VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

NLP Task

B.Tech in Computer Science and Engineering **(CSE)**, **Fall** Semester **2020**

Name:	Abhishek Kushwaha
Registration Number:	18BCE0492
Slot:	G2
Faculty Name:	Mrs Sharmila Banu
Slot:	E2+TE2
Team:	3

As assigned by our team leader, Shivam Anand, I worked on cosine similarity between the set of texts we decided to analyse and come to a collective outcome.

Cosine Similarity is the process by which we can determine a similarity index between two documents.

Mathematically, it measures the cosine of the angle between the vectors formed by the two text documents in a multi-dimensional space.

So, the smaller the angle, more is the value of cosine and hence we can determine the uniqueness of the document with respect to another document already in database.

Cosine Similarity

```
[ ]  from sklearn.feature_extraction.text import TfidfVectorizer
     tfidf_vectorizer = TfidfVectorizer()
```

```
[ ]  import glob
     documents_list = glob.glob('./*.txt')
     documents_list
```

```
['./WhatDreamsMayCome.txt',
 './SixYearsAndCountingNew.txt',
 './GettingSaucyAboutFood.txt',
 './SixYearsAndCounting.txt',
 './TrainToNowhere.txt']
```

```
[ ]  documents_instances_list = []
     for i in range(len(documents_list)):
       with open(documents_list[i]) as e:
         documents_instances_list.append(e.read())
     tfidf_matrix = tfidf_vectorizer.fit_transform(documents_instances_list)
```

```
[ ] from itertools import combinations

    numbers = range(0, len(documents_instances_list))
    k = list(combinations(numbers, 2))
    m = lambda s: s.strip('./')
    document_map = dict(zip(numbers, list(map(m, documents_list))))
```

```
[ ] print(document_map)
    k
```

```
{0: 'WhatDreamsMayCome.txt', 1: 'SixYearsAndCountingNew.txt', 2: 'GettingSaucyAboutFood.txt', 3: 'SixYear
[(0, 1),
 (0, 2),
 (0, 3),
 (0, 4),
 (1, 2),
 (1, 3),
 (1, 4),
 (2, 3),
 (2, 4),
 (3, 4)]
```

```
[ ] from sklearn.metrics.pairwise import cosine_similarity
```

```
[ ] for i in range(len(k)):
      first_doc = k[i][0]
      second_doc = k[i][1]
      print("Document Similarity between document {}({}) and {}({}) is : \t".format(first_doc,document_map[f
```

```
Document Similarity between document 0(WhatDreamsMayCome.txt) and 1(SixYearsAndCountingNew.txt) is :
Document Similarity between document 0(WhatDreamsMayCome.txt) and 2(GettingSaucyAboutFood.txt) is :
Document Similarity between document 0(WhatDreamsMayCome.txt) and 3(SixYearsAndCounting.txt) is :
Document Similarity between document 0(WhatDreamsMayCome.txt) and 4(TrainToNowhere.txt) is :     [0.46949
Document Similarity between document 1(SixYearsAndCountingNew.txt) and 2(GettingSaucyAboutFood.txt) is :
Document Similarity between document 1(SixYearsAndCountingNew.txt) and 3(SixYearsAndCounting.txt) is :
Document Similarity between document 1(SixYearsAndCountingNew.txt) and 4(TrainToNowhere.txt) is :
Document Similarity between document 2(GettingSaucyAboutFood.txt) and 3(SixYearsAndCounting.txt) is :
Document Similarity between document 2(GettingSaucyAboutFood.txt) and 4(TrainToNowhere.txt) is :
Document Similarity between document 3(SixYearsAndCounting.txt) and 4(TrainToNowhere.txt) is :   [0.49420
```

Conclusion:

Moving on next are two ways for checking similarity in documents. Since the dataset is too small, cosine similarity index shows almost all the texts are unique with respect to each other. Also if we observe, doc-1 and doc-4 have approximately same content, hence the higher similarity value.

Hence, out of 5 text files, 3 are unique i.e.

- GettingSaucyAboutFood.txt
- TrainToNowhere.txt
- WhatDreamsMayCome.txt

And 2 files are similar and can be considered as one i.e.

- SixYearsAndCounting.txt
- SixYearsAndCountingNew.txt