**Assignment No 7**

**Code:**

import pandas as pd

import numpy as nu

import nltk

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('averaged\_perceptron\_tagger')

text="The Greatest Gold Robbery took place on the night of 15 May 1855, when a shipment of gold to Paris was stolen from the guard's van of the rail service between London and Folkestone. There were four robbers: two employees of the rail company, a former employee and Edward Agar, a career criminal. They took wax impressions of the keys to the train safes and made copies. One of them ensured he was on guard duty when a shipment was taking place, and Agar hid in the guard's van."

from nltk.tokenize import word\_tokenize

token= word\_tokenize(text)

print("\nWord Tokenised:")

print(token)

from nltk.corpus import stopwords

stop\_words=set(stopwords.words("english"))

print("\nStop Words")

print(stop\_words)

#text=re.sub('[a-zA-Z]',' ', text)

tokens=word\_tokenize(text.lower())

filtered\_text=[]

for w in tokens:

    if w not in stop\_words:

        filtered\_text.append(w)

print("\n")

print("Filtered Sentence:")

print(filtered\_text)

from nltk.stem import PorterStemmer

ps=PorterStemmer()

#filtered\_text1=["wait","waiting","waited","waits"]

for w in token:

    rootWords=ps.stem(w)

print("\n Stemming:")

print(rootWords)

from nltk.stem import WordNetLemmatizer

words=WordNetLemmatizer()

for w in token:

    print("Lemma for {} is {}".format(w,words.lemmatize(w)))

from nltk import pos\_tag

pos=pos\_tag(token)

print("POS Tagging")

print(pos)

def computeTF(wordDict, bagOfWords):

    tfDict = {}

    bagOfWordsCount = len(bagOfWords)

    for word, count in wordDict.items():

        tfDict[word] = count / float(bagOfWordsCount)

    return tfDict

def computeIDF(documents):

    import math

    N = len(documents)

    idfDict = dict.fromkeys(documents[0].keys(), 0)

    for document in documents:

        for word, val in document.items():

            if val > 0:

                idfDict[word] += 1

                for word, val in idfDict.items():

                    if val > 0: idfDict[word] = math.log(N / float(val))

                    else: idfDict[word] = 0

    return idfDict

def computeTFIDF(tfBagOfWords, idfs):

    tfidf = {}

    for word, val in tfBagOfWords.items():

        tfidf[word] = val \* idfs[word]

    return tfidf

# Algorithm for Create representation of document by calculating TFIDF

# Step 1: Import the necessary libraries.

from sklearn.feature\_extraction.text import TfidfVectorizer

# Step 2: Initialize the Documents.

documentA = 'Jupiter is the largest planet'

documentB = 'Mars is the fourth planet from the Sun'

# Step 3: Create BagofWords (BoW) for Document A and B. word tokenization

bagOfWordsA = documentA.split(' ')

bagOfWordsB = documentB.split(' ')

# Step 4: Create Collection of Unique words from Document A and B.

uniqueWords = set(bagOfWordsA).union(set(bagOfWordsB))

# Step 5: Create a dictionary of words and their occurrence for each document in the corpus

numOfWordsA = dict.fromkeys(uniqueWords, 0)

for word in bagOfWordsA:

    numOfWordsA[word] += 1 #How many times each word is repeated

numOfWordsB = dict.fromkeys(uniqueWords, 0)

for word in bagOfWordsB:

    numOfWordsB[word] += 1

# Step 6: Compute the term frequency for each of our documents.

tfA = computeTF(numOfWordsA, bagOfWordsA)

tfB = computeTF(numOfWordsB, bagOfWordsB)

# Step 7: Compute the term Inverse Document Frequency.

print('----------------Term Frequency----------------------')

df = pd.DataFrame([tfA, tfB])

print(df)

# Step 8: Compute the term TF/IDF for all words.

idfs = computeIDF([numOfWordsA, numOfWordsB])

print('----------------Inverse Document Frequency----------------------')

print(idfs)

tfidfA = computeTFIDF(tfA, idfs)

tfidfB = computeTFIDF(tfB, idfs)

print('------------------- TF-IDF--------------------------------------')

df = pd.DataFrame([tfidfA, tfidfB])

print(df)

**Output:**

[nltk\_data] Downloading package punkt to C:\Users\Himanshu

[nltk\_data]     Lonkar\AppData\Roaming\nltk\_data...

[nltk\_data]   Package punkt is already up-to-date!

[nltk\_data] Downloading package stopwords to C:\Users\Himanshu

[nltk\_data]     Lonkar\AppData\Roaming\nltk\_data...

[nltk\_data]   Package stopwords is already up-to-date!

[nltk\_data] Downloading package wordnet to C:\Users\Himanshu

[nltk\_data]     Lonkar\AppData\Roaming\nltk\_data...

[nltk\_data]   Package wordnet is already up-to-date!

[nltk\_data] Downloading package averaged\_perceptron\_tagger to

[nltk\_data]     C:\Users\Himanshu Lonkar\AppData\Roaming\nltk\_data...

[nltk\_data]   Package averaged\_perceptron\_tagger is already up-to-

[nltk\_data]       date!

Word Tokenised:

['The', 'Greatest', 'Gold', 'Robbery', 'took', 'place', 'on', 'the', 'night', 'of', '15', 'May', '1855', ',', 'when', 'a', 'shipment', 'of', 'gold', 'to', 'Paris', 'was', 'stolen', 'from', 'the', 'guard', "'s", 'van', 'of', 'the', 'rail', 'service', 'between', 'London', 'and', 'Folkestone', '.', 'There', 'were', 'four', 'robbers', ':', 'two', 'employees', 'of', 'the', 'rail', 'company', ',', 'a', 'former', 'employee', 'and', 'Edward', 'Agar', ',', 'a', 'career', 'criminal', '.', 'They', 'took', 'wax', 'impressions', 'of', 'the', 'keys', 'to', 'the', 'train', 'safes', 'and', 'made', 'copies', '.', 'One', 'of', 'them', 'ensured', 'he', 'was', 'on', 'guard', 'duty', 'when', 'a', 'shipment', 'was', 'taking', 'place', ',', 'and', 'Agar', 'hid', 'in', 'the', 'guard', "'s", 'van', '.']

Stop Words

{'mustn', 'a', 'd', 'not', 'her', 'she', 'here', 'can', 'than', 'shouldn', 'having', "wouldn't", 'doing', 'shan', 'with', 'most', 'needn', 'same', 'has', 'does', 'their', 'so', 'just', "you've", 'theirs', 'or', 'this', 'is', 'until', "needn't", 'of', 've', 's', 'for', 'through', 'will', 'any', "hasn't", 'myself', 'about', 'while', "shan't", "you're", 'because', 'my', 'your', 'i', 'our', "don't", 'weren', 'they', 'yourself', 'where', 'only', "weren't", 'to', 'all', 'these', 'haven', 'should', 'yourselves', 'who', 'itself', 'few', 'other', "it's", 'up', 'mightn', 'when', 'y', 'll', "you'll", 'won', 'over', 'hasn', 'ours', 'more', 'aren', 'whom', "haven't", 'be', "doesn't", 'ourselves', 'then', 'there', 'from', 'm', "mustn't", 'some', "wasn't", 'been', 'below', 'both', 'those', 'don', 'no', 'down', "didn't", 'yours', 'we', 'in', 'between', "mightn't", 'himself', "she's", 'ain', 't', 'me', 'above', 'it', 'own', 'nor', 'that', 'herself', 'after', 'am', 'ma', "should've", 'at', 'he', 'under', 'doesn', "aren't", 'wouldn', 'such', 'o', 'being', 'by', "won't", 'had', 'hers', 'too', 'but', 'are', 'during', 'if', 'couldn', 'didn', 'hadn', 'them', 'was', 'what', 'themselves', 'off', 'into', 'once', 'how', 'you', 'further', "couldn't", 'him', 'did', 'as', "isn't", 'were', 'out', "you'd", 'do', 'the', 'its', 'again', "shouldn't", 'which', 'now', 'against', 'isn', 'why', 'wasn', 'his', 'very', 'each', 'have', "that'll", 'before', 'on', 're', "hadn't", 'and', 'an'}

Filtered Sentence:

['greatest', 'gold', 'robbery', 'took', 'place', 'night', '15', 'may', '1855', ',', 'shipment', 'gold', 'paris', 'stolen', 'guard', "'s", 'van', 'rail', 'service', 'london', 'folkestone', '.', 'four', 'robbers', ':', 'two', 'employees', 'rail', 'company', ',', 'former', 'employee', 'edward', 'agar', ',', 'career', 'criminal', '.', 'took', 'wax', 'impressions', 'keys', 'train', 'safes', 'made', 'copies', '.', 'one', 'ensured', 'guard', 'duty', 'shipment', 'taking', 'place', ',', 'agar', 'hid', 'guard', "'s", 'van', '.']

 Stemming:

.

Lemma for The is The

Lemma for Greatest is Greatest

Lemma for Gold is Gold

Lemma for Robbery is Robbery

Lemma for took is took

Lemma for place is place

Lemma for on is on

Lemma for the is the

Lemma for night is night

Lemma for of is of

Lemma for 15 is 15

Lemma for May is May

Lemma for 1855 is 1855

Lemma for , is ,

Lemma for when is when

Lemma for a is a

Lemma for shipment is shipment

Lemma for of is of

Lemma for gold is gold

Lemma for to is to

Lemma for Paris is Paris

Lemma for was is wa

Lemma for stolen is stolen

Lemma for from is from

Lemma for the is the

Lemma for guard is guard

Lemma for 's is 's

Lemma for van is van

Lemma for of is of

Lemma for the is the

Lemma for rail is rail

Lemma for service is service

Lemma for between is between

Lemma for London is London

Lemma for and is and

Lemma for Folkestone is Folkestone

Lemma for . is .

Lemma for There is There

Lemma for were is were

Lemma for four is four

Lemma for robbers is robber

Lemma for : is :

Lemma for two is two

Lemma for employees is employee

Lemma for of is of

Lemma for the is the

Lemma for rail is rail

Lemma for company is company

Lemma for , is ,

Lemma for a is a

Lemma for former is former

Lemma for employee is employee

Lemma for and is and

Lemma for Edward is Edward

Lemma for Agar is Agar

Lemma for , is ,

Lemma for a is a

Lemma for career is career

Lemma for criminal is criminal

Lemma for . is .

Lemma for They is They

Lemma for took is took

Lemma for wax is wax

Lemma for impressions is impression

Lemma for of is of

Lemma for the is the

Lemma for keys is key

Lemma for to is to

Lemma for the is the

Lemma for train is train

Lemma for safes is safe

Lemma for and is and

Lemma for made is made

Lemma for copies is copy

Lemma for . is .

Lemma for One is One

Lemma for of is of

Lemma for them is them

Lemma for ensured is ensured

Lemma for he is he

Lemma for was is wa

Lemma for on is on

Lemma for guard is guard

Lemma for duty is duty

Lemma for when is when

Lemma for a is a

Lemma for shipment is shipment

Lemma for was is wa

Lemma for taking is taking

Lemma for place is place

Lemma for , is ,

Lemma for and is and

Lemma for Agar is Agar

Lemma for hid is hid

Lemma for in is in

Lemma for the is the

Lemma for guard is guard

Lemma for 's is 's

Lemma for van is van

Lemma for . is .

[('The', 'DT'), ('Greatest', 'NNP'), ('Gold', 'NNP'), ('Robbery', 'NNP'), ('took', 'VBD'), ('place', 'NN'), ('on', 'IN'), ('the', 'DT'), ('night', 'NN'), ('of', 'IN'), ('15', 'CD'), ('May', 'NNP'), ('1855', 'CD'), (',', ','), ('when', 'WRB'), ('a', 'DT'), ('shipment', 'NN'), ('of', 'IN'), ('gold', 'NN'), ('to', 'TO'), ('Paris', 'NNP'), ('was', 'VBD'), ('stolen', 'VBN'), ('from', 'IN'), ('the', 'DT'), ('guard', 'NN'), ("'s", 'POS'), ('van', 'NN'), ('of', 'IN'), ('the', 'DT'), ('rail', 'NN'), ('service', 'NN'), ('between', 'IN'), ('London', 'NNP'), ('and', 'CC'), ('Folkestone', 'NNP'), ('.', '.'), ('There', 'EX'), ('were', 'VBD'), ('four', 'CD'), ('robbers', 'NNS'), (':', ':'), ('two', 'CD'), ('employees', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('rail', 'NN'), ('company', 'NN'), (',', ','), ('a', 'DT'), ('former', 'JJ'), ('employee', 'NN'), ('and', 'CC'), ('Edward', 'NNP'), ('Agar', 'NNP'), (',', ','), ('a', 'DT'), ('career', 'NN'), ('criminal', 'NN'), ('.', '.'), ('They', 'PRP'), ('took', 'VBD'), ('wax', 'JJ'), ('impressions', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('keys', 'NNS'), ('to', 'TO'), ('the', 'DT'), ('train', 'NN'), ('safes', 'NN'), ('and', 'CC'), ('made', 'VBD'), ('copies', 'NNS'), ('.', '.'), ('One', 'CD'), ('of', 'IN'), ('them', 'PRP'), ('ensured', 'VBD'), ('he', 'PRP'), ('was', 'VBD'), ('on', 'IN'), ('guard', 'NN'), ('duty', 'NN'), ('when', 'WRB'), ('a', 'DT'), ('shipment', 'NN'), ('was', 'VBD'), ('taking', 'VBG'), ('place', 'NN'), (',', ','), ('and', 'CC'), ('Agar', 'NNP'), ('hid', 'NN'), ('in', 'IN'), ('the', 'DT'), ('guard', 'NN'), ("'s", 'POS'), ('van', 'NN'), ('.', '.')]

----------------Term Frequency----------------------

     Sun     is   from   Mars  Jupiter   the  largest  fourth  planet

0  0.000  0.200  0.000  0.000      0.2  0.20      0.2   0.000   0.200

1  0.125  0.125  0.125  0.125      0.0  0.25      0.0   0.125   0.125

----------------Inverse Document Frequency----------------------

{'Sun': 0.4462224708037222, 'is': 0, 'from': 0.5560285979740467, 'Mars': 1.1469641504314008, 'Jupiter': 0.030849451681496814, 'the': 0.09454924633113536, 'largest': 0.28762552843253275, 'fourth': 1.0596601011416096, 'planet': 0.324192216096107}

------------------- TF-IDF--------------------------------------

        Sun   is      from      Mars  ...       the   largest    fourth    planet

0  0.000000  0.0  0.000000  0.000000  ...  0.018910  0.057525  0.000000  0.064838

1  0.055778  0.0  0.069504  0.143371  ...  0.023637  0.000000  0.132458  0.040524

[2 rows x 9 columns]