1. Name: Chaoran Lei
   SJSU ID: 015264119

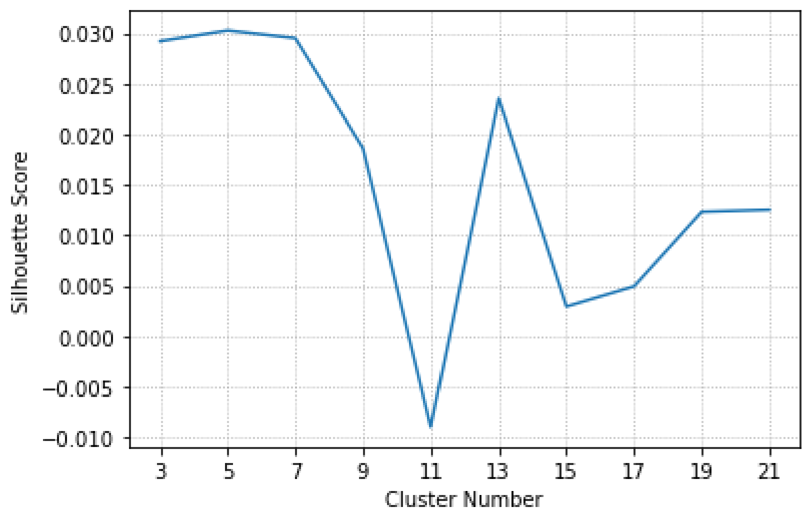
2. In this project, we need to process a slightly processed dataset which already labeled the each words of many documents with unique numbers, and also frequency counted. To deal with dataset which is a huge sparse matrix, csr matrix would save a lot of space.
   The most important part of this project is dimension reduction and bisecting k-means algorithm implementation.
   To approach the bisecting k-means, I need to create it based on k-means algorithm. For the k-means part, I didn't use the existed sklearn KMeans methods. I create several functions to help me go through it.
   First, I shuffle the matrix to get the random centroids. And then calculate the similarity between matrix and the centroids. Then I use the similarity matrix to get the two clusters. After this step, I recalculate the new centroid until reach the iteration cycle I set later in K-means function.
   For the bisecting k-means class part. After I got two sub-clusters from k-means function, I calculate the SSE of each of them. The cluster with larger SEE will be took out and ready to be split in next loop. The cluster with smaller SSE will be added to existed clusters. Then repeat this progress until I got 7 clusters.

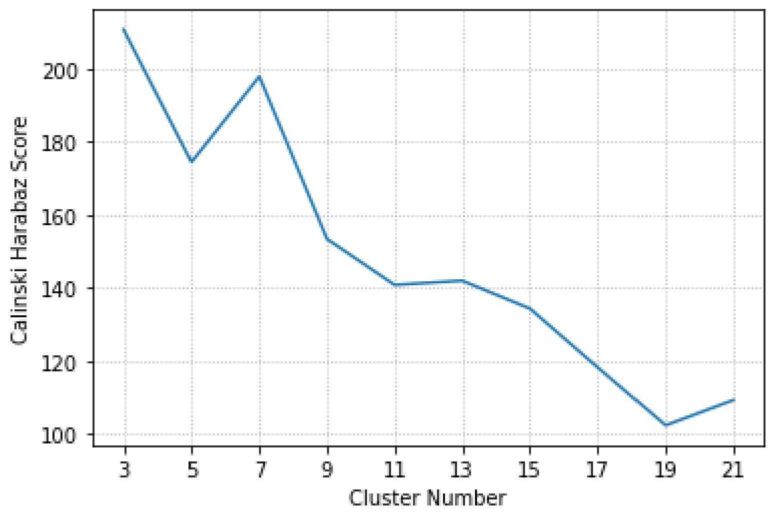
```
Initialize a list of clusters
Initialize a list of initialcluster
for each number of input matrix index:
    initialcluster + each number
clusters + initialcluster

while len(clusters) < k:
    dropClusterIndex = SSE(matrix, clusters)
    droppedCluster = clusters[dropClusterIndex]
    cluster1, cluster2 = Kmeans(matrix[droppedCluster,:], numberOfIterations)
    delete clusters[dropClusterIndex]
    Initialize a list of actualCluster1
    Initialize a list of actualCluster2
    for index in cluster1:
        actualCluster1 + droppedCluster[index]
    for index in cluster2:
        actualCluster2 + droppedCluster[index]
    clusters + actualCluster1
    clusters + actualCluster2

Initialize a list of labels # result will be returned
for index, cluster in enumerate(clusters):
    for idx in cluster:
        labels[idx] = index + 1
```

3. Evaluation metrics
   I used two sklearn build in methods to validate the result. First one is silhouette score, it can be used to determine the degree of separation between clusters. The other one is Calinski-Harabasz score. It is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters. The best silhouette score for k = 7 was about 0.03. The best Calinski-Harabasz score was about 210. Here is the result with k from 3 to 21 in steps of 2.





4. For dimension reduction, I first used TF-IDF（Term Frequency–Inverse Document Frequency）to process the data. This is a common term weighting scheme in information retrieval that has also found good use in document classification. Then I used TruncatedSVD to reduce the dimension to 256. I tried several other dimensions but this one was the best. Both methods are built in in sklearn.
   I tried sparse PCA method at first, but the result was not good. I also tried to do some data cleaning before dimension reduction. However due to the lack of enough memory space it doesn’t work. I believe if I can do more pre-processing on the data set. The result will be better.

References：

https://towardsdatascience.com/bisecting-k-means-algorithm-clustering-in-machine-learning-1bd32be71c1c
https://medium.com/@afrizalfir/bisecting-kmeans-clustering-5bc17603b8a2
https://github.com/AchillesnoMY/K-means-and-Bisecting-K-means-Method
https://github.com/sowmyagowri/Text-Clustering