**Q1. Pre-processing & Token Analysis Select a sample text (5–6 movie reviews or a short paragraph). Perform:**

1. **Tokenization**

import nltk

import spacy

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

nlp=spacy.load('en\_core\_web\_sm')

text='''Everyone knows that paper is made from trees. But when one looks at trees, one cannot imagine that something so soft and fragile as the paper is made is so hard and strong. Plant materials such as wood are made of fibres known as cellulose. It is the primary ingredient in paper making. Raw wood is first converted into pulp consisting of a mixture of Cellulose, lignin, water and some chemicals. The pulp can be made mechanically through grinders or through chemical processes. Short fibres are produced by mechanical grinding. The paper produced in this way is weak and is used to make newspapers, magazines and phonebooks.'''

doc=nlp(text)

li=[]

for tokens in doc:

  li.append(tokens.text)

print(li)

#**stop word removel**

no\_stop\_word=[]

for words in doc:

  if not words.is\_stop:

    no\_stop\_word.append(words.text)

print(no\_stop\_word)

**#stemming vs lemmatization**

from nltk.stem import PorterStemmer

ak=PorterStemmer()

for words in doc[1:11]:

  print(words.text,"|",ak.stem(words.text))

for words in doc[1:11]:

  print(words.text,"|",words.lemma\_)

#pos tagging

for words in doc:

print(words.text,"|",words.pos\_)

|  |  |
| --- | --- |
| Input | output |
| '''Everyone knows that paper is made from trees. But when one looks at trees, one cannot imagine that something so soft and fragile as the paper is made is so hard and strong. Plant materials such as wood are made of fibres known as cellulose. It is the primary ingredient in paper making. Raw wood is first converted into pulp consisting of a mixture of Cellulose, lignin, water and some chemicals. The pulp can be made mechanically through grinders or through chemical processes. Short fibres are produced by mechanical grinding. The paper produced in this way is weak and is used to make newspapers, magazines and phonebooks.''' | ['Everyone', 'knows', 'that', 'paper', 'is', 'made', 'from', 'trees', '.', 'But', 'when', 'one', 'looks', 'at', 'trees', ',', 'one', 'can', 'not', 'imagine', 'that', 'something', 'so', 'soft', 'and', 'fragile', 'as', 'the', 'paper', 'is', 'made', 'is', 'so', 'hard', 'and', 'strong', '.', 'Plant', 'materials', 'such', 'as', 'wood', 'are', 'made', 'of', 'fibres', 'known', 'as', 'cellulose', '.', 'It', 'is', 'the', 'primary', 'ingredient', 'in', 'paper', 'making', '.', 'Raw', 'wood', 'is', 'first', 'converted', 'into', 'pulp', 'consisting', 'of', 'a', 'mixture', 'of', 'Cellulose', ',', 'lignin', ',', 'water', 'and', 'some', 'chemicals', '.', 'The', 'pulp', 'can', 'be', 'made', 'mechanically', 'through', 'grinders', 'or', 'through', 'chemical', 'processes', '.', 'Short', 'fibres', 'are', 'produced', 'by', 'mechanical', 'grinding', '.', 'The', 'paper', 'produced', 'in', 'this', 'way', 'is', 'weak', 'and', 'is', 'used', 'to', 'make', 'newspapers', ',', 'magazines', 'and', 'phonebooks', '.' |
| '''Everyone knows that paper is made from trees. But when one looks at trees, one cannot imagine that something so soft and fragile as the paper is made is so hard and strong. Plant materials such as wood are made of fibres known as cellulose. It is the primary ingredient in paper making. Raw wood is first converted into pulp consisting of a mixture of Cellulose, lignin, water and some chemicals. The pulp can be made mechanically through grinders or through chemical processes. Short fibres are produced by mechanical grinding. The paper produced in this way is weak and is used to make newspapers, magazines and phonebooks.''' | ['knows', 'paper', 'trees', '.', 'looks', 'trees', ',', 'imagine', 'soft', 'fragile', 'paper', 'hard', 'strong', '.', 'Plant', 'materials', 'wood', 'fibres', 'known', 'cellulose', '.', 'primary', 'ingredient', 'paper', 'making', '.', 'Raw', 'wood', 'converted', 'pulp', 'consisting', 'mixture', 'Cellulose', ',', 'lignin', ',', 'water', 'chemicals', '.', 'pulp', 'mechanically', 'grinders', 'chemical', 'processes', '.', 'Short', 'fibres', 'produced', 'mechanical', 'grinding', '.', 'paper', 'produced', 'way', 'weak', 'newspapers', ',', 'magazines', 'phonebooks', '.'] |
| knows  that  paper  is  made  from  trees | Stemming lemmatization   |  |  | | --- | --- | | know | know | | that | that | | paper | paper | | is | be | | made | make | | from | from | | tree | tree | |
| Everyone  Knows  That  Paper  Is  Made  From  Trees  .  But  when  one  looks  at  Trees  ,  One  Can  Not  Imagine  That  Something  So  soft | Everyone | PRON  knows | VERB  that | DET  paper | NOUN  is | AUX  made | VERB  from | ADP  trees | NOUN  . | PUNCT  But | CCONJ  when | SCONJ  one | NUM  looks | VERB  at | ADP  trees | NOUN  , | PUNCT  one | PRON  can | AUX  not | PART  imagine | VERB  that | SCONJ  something | PRON  so | ADV  soft | ADJ |

**Q2. Vectorization Comparison Using 20 text samples**: ●

1. **Create features using BoW, TF-IDF, and (optional) word embeddings**

df=pd.read\_csv('Emotion\_classify\_Data.csv')

samp=df['Comment'].sample(20)

print(samp)

3147 im still not sure why reilly feels the need to...

3094 i may pour out the half empty cup here i will ...

5527 im excited to get home and spend time with eve...

4961 i checked on you was a long time ago i can say...

5243 i keep having all of these wonderful feelings ...

4892 i have said this before being a mom has made m...

399 i do find myself feeling anxious seeing what e...

2352 i don t feel stressed

4374 i have kept quiet when someone did or said som...

333 i feel my life being threatened by illness i l...

2504 id be feeling shaky too if id spent a week con...

2147 i feel as a person and a politician i cannot a...

4627 i did on weekends was sleep and feel bitter ab...

1643 i was feeling a little like a cold was coming on

3000 i must admit no matter how early i start playi...

3341 i feel sarcastic more often than not

5148 i started feeling a bit alarmed but i was not ...

1764 i trust my kids however i feel helpless enough...

993 i wasnt feeling well at all so had to take a f...

2710 i feel stressed my intention is to remain in c...

Name: Comment, dtype: object

Tfidf

from sklearn.feature\_extraction.text import TfidfVectorizer

tf=TfidfVectorizer()

vect=tf.fit\_transform(sample.values)

Bow

from sklearn.feature\_extraction.text import CountVectorizer

tf1=CountVectorizer()

vect1=tf1.fit\_transform(df['preprosessed\_text'].values)

wordembedding

import spacy.cli

spacy.cli.download("en\_core\_web\_lg")

nlp = spacy.load("en\_core\_web\_lg")

do=samp.apply(lambda x:nlp(x))

import numpy as np

vectors = np.array([doc.vector for doc in do])

**b) Show matrix shapes (samples × features)**

print(vect.shape)

print(vect1.shape)

print(vectors.shape)

output

(20, 213)

(20, 213)

(20, 300)

**c)Discuss which captures semantics better and why?**

Embeddings (like GloVe, Word2Vec, or BERT) represent words in **dense vectors based on context**.

They **capture similarity**: e.g., "king" and "queen" are close in vector space.

TF-IDF and BoW only encode word **frequency**, not **meaning** or **relationships**.

**Q3. Text Classification: Logistic Regression vs Naive Bayes Using 30–50 labeled text samples:** ●

**a) Preprocess & vectorize (BoW or TF-IDF)**

df=pd.read\_csv('Emotion\_classify\_Data.csv')

def preprocess(text):

    # remove stop words and lemmatize the text

    doc = nlp(text)

    filtered\_tokens = []

    for token in doc:

        if token.is\_stop or token.is\_punct:

            continue

        filtered\_tokens.append(token.lemma\_)

    return " ".join(filtered\_tokens)

df['preprosessed\_text']=df['Comment'].apply(lambda x:preprocess(x))

from sklearn.feature\_extraction.text import TfidfVectorizer

tf=TfidfVectorizer()

vect1=tf.fit\_transform(df['preprosessed\_text'].values)

1. **Train Naive Bayes & Logistic Regression models**
2. **Evaluate with Accuracy, F1-score, and Confusion Matrix**

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(vect1,df['Emotion'],test\_size=0.2,random\_state=42)

naivebayes

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import classification\_report,confusion\_matrix,accuracy\_score

clf=MultinomialNB()

clf.fit(x\_train,y\_train)

y\_pred=clf.predict(x\_test)

print(classification\_report(y\_test,y\_pred))

print(confusion\_matrix(y\_test,y\_pred))

print(accuracy\_score(y\_test,y\_pred))

precision recall f1-score support

OUTPUT:

precision recall f1-score support

anger 0.89 0.93 0.91 392

fear 0.91 0.90 0.91 416

joy 0.92 0.88 0.90 380

accuracy 0.91 1188

macro avg 0.91 0.91 0.91 1188

weighted avg 0.91 0.91 0.91 1188

[[364 15 13]

[ 25 376 15]

[ 22 22 336]]

0.9057239057239057

LogesticRegression

from sklearn.linear\_model import LogisticRegression

cl=LogisticRegression()

cl.fit(x\_train,y\_train)

y\_pred=cl.predict(x\_test)

print(classification\_report(y\_test,y\_pred))

print(confusion\_matrix(y\_test,y\_pred))

print(accuracy\_score(y\_test,y\_pred))

OUTPUT:

precision recall f1-score support

anger 0.93 0.92 0.92 392

fear 0.95 0.91 0.93 416

joy 0.91 0.95 0.93 380

accuracy 0.93 1188

macro avg 0.93 0.93 0.93 1188

weighted avg 0.93 0.93 0.93 1188

[[362 10 20]

[ 20 378 18]

[ 9 9 362]]

0.9276094276094277

**Which model performed better and why?**

**Logistic Regression** outperformed **Naive Bayes** based on:

* **Higher accuracy (93% vs 91%)**
* **Higher F1-score (0.93 vs 0.91)**
* **Better confusion matrix (fewer false positives)**

**Why?** Logistic Regression tends to perform better with small-to-medium datasets when:

* Feature independence (assumed by Naive Bayes) doesn't hold perfectly
* TF-IDF is used, which creates sparse continuous values — Logistic Regression handles this well

**Q4. Emotional Trajectory in a Passage**

Take a 3–4 paragraph text (e.g., from Harry Potter): ●

paragraphs=**'''JULY 16, 1833. --This is a memorable anniversary for me; on it I complete my three hundred and twenty-third year!**

**The Wandering Jew?--certainly not. More than eighteen centuries have passed over his head. In comparison with him, I am a very young Immortal.**

**Am I, then, immortal? This is a question which I have asked myself, by day and night, for now three hundred and three years, and yet cannot answer it. I detected a gray hair amidst my brown locks this very day-- that surely signifies decay. Yet it may have remained concealed there for three hundred years--for some persons have become entirely white headed before twenty years of age.**

**I will tell my story, and my reader shall judge for me. I will tell my story, and so contrive to pass some few hours of a long eternity, become so wearisome to me. For ever! Can it be? to live for ever! I have heard of enchantments, in which the victims were plunged into a deep sleep, to wake, after a hundred years, as fresh as ever: I have heard of the Seven Sleepers--thus to be immortal would not be so burthensome: but, oh! the weight of never-ending time--the tedious passage of the still-succeeding hours! How happy was the fabled Nourjahad!----But to my task.**

**All the world has heard of Cornelius Agrippa. His memory is as immortal as his arts have made me. All the world has also heard of his scholar, who, unawares, raised the foul fiend during his master's absence, and was destroyed by him. The report, true or false, of this accident, was attended with many inconveniences to the renowned philosopher. All his scholars at once deserted him--his servants disappeared. He had no one near him to put coals on his ever-burning fires while he slept, or to attend to the changeful colours of his medicines while he studied. Experiment after experiment failed, because one pair of hands was insufficient to complete them: the dark spirits laughed at him for not being able to retain a single mortal in his service.**

**I was then very young--very poor--and very much. in love. I had been for about a year the pupil of Cornelius, though I was absent when this accident took place. On my return, my friends implored me not to return to the alchymist's abode. I trembled as I listened to the dire tale they told; I required no second warning; and when Cornelius came and offered me a purse of gold if I would remain under his roof, I felt as if Satan himself tempted me. My teeth chattered--my hair stood on end:--I ran off as fast as my trembling knees would permit.'''**

**a) Split into 5 segments**

def splita(para,segment):

  words=para.split()

  piece\_length=len(words)//segment

  pices=[]

  for i in range(segment):

    start=i\*piece\_length

    end=(i+1)\*piece\_length

    piece=' '.join(words[start:end])

    pices.append(piece)

  return pices

segments=splita(paragraphs,5)

**b) Compute sentiment for each using VADER or TextBlob**

import nltk

from nltk.sentiment import SentimentIntensityAnalyzer

import matplotlib.pyplot as plt

import pandas as pd

nltk.download('vader\_lexicon')

sia=SentimentIntensityAnalyzer()

data={'segment':[1,2,3,4,5],

      'text':segments}

df=pd.DataFrame(data)

df['sentiment\_score']=df['text'].apply(lambda x:sia.polarity\_scores(x)['compound'])

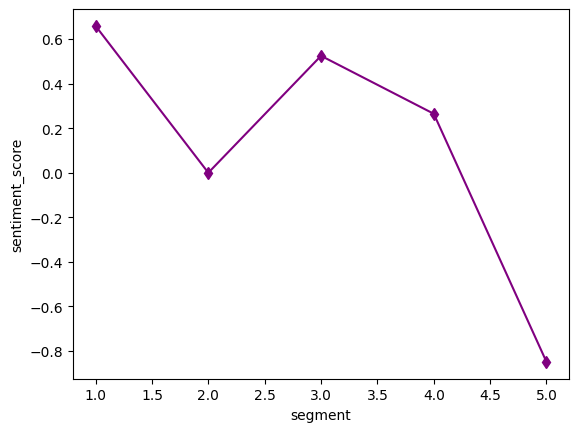
**c) Plot sentiment vs segment number**

plt.plot(df['segment'],df['sentiment\_score'],marker='d',color='purple')

plt.xlabel('segment')

plt.ylabel('sentiment\_score')

plt.show()



In 3–4 sentences, interpret the emotional journey.

**This paragraphs have emotional roller coster.It is the story of a immortel who shares his emotions**

**In initial segment the story beganed happily.in second segment it somewhat become nutrel .Then again he shares how immortal is different and what he seen in these years. Which is somewhat joyish in 3rd segment. But at last segment he explain his lonliness . and sentiment of sorrow increases drastically after 4th segment.**

**Q5. Conceptual Reflection (1–2 lines each)**

**1. Why is lemmatization often preferred over stemming?**  
Lemmatization returns actual dictionary words (lemmas), preserving meaning, while stemming may cut words crudely, sometimes resulting in non-words.

**2. How does TF-IDF down-weight common words?**  
TF-IDF reduces the importance of words that appear frequently across many documents by assigning them lower inverse document frequency (IDF) scores.

**3. Describe the curse of dimensionality in text data.**  
Text data often leads to high-dimensional sparse vectors (one dimension per word), making models slower, less accurate, and more prone to overfitting.

**4. When should you use word embeddings instead of BoW/TF-IDF?**  
Use word embeddings when semantic similarity and context are important, as they capture meaning and relationships between words in dense vectors.

**5. How can POS tagging enhance NLP pipelines?**  
POS tagging helps disambiguate word meaning and supports tasks like lemmatization, syntactic parsing, and named entity recognition.