Practical No 7 AIM;-

Text Analytics

1. Extract Sample document and apply following document preprocessing methods:

Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.

1. Create representation of document by calculating Term Frequency and Inverse Document

Frequency.

//Tokenization

import nltk nltk.download('punkt') nltk.download('wordnet')

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

from nltk import sent\_tokenize from nltk import word\_tokenize from nltk.corpus import stopwords

[nltk\_data] Downloading package punkt to C:\Users\SOFT [nltk\_data] LAB01\AppData\Roaming\nltk\_data... [nltk\_data] Package punkt is already up-to-date! [nltk\_data] Downloading package wordnet to C:\Users\SOFT [nltk\_data] LAB01\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\SOFT LAB01\AppData\Roaming\nltk\_data... [nltk\_data] Package averaged\_perceptron\_tagger is already up-to- [nltk\_data] date!

[nltk\_data] Downloading package stopwords to C:\Users\SOFT [nltk\_data] LAB01\AppData\Roaming\nltk\_data... [nltk\_data] Package stopwords is already up-to-date!

text='Real madrid is set to win the UCL for the season . Benzema might win Balon dor . Salah might be the runner up'

tokens\_sents = nltk.sent\_tokenize(text) print(tokens\_sents)

['Real madrid is set to win the UCL for the season .', 'Benzema might win Balon dor .', 'Salah might be the runner up']

tokens\_words = nltk.word\_tokenize(text) print(tokens\_words)

['Real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'UCL', 'for',

'the', 'season', '.', 'Benzema', 'might', 'win', 'Balon', 'dor', '.', 'Salah', 'might', 'be', 'the', 'runner', 'up']

from nltk.stem import PorterStemmer

from nltk.stem.snowball import SnowballStemmer from nltk.stem import LancasterStemmer

stem=[]

for i in tokens\_words: ps = PorterStemmer() stem\_word= ps.stem(i) stem.append(stem\_word)

print(stem)

['real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'ucl', 'for',

'the', 'season', '.', 'benzema', 'might', 'win', 'balon', 'dor', '.', 'salah', 'might', 'be', 'the', 'runner', 'up']

//Lemmatization

import nltk

from nltk.stem import WordNetLemmatizer lemmatizer = WordNetLemmatizer()

lemmatized\_output = ' '.join([lemmatizer.lemmatize(w) for w in stem]) print(lemmatized\_output)

real madrid is set to win the ucl for the season . benzema might win balon dor . salah might be the runner up

leme=[]

for i in stem: lemetized\_word=lemmatizer.lemmatize(i) leme.append(lemetized\_word)

print(leme)

['real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'ucl', 'for',

'the', 'season', '.', 'benzema', 'might', 'win', 'balon', 'dor', '.', 'salah', 'might', 'be', 'the', 'runner', 'up']

//Part of Speech Tagging

print("Parts of Speech: ",nltk.pos\_tag(leme))

Parts of Speech: [('real', 'JJ'), ('madrid', 'NN'), ('is', 'VBZ'),

('set', 'VBN'), ('to', 'TO'), ('win', 'VB'), ('the', 'DT'), ('ucl',

'NN'), ('for', 'IN'), ('the', 'DT'), ('season', 'NN'), ('.', '.'),

('benzema', 'NN'), ('might', 'MD'), ('win', 'VB'), ('balon', 'NN'),

('dor', 'NN'), ('.', '.'), ('salah', 'NN'), ('might', 'MD'), ('be',

'VB'), ('the', 'DT'), ('runner', 'NN'), ('up', 'RP')]

//Stop Word

sw\_nltk = stopwords.words('english') print(sw\_nltk)

['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all',

'am', 'an', 'and', 'any', 'are', 'aren', "aren't", 'as', 'at', 'be', 'because', 'been', 'before', 'being', 'below', 'between', 'both',

'but', 'by', 'can', 'couldn', "couldn't", 'd', 'did', 'didn',

"didn't", 'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't",

'down', 'during', 'each', 'few', 'for', 'from', 'further', 'had',

'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven', "haven't",

'having', 'he', "he'd", "he'll", 'her', 'here', 'hers', 'herself',

"he's", 'him', 'himself', 'his', 'how', 'i', "i'd", 'if', "i'll",

"i'm", 'in', 'into', 'is', 'isn', "isn't", 'it', "it'd", "it'll",

"it's", 'its', 'itself', "i've", 'just', 'll', 'm', 'ma', 'me',

'mightn', "mightn't", 'more', 'most', 'mustn', "mustn't", 'my',

'myself', 'needn', "needn't", 'no', 'nor', 'not', 'now', 'o', 'of',

'off', 'on', 'once', 'only', 'or', 'other', 'our', 'ours',

'ourselves', 'out', 'over', 'own', 're', 's', 'same', 'shan',

"shan't", 'she', "she'd", "she'll", "she's", 'should', 'shouldn',

"shouldn't", "should've", 'so', 'some', 'such', 't', 'than', 'that',

"that'll", 'the', 'their', 'theirs', 'them', 'themselves', 'then',

'there', 'these', 'they', "they'd", "they'll", "they're", "they've",

'this', 'those', 'through', 'to', 'too', 'under', 'until', 'up', 've',

'very', 'was', 'wasn', "wasn't", 'we', "we'd", "we'll", "we're",

'were', 'weren', "weren't", "we've", 'what', 'when', 'where', 'which',

'while', 'who', 'whom', 'why', 'will', 'with', 'won', "won't",

'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", 'your', "you're", 'yours', 'yourself', 'yourselves', "you've"]

words = [word for word in text.split() if word.lower() not in sw\_nltk] new\_text = " ".join(words)

print(new\_text)

Real madrid set win UCL season . Benzema might win Balon dor . Salah might runner