WEB MINING LAB

16BCE1184
SOUMOK DUTTA

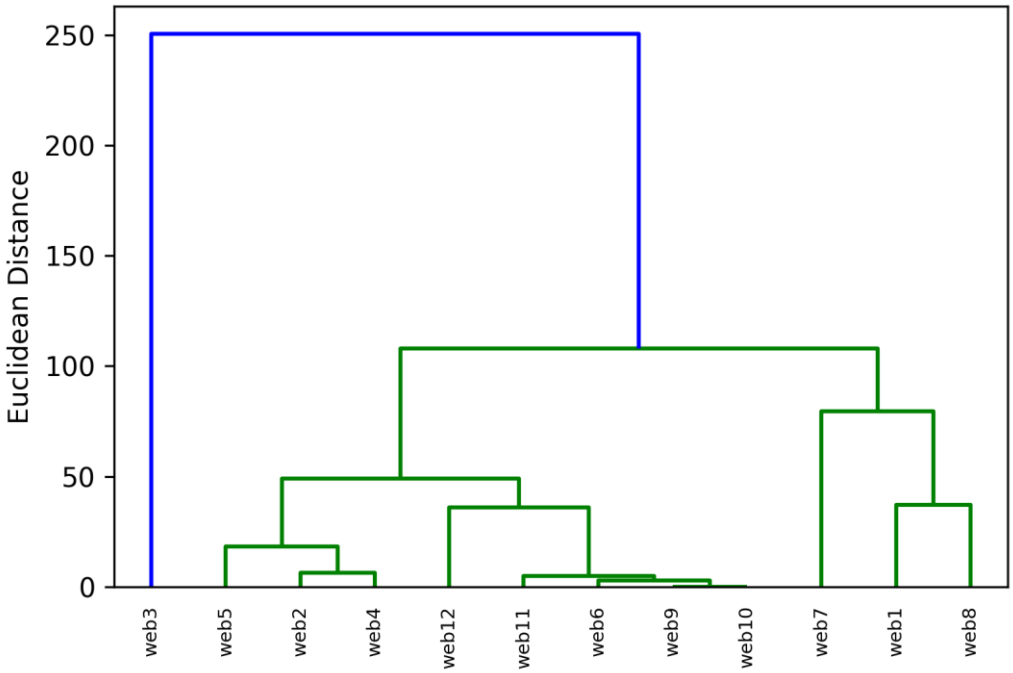
Q . Document hierarchical indexing with cosine distance formula.

CODE:

```
import string
import pandas as pd
import math
import matplotlib.pyplot as plt

class document_clustering(object):
    def __init__(self, file_dict, word_list):
        self.file_dict = file_dict
        self.word_list = word_list

    def tokenize_document(self, document):
        terms = document.lower().split()
        return [term.strip(string.punctuation) for term in terms]

    def create_word_listing(self):
        self.listing_dict_ = {}

        for id in self.file_dict:
            temp_word_list = []
            f = open(self.file_dict[id], 'r')
            document = f.read()
            terms = self.tokenize_document(document)
            for term in self.word_list:
                temp_word_list.append(terms.count(term.lower()))
            self.listing_dict_[id] = temp_word_list

        print('Word listing of each document')
        for id in self.listing_dict_:
            print('%d:  %s' % (id, self.listing_dict_[id]))

    def create_document_matrix(self):
```

```
        self.labels_ = ['doc%d' % (id) for id in self.file_dict]
        main_list = []
        for id1 in self.file_dict:
            temp_list = []
            for id2 in self.file_dict:
                dist = 0
                for term1, term2 in zip(self.listing_dict_[id1], self.listing_dict_[id2]):
                    dist += (term1-term2)**2
                temp_list.append(round(math.sqrt(dist), 4))
            main_list.append(temp_list)

        self.distance_matrix_ = pd.DataFrame(main_list, index = self.labels_, columns = self.labels_)
        print('\nDistance Matrix')
        print(self.distance_matrix_)

    def cluster(self):
        from scipy.cluster.hierarchy import linkage
        row_cluster = linkage(self.distance_matrix_.values,
                              method = 'complete',
                              metric = 'cosine')
        from scipy.cluster.hierarchy import dendrogram
        dn = dendrogram(row_cluster, labels = self.labels_)
        plt.ylabel('Euclidean Distance')
        plt.xticks(rotation = 90)
        plt.savefig('dendrogram1.png', dpi = 300)
        plt.show()

file_dict = {1: './documents/doc1.txt',
             2: './documents/doc2.txt',
             3: './documents/doc3.txt',
             4: './documents/doc4.txt',
             5: './documents/doc5.txt',
             6: './documents/doc6.txt',
             7: './documents/doc7.txt',
             8: './documents/doc8.txt',
             9: './documents/doc9.txt'}
word_list = ['Tesla', 'Electric', 'Car', 'pollution', 'de-monetisation', 'GST' ,'black money']

document_cluster = document_clustering(file_dict = file_dict, word_list = word_list)
document_cluster.create_word_listing()
document_cluster.create_document_matrix()
document_cluster.cluster()
```

OUTPUT:


Euclidean Distance
250
200
150
100
50
0
web3
web5
web2
web4
web12
web11
web6
web9
web10
web7
web1
web8