Text Clustering Project

Introduction

The dataset contains text data from Siena courses, we will use clustering to decide which courses belong in which group. We will specify a different number of clusters, 3 clusters, 33 clusters and 57 clusters. After which we will analyze the optimal number of clusters for this dataset. After extracting the appropriate data, we will use k-means, agglomerative clustering and LDA clustering algorithms. After obtaining these results we will use metrics such as a silhouette score and adjusted rand score to see which algorithm gives us the best results and discuss findings related to each algorithm.

Data Preprocessing and Clustering

The data we have is in a txt file, after storing it in a variable and removing the columns of index we do not need, we use the CountVectorizer() method to transform the data appropriately and fitting and transforming the “decriptions.txt” file. This process sets the stage for the data which will be used for clustering.

K-means

In KMeans clustering we take our data points and group them in k (being the hyperparameter) clustering based on the distance from each data point to the centroid. We use the KMeans method and fit our data to the model and specify the 3 data clusters we need.

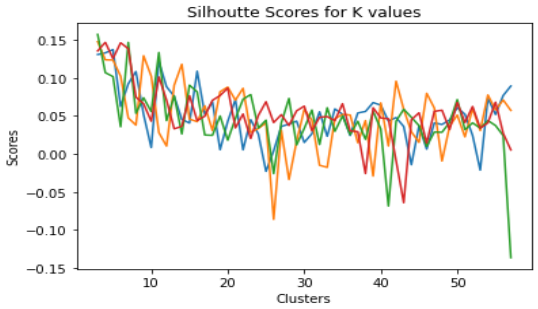


Figure 1.1

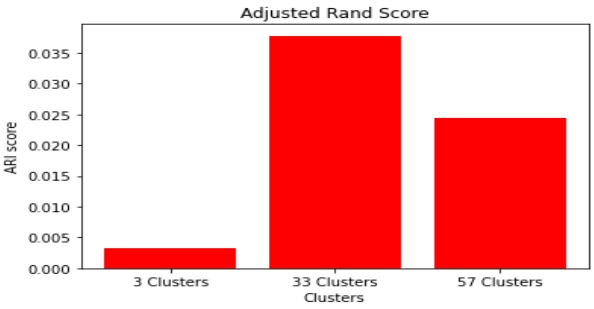


Figure 1.2

Figure 1.1 showcases the silhouette scores of KMeans clustering. While there are outliers, they are all within or close to 0. This is means that the number of clusters ranging from 3 to 57 are somewhat “optimal” as they are close to 0 and this signifies that the cluster is very close and sharing some of other cluster’s decision boundaries. The adjusted random scores for all 3 clusters are under 0.65 therefore it is not a very accurate clustering algorithm.

Agglomerative Clustering

Agglomerative Clustering is a form of Hierarchical Clustering which uses a bottom-up approach and greedily adds data points to its cluster. Here we use the agglomerative clustering method and yields more consistent results compared to KMeans.

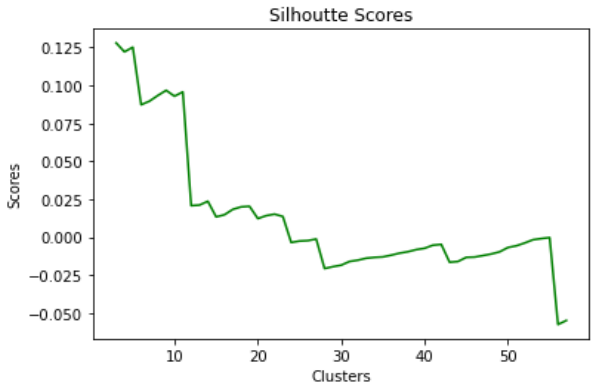


Figure 1.3

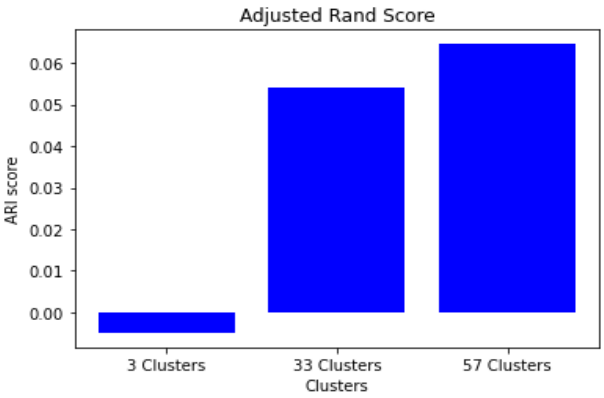


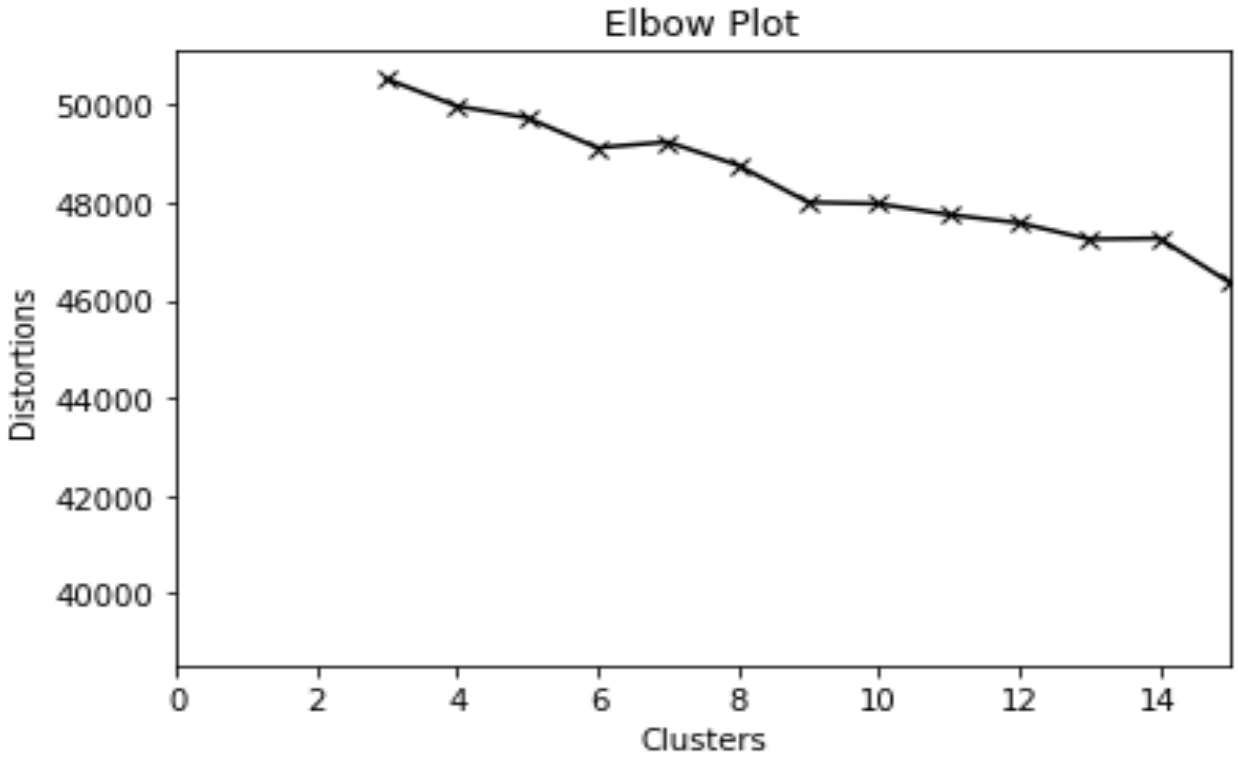
Figure 1.4

Figure 1.3 showcases the silhouette scores from 3 to 58 clusters, the algorithm yields a good score until the number of clusters increases the silhouette coefficient decreases. It is clear from Figure 1.3 for this clustering algorithm the number of clusters must be low to decrease outliers. As shown in Figure 1.4 we can see the adjusted rand index returns poor recovery therefore also not a very accurate clustering algorithm.

Finding Optimal Clusters & LDA

K-Means

Figuring out the optimal number of clusters for KMeans is difficult from the silhouette score so we must take a different approach. Instead, we will utilize the “Elbow Method” to determine the optimal number of clusters. We take the inertias from the KMeans model with different clustering size and see where we see an elbow.



We can see that the elbow occurs around 6 clusters and such after running the appropriate tests it returned a silhouette score of 0.079 which is decent as there are some small number of outliers. This could show that number of schools should be 6 rather than 3. We can dissect this further with topic matching when we use LDA.

Agglomerative

In Agglomerative we saw consistent scores in the silhouette tests and as such we can just observe Figure 1.3 to see what number of clusters yielded the best silhouette score. We will be looking for an elbow same as with KMeans inertia. We can see it occurring at around 6 clusters and yielding a consistent silhouette score of 0.087 meaning again a small number of outliers but most data points are in acceptable decision boundaries. Next, we will use LDA to select 6 topics from the dataset to see how we could divide them up.

After running LDA and selecting 6 topics we see the courses can be divided up based on different keywords like “writing” and “analysis”. Both these patterns indicate that we can further break down the number of schools.