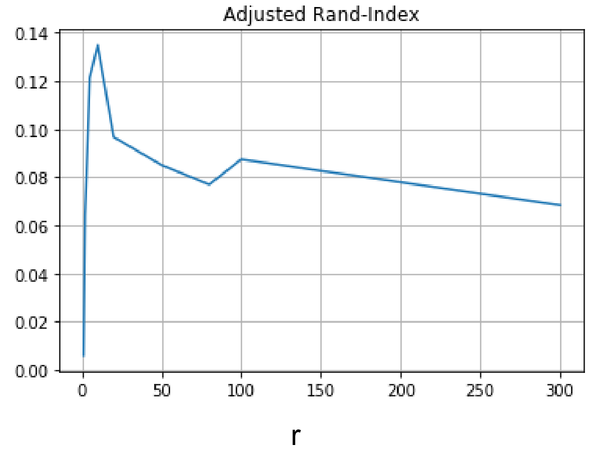
Results for Testing Different Dimensions

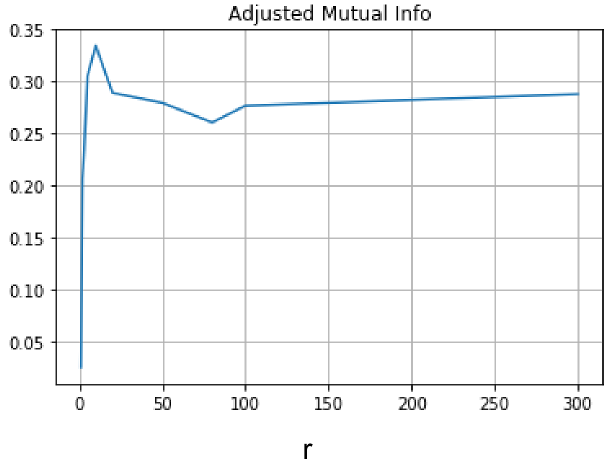
In this section, we tested out different dimension to determine which has the best results. We decided to use the r values [1, 2, 3, 5, 10, 20, 50, 80, 100, 300].









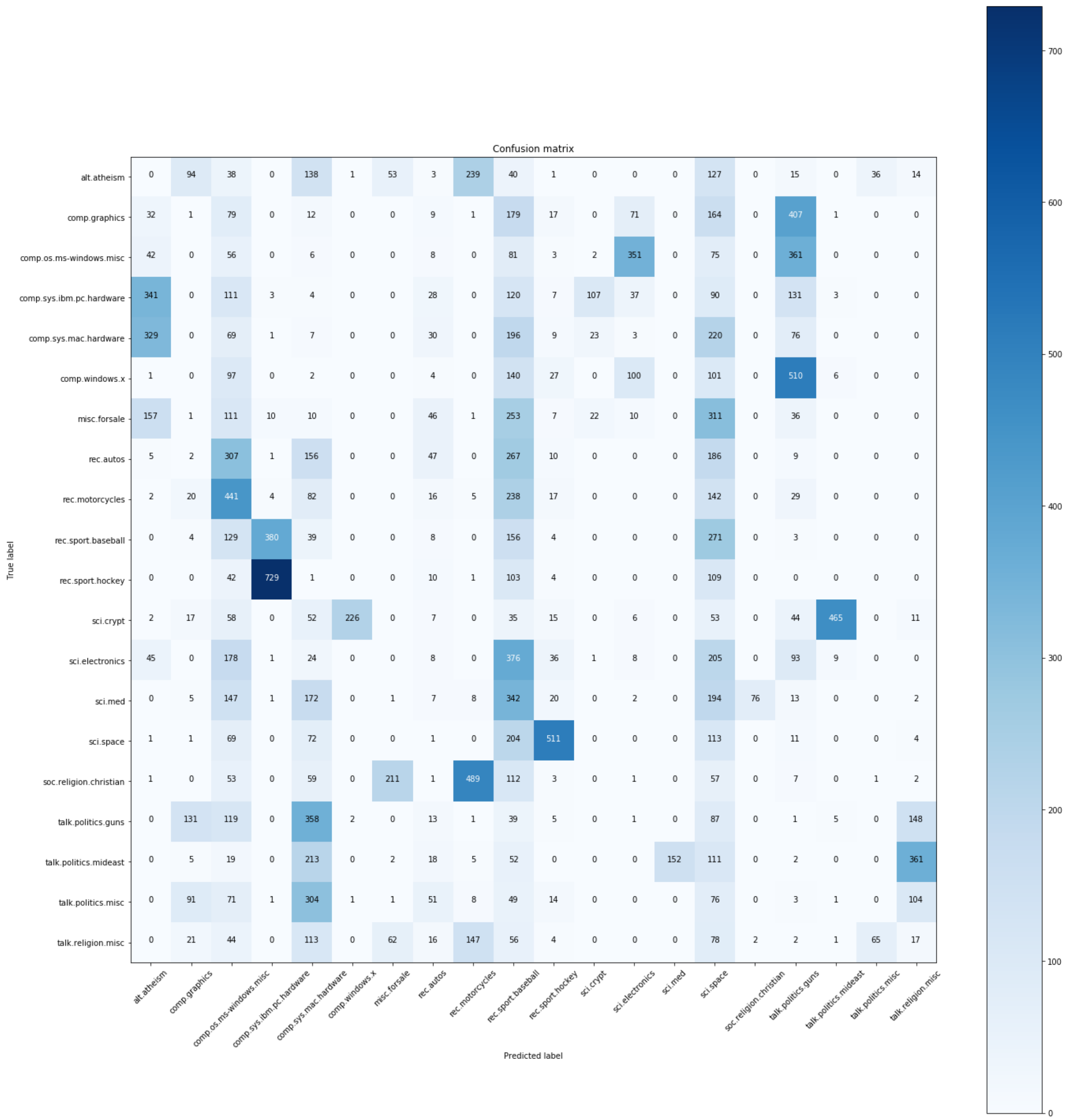


Results for the best dimension reduction

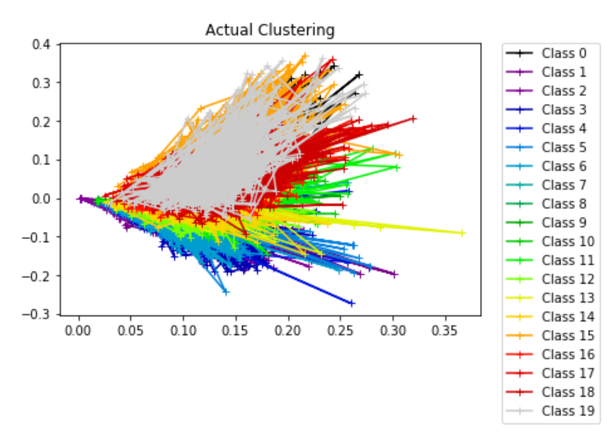
From the plots above, we decided the best results happens when the dimensions are reduced to 10. We found the confusion matrix and the five measurements. We also reduced the dimensions to two to visualize the plots on a 2D plane.

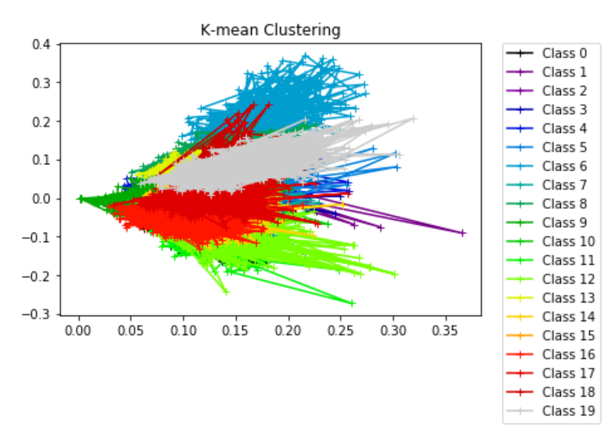
The five measurements are:

- Homogeneity: 0.318
- Completeness: 0.367
- V-measure: 0.341
- Adjusted Rand-Index: 0.129
- Adjusted Mutual info score: 0.316



Confusion Matrix when r = 10



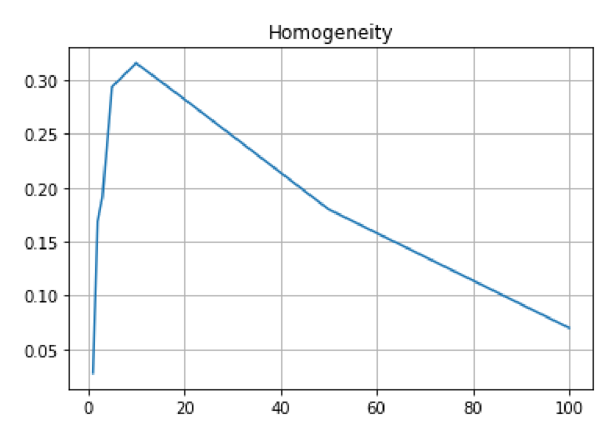


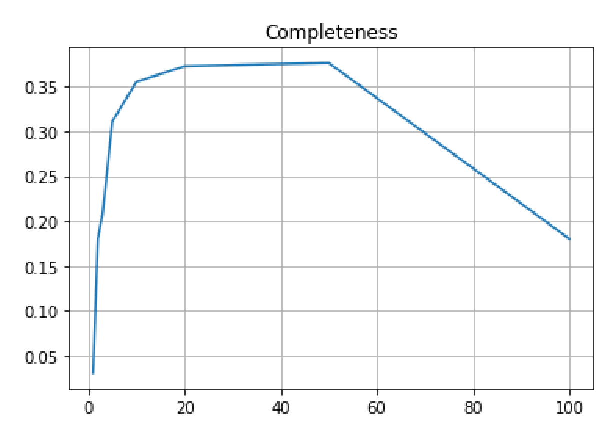
As we can see, our results are not the best. All the five measurements are really low. Having a larger class number is harder to cluster because they probably do not map nicely for the k-means algorithm. As we see in the figure above, we see that the ground truth clusters (on the left) overlap each other in the 2D sense. The bad results can also be due to not balanced datal.

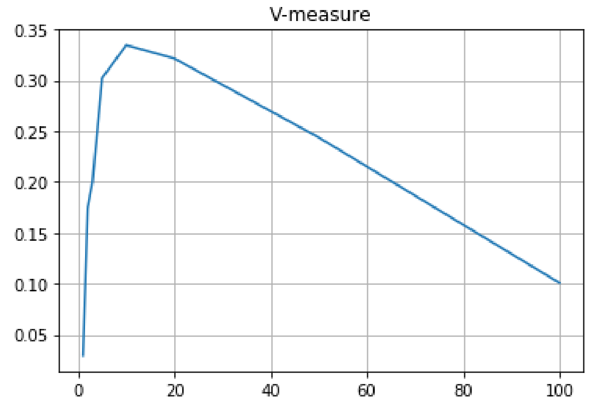
Multi-class Clustering (NMF)

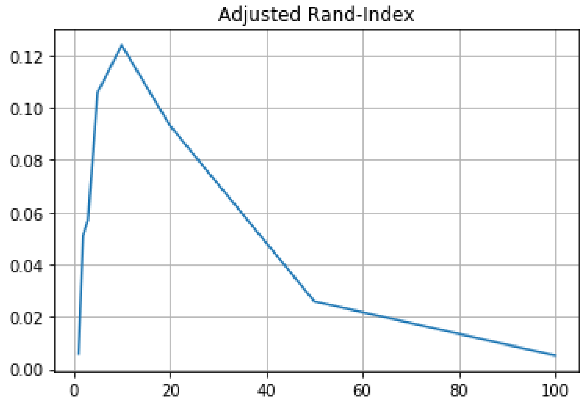
Results for Testing Different Dimensions

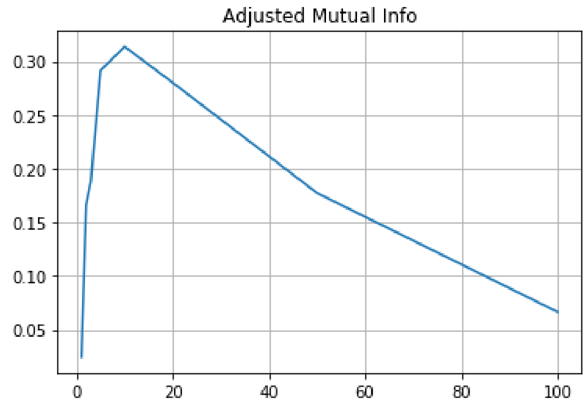
In this section, we tested out different dimension to determine which has the best results. We decided to use the r values [1, 2, 3, 5, 10, 20, 50, 80, 100]. We first did a sweep without using r = 300 because the run time was really long. Looking at the results, it seemed the purity went down as the dimension went up. Therefore we just stopped at these values.











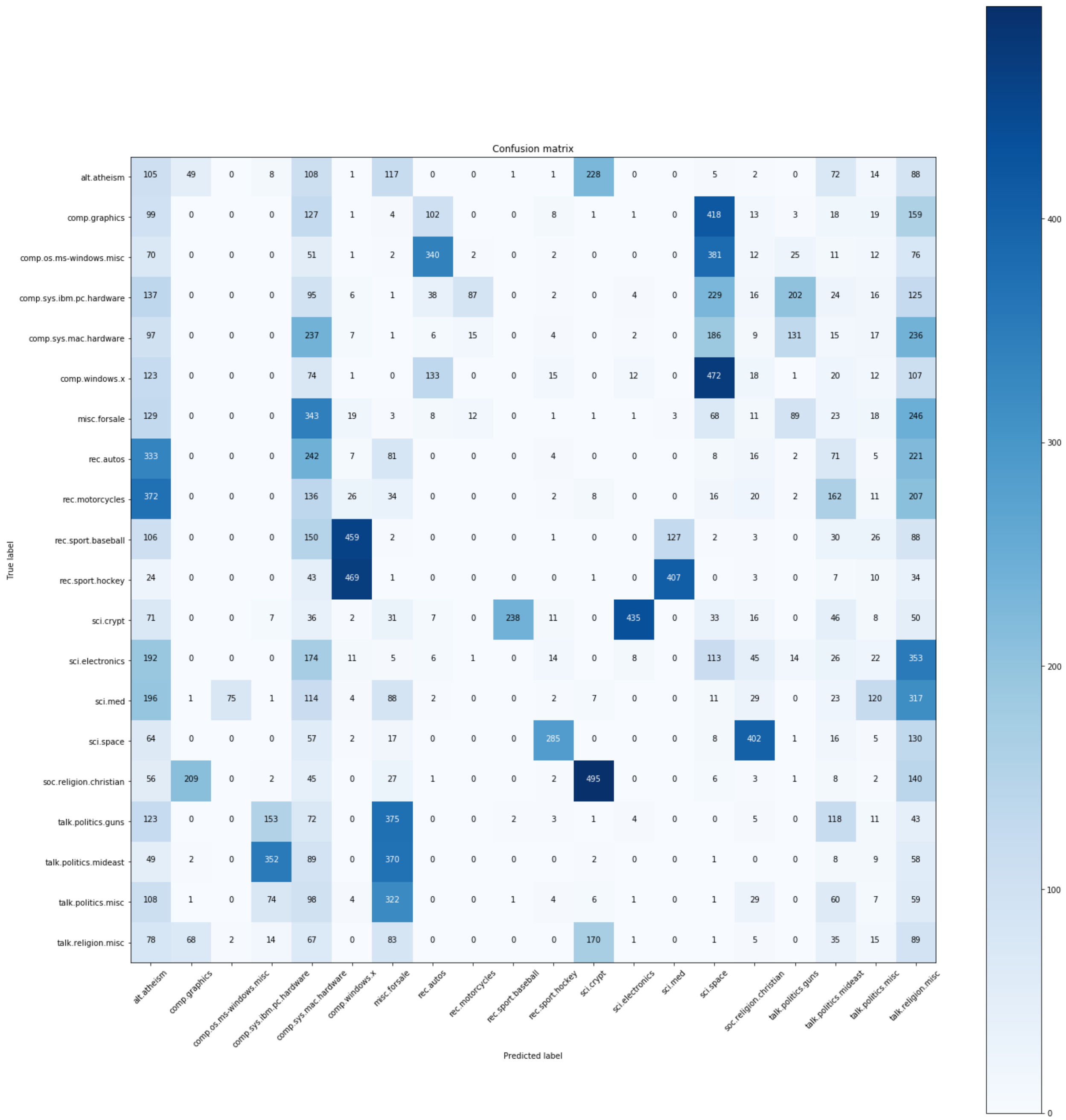
From the results above, we decided that we get the best clustering happened when r = 10.

Results for the best dimension reduction

Just as stated before, we decided the best results happens when the dimensions are reduced to 10. We found the confusion matrix and the five measurements. We also reduced the dimensions to two to visualize the plots on a 2D plane.

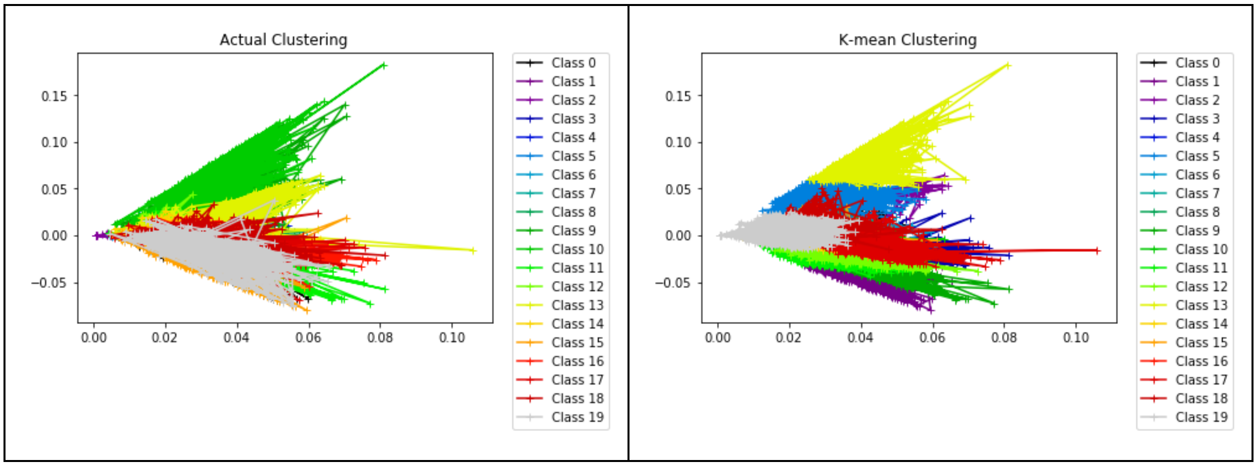
The five measurements are:

- Homogeneity: 0.317
- Completeness: 0.357
- V-measure: 0.335
- Adjusted Rand-Index: 0.124
- Adjusted Mutual info score: 0.314



	alt.atheism	comp.graphics	comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware	comp.sys.mac.hardware	comp.windows.x	misc.forsale	rec.autos	rec.motorcycles	rec.sport.baseball	rec.sport.hockey	sci.crypt	sci.electronics	sci.med	sci.space	soc.religion.christian	talk.politics.guns	talk.politics.mideast	talk.politics.misc	talk.religion.misc
alt.atheism	105	49	0	8	108	1	117	0	0	1	1	228	0	0	5	2	0	72	14	88
comp.graphics	99	0	0	0	127	1	4	102	0	0	8	1	1	0	418	13	3	18	19	159
comp.os.ms-windows.misc	70	0	0	0	51	1	2	340	2	0	2	0	0	0	381	12	25	11	12	76
comp.sys.ibm.pc.hardware	137	0	0	0	95	6	1	38	87	0	2	0	4	0	229	16	202	24	16	125
comp.sys.mac.hardware	97	0	0	0	237	7	1	6	15	0	4	0	2	0	186	9	131	15	17	236
comp.windows.x	123	0	0	0	74	1	0	133	0	0	15	0	12	0	472	18	1	20	12	107
misc.forsale	129	0	0	0	343	19	3	8	12	0	1	1	1	3	68	11	89	23	18	246
rec.autos	333	0	0	0	242	7	81	0	0	0	4	0	0	0	8	16	2	71	5	221
rec.motorcycles	372	0	0	0	136	26	34	0	0	0	2	8	0	0	16	20	2	162	11	207
rec.sport.baseball	106	0	0	0	150	459	2	0	0	0	1	0	0	127	2	3	0	30	26	88
rec.sport.hockey	24	0	0	0	43	469	1	0	0	0	0	1	0	407	0	3	0	7	10	34
sci.crypt	71	0	0	7	36	2	31	7	0	238	11	0	435	0	33	16	0	46	8	50
sci.electronics	192	0	0	0	174	11	5	6	1	0	14	0	8	0	113	45	14	26	22	353
sci.med	196	1	75	1	114	4	88	2	0	0	2	7	0	0	11	29	0	23	120	317
sci.space	64	0	0	0	57	2	17	0	0	0	285	0	0	0	8	402	1	16	5	130
soc.religion.christian	56	209	0	2	45	0	27	1	0	0	2	495	0	0	6	3	1	8	2	140
talk.politics.guns	123	0	0	153	72	0	375	0	0	2	3	1	4	0	0	5	0	118	11	43
talk.politics.mideast	49	2	0	352	89	0	370	0	0	0	0	2	0	0	1	0	0	8	9	58
talk.politics.misc	108	1	0	74	98	4	322	0	0	1	4	6	1	0	1	29	0	60	7	59
talk.religion.misc	78	68	2	14	67	0	83	0	0	0	0	170	1	0	1	5	0	35	15	89

Confusion Matrix when r = 10



As we can see, our results are not the best. All the five measurements are really low. Having a larger class number is harder to cluster because they probably do not map nicely for the k-means algorithm. As we see in the figure above, we see that the ground truth clusters (on the left) overlap each other in the 2D sense. The bad results can also be due to unbalanced data.