Tweets clustering

Using k-means clustering algorithms Implemented from scratch

(Project report)

Preprocessing data (the process of cleaning data):

1- By putting data in excel sheets we divided it into three columns
-Id - date - tweets

2- we used only data in tweets column, retrieved it and put it in a list

3- looping on each string element (tweet) in the list, we do some functionalities to make changes on each element to be able to use it in the clustering algorithm later and these changes are

- Remove every "#", "@" symbols in the tweet using replace function (built-in function).
- Remove every URL in the tweet (http\....) by using re module and regular expression "http\S+" with sub function.
- Convert each character to the lowercase using lowercase function.
- Remove all the punctuation signs.
- Remove all the extra spaces out of the previous functionalities by split them including these spaces and join them together to be one string separated by string separator (one space).
- Turn the string text into list of words and include it in a list to be used in the clustering process.

After all these functionalities, we get list of clean data that easy to use in the k-means algorithm.

K-means algorithm:

There are four steps to perform this algorithm:

1. Initialize centroids of the clusters randomly according to the number of clusters we want to create, and it must be chosen from data we classify.
2. Assign the tweets to their clusters.
3. Update the centroids of the clusters.
4. Repeat the last two steps until we get the centroids converged.

By The last updated centroids and clusters, we compute SSE (sum squared error).

Assign tweets to clusters:

The only way we can classify the tweets is by measuring the similarities between them and each centroid we have and the centroid (cluster)where the most similarities found, we assign the tweet.

Similarity measurement method (Jaccard distance):

The measurement of distances determined by dissimilarity between each tweet and another. We will compare distances with each other and determine the minimum distance between the tweets (more similarity).

We use Jaccard to measure the similarity between two strings as we can't use the normal get distance method between two points.

Jaccard distance = 1 – (Len(intersection(tweet1,tweet2))/Len(union(tweet1,tweet2)))

Update clusters 'centroids:

the centroids we got in the clusters may not be the best case, so we perform the Jaccard method between each tweet with the other tweets in the same cluster and add them together then repeat this loop for each tweet in this cluster and find the average (minimum) sum of distances between that tweet and the others with it in the cluster and set it the new center of this cluster.

Check Convergence:

Checking the convergence in one of the two ways to end the clustering algorithms.

Convergence splited into two parts:

- Check maintaining of the number of centroids the same after and before update method.
- Check if the centroids don't change after and before the update method.

SSE "sum of squared error":

With each experiment we perform on data by k-means algorithms, we get percentage of error. That's how we see the improvement in the results of error we get by increasing the number of clusters.

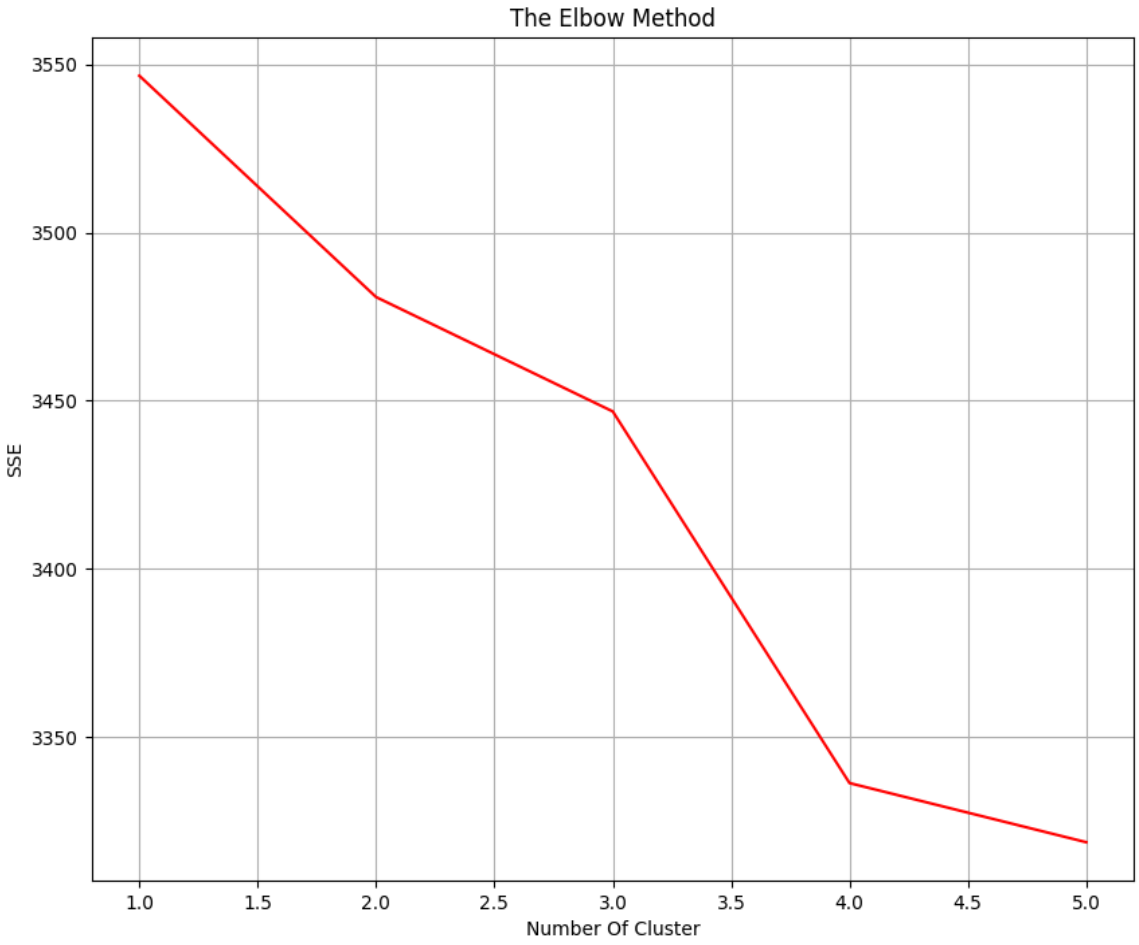
Testing the clustering algorithms

Test on "bbchealth" file:

K = 1 "converged"	SSE = 3546.618269	cluster1: 3929
K = 2 "converged"	SSE = 3480.831409	cluster1: 2306 tweets cluster2: 1623 tweets
K = 3 "converged"	SSE = 3446.825753	cluster1: 1393 tweets cluster2: 1611 tweets cluster3: 925 tweets
K = 4 "converged"	SSE = 3336.369068	cluster1: 963 tweets cluster2: 717 tweets cluster3: 1033 tweets cluster4: 1216 tweets
K = 5 "converged"	SSE = 3318.369068	cluster1: 981 tweets cluster2: 480 tweets cluster3: 878 tweets cluster4: 687 tweets cluster5: 903 tweets

:

Graph between the SSE and number of clusters “k”

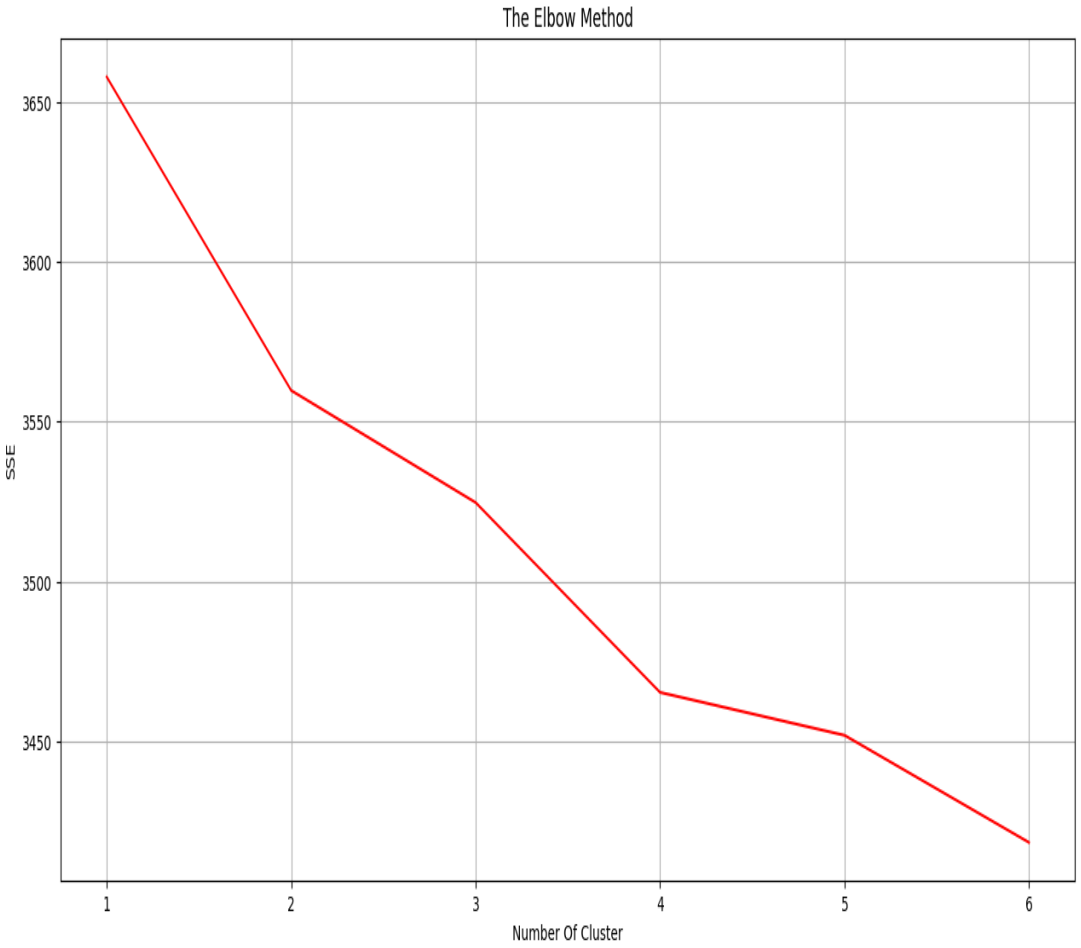


- using the elbow method, we can find the optimal or the minimum number of clusters we can divide the data into.

Here the optimal number of clusters = 4.

TEST ON “cnnhealth” file

K = 1 “converged”	SSE = 3657.76829	Cluster1: 4061 tweets
K = 2 “converged”	SSE = 3559.77021	Cluster1: 1563 tweets Cluster2: 2498 tweets
K = 3 “converged”	SSE = 3524.75735	Cluster1: 867 tweets Cluster2: 2179 tweets Cluster3: 1015 tweets
K = 4 “converged”	SSE = 3465.41880	Cluster1: 1289 tweets Cluster2: 1150 tweets Cluster3: 541 tweets Cluster4: 1081 tweets
K = 5 “converged”	SSE = 3452.01322	Cluster1: 1208 tweets Cluster2: 942 tweets Cluster3: 788 tweets Cluster4: 481 tweets Cluster5: 642 tweets
K = 6 “converged”	SSE = 3418.52879	Cluster1: 677 tweets Cluster2: 732 tweets Cluster3: 553 tweets Cluster4: 514 tweets Cluster5: 925 tweets Cluster6: 660 tweets



Graph between the SSE and number of clusters “k”

- Using the elbow method, we find that the optimal number of clusters we can assign data to may be 4 or 5