**NAME: -------- -------**

**ROLL NO: 10---**

**CLASS: BSSE/7th/B**

**SUBJECT: Natural Language Processing**

**Q1:** Choose any corpus of your choice of at least 200 MBs of any domain in NLP and perform the following tasks:

* Text Preprocessing (Text Cleaning, Stemming / Lemmatization)
* Word Embedding (using an algorithm like Word2Vec, Glove, FastText)
* Encoding Techniques (Bag of Words, One – Hot)
* Parts of Speech tagging.

from google.colab import drive drive.mount('/content/drive')

Mounted at /content/drive

from google.colab import drive import os

# Mount your Google Drive drive.mount('/content/drive')

# Define the file path (make sure this is the correct path in your drive) file\_path = '/content/drive/MyDrive/carsnew\_data.csv'

# Check if the file exists if os.path.exists(file\_path):

 print("File found and loaded!") else: print("File not found. Please check the path.")

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_

File found and loaded!

import pandas as pd

file\_path = '/content/drive/MyDrive/carsnew\_data.csv' # Removed extra spaces at the beginning of this line df = pd.read\_csv(file\_path) # Removed extra spaces at the beginning of this line print(df.columns) # Removed extra spaces at the beginning of this line

Index(['abstract', 'author', 'date', 'pdf\_url', 'title', 'pdf\_text'], dtype='object')

import pandas as pd

# Load the dataset (assuming it's a CSV) file\_path = '/content/drive/MyDrive/carsnew\_data.csv' # Adjust file path

# Read the CSV file df = pd.read\_csv(file\_path)

# Check the columns and data types print(df.columns)

# Select the 'abstract' or 'pdf\_text' column for NLP tasks texts = df['abstract'] # or df['pdf\_text'] if that's more relevant print(texts.head()) # Preview the first few entries Index(['abstract', 'author', 'date', 'pdf\_url', 'title', 'pdf\_text'], dtype='object') 0 We first present our view of detection and cor...

1. We first present our view of detection and cor...
2. The choice of modeling units is critical to au...
3. Why should computers interpret language increm...
4. Stance detection is a classification problem i...

Name: abstract, dtype: object

!ls /content/drive/MyDrive/vehicles\_data.csv

/content/drive/MyDrive/vehicles\_data.csv

# Step 1: Text Preprocessing (Text Cleaning, Tokenization, Lemmatization)

import nltk import re from nltk.corpus import stopwords from nltk.tokenize import word\_tokenize from nltk.stem import WordNetLemmatizer

# Download necessary NLTK resources nltk.download('punkt') nltk.download('stopwords') nltk.download('wordnet')

# Preprocessing function def preprocess\_text(text):

# Remove non-alphabetic characters and convert to lowercase cleaned\_text = re.sub(r'[^a-zA-Z\s]', '', text.lower())

# Tokenize the cleaned text tokens = word\_tokenize(cleaned\_text)

# Remove stopwords stop\_words = set(stopwords.words('english')) filtered\_tokens = [word for word in tokens if word not in stop\_words]

# Lemmatization

lemmatizer = WordNetLemmatizer() lemmatized\_words = [lemmatizer.lemmatize(word) for word in filtered\_tokens] return lemmatized\_words

# Apply preprocessing to all abstracts

df['processed\_text'] = df['abstract'].apply(preprocess\_text) # or df['pdf\_text'] print(df['processed\_text'].head())

[nltk\_data] Downloading package punkt to /root/nltk\_data...

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

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

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

[nltk\_data] Downloading package wordnet to /root/nltk\_data...

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

1. [first, present, view, detection, correction, ...
2. [first, present, view, detection, correction, ...
3. [choice, modeling, unit, critical, automatic, ...
4. [computer, interpret, language, incrementally,...
5. [stance, detection, classification, problem, n...

Name: processed\_text, dtype: object

# Step 2: Word Embedding (Using Word2Vec)

from gensim.models import Word2Vec

# Prepare the data for Word2Vec (list of token lists) text\_list = df['processed\_text'].tolist()

# Create the Word2Vec model

word2vec\_model = Word2Vec(text\_list, vector\_size=100, window=5, min\_count=2, workers=4)

# Save the model word2vec\_model.save("word2vec.model")

# Example: Find similar words to 'data' print(word2vec\_model.wv.most\_similar('data'))

[('corpus', 0.5758063197135925), ('labeled', 0.5582237839698792), ('sample', 0.5454695224761963), ('indomain', 0.5

**Step 3:** Encoding Techniques (Bag of Words and One-Hot Encoding) **(i)Bag of Words (BoW):**

from sklearn.feature\_extraction.text import CountVectorizer

# Convert processed text back to strings for BoW

lemmatized\_texts = [' '.join(text) for text in df['processed\_text']]

# Create a Bag of Words representation vectorizer = CountVectorizer() X\_bow = vectorizer.fit\_transform(lemmatized\_texts)

# Display the BoW matrix and feature names print(X\_bow.toarray()) print(vectorizer.get\_feature\_names\_out()[:20]) # Show first 20 feature names

[[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ...

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]]

['aa' 'aaai' 'aac' 'aadit' 'aae' 'aaelike' 'aalstm' 'aalto' 'aam' 'aan'

'aapr' 'aardvark' 'aarnethompsonuther' 'aat' 'ab' 'abacha' 'abandon'

'abandoned' 'abandoning' 'abater']

**(ii)One-Hot Encoding:**

!pip install --upgrade scikit-learn from scipy.sparse import csr\_matrix import pandas as pd import numpy as np

from sklearn.feature\_extraction.text import CountVectorizer

# Assuming your data is in a CSV file named 'your\_data.csv' df = pd.read\_csv('/content/drive/MyDrive/carsnew\_data.csv') # Load your data into a DataFrame called 'df'

# Assuming 'abstract' column contains the text data

# If you have a different column with text data, replace 'abstract' with that column name corpus = df['abstract'].tolist() # Use the 'abstract' column directly

# Use CountVectorizer to create a sparse matrix of word counts

vectorizer = CountVectorizer(binary=True) # binary=True for one-hot encoding one\_hot\_encoded\_matrix = vectorizer.fit\_transform(corpus) print(one\_hot\_encoded\_matrix) # Display the one-hot encoded matrix

Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.5.2)

Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.26

.

Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.13.

1

Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.

2

Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn

)

(0, 31999)

1

(0, 11922)

1

(0, 22867)

1

(0, 21040)

1

(0, 31600)

1

(0, 20583)

1

(0, 8717)

1

(0, 2743)

1

(0, 7326)

1

(0, 28740)

1

(0, 10834)

1

(0, 29404)

1

(0, 15650)

1

(0, 19894)

1

(0, 18489)

1

(0, 3971)

1

(0, 20692)

1

(0, 13818)

1

(0, 7588)

1

(0, 31227)

1

(0, 29682)

1

(0, 8164)

1

(0, 32170)

1

(0, 26615)

1

(0, 4042)

1

:

:

(17217, 18024)

1

(17217, 19578)

1

(17217, 2691)

1

(17217, 32509)

1

(17217, 13584)

1

(17217, 9194)

1

(17217, 6705)

1

(17217, 7479)

1

(17217, 2517)

1

(17217, 3824)

1

(17217, 27769)

1

(17217, 6274)

1

(17217, 22168)

1

(17217, 21986)

1

(17217, 6509)

1

(17217, 29690)

1

(17217, 19479)

1

(17217, 23311)

1

(17217, 19478)

1

(17217, 5463)

1

(17217, 8006)

1

(17217, 2314)

1

(17217, 29271)

1

(17217, 24647)

1

(17217, 11961)

1

# Step 4: Parts of Speech (POS) Tagging

!pip install nltk import nltk nltk.download('averaged\_perceptron\_tagger') nltk.download('punkt') # Download the punkt tokenizer

# Function for POS tagging def pos\_tagging(text):

# Tokenize the text into words before POS tagging tokens = nltk.word\_tokenize(text) return nltk.pos\_tag(tokens)

# Assuming you want to apply POS tagging to the 'abstract' column df['pos\_tags'] = df['abstract'].apply(pos\_tagging)

# Display POS tags for the first row print(df['pos\_tags'].head())

Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)

Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)

Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.4.2)

Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2024.9.11)

Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.5)

[nltk\_data] Downloading package averaged\_perceptron\_tagger to [nltk\_data] /root/nltk\_data...

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

[nltk\_data] Downloading package punkt to /root/nltk\_data...

[nltk\_data] Unzipping tokenizers/punkt.zip.

1. [(We, PRP), (first, RB), (present, VBD), (our,...
2. [(We, PRP), (first, RB), (present, VBD), (our,...
3. [(The, DT), (choice, NN), (of, IN), (modeling,...
4. [(Why, WRB), (should, MD), (computers, NNS), (...
5. [(Stance, NNP), (detection, NN), (is, VBZ), (a...

Name: pos\_tags, dtype: object

**Q2:** Basic NLP Tasks For the second part, we’ll choose Named Entity Recognition (NER) and Sentiment Analysis.

**Task 1:** Named Entity Recognition (NER) We’ll use spaCy to extract named entities from each abstract or pdf\_text.

import spacy import nltk import pandas as pd

# Download spaCy's small English model if not already downloaded

!python -m spacy download en\_core\_web\_sm

# Load the model nlp = spacy.load("en\_core\_web\_sm", disable=["tagger", "parser"])

# Download necessary NLTK resources if not already downloaded nltk.download("averaged\_perceptron\_tagger") nltk.download("punkt")

# Function for POS tagging def pos\_tagging(text):

tokens = nltk.word\_tokenize(text) return nltk.pos\_tag(tokens)

# Function to perform Named Entity Recognition def ner(text):

doc = nlp(text) # Process the text directly return [(ent.text, ent.label\_) for ent in doc.ents]

# Sample the data (e.g., 10% of the original data) sample\_df = df.sample(frac=0.1, random\_state=42) # Adjust 'frac' for desired sample size

# Apply NER to the sample sample df["entities"] = sample df["abstract"].apply(ner) p \_ [ ] p \_ [ ] pp y( )

sample\_df["pos\_tags"] = sample\_df["abstract"].apply(pos\_tagging)

# Display the results for the sample

print(sample\_df[["pos\_tags", "entities"]].head())

Requirement already satisfied: typer<1.0.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,

Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0, Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.

Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (

Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.2->en-c

Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.2-> Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.

Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.

Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.

Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.10/dist-packages (from langcodes<4.0 Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic!

Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (from pydantic!=

Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydanti

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from request Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.

Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from thinc<8.3.0,>

Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from thinc<8

Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,>=0.3.0

Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,> Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,>=0.3.

Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from wease

Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from weasel< Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->spacy<3.

Requirement already satisfied: marisa-trie>=0.7.7 in /usr/local/lib/python3.10/dist-packages (from language-data

Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.1

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10

Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages (from smart-open<8.0.0,>=5.2.1-> Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0 ✔ Download and installation successful



You can now load the package via spacy.load('en\_core\_web\_sm')

⚠ Restart to reload dependencies

If you are in a Jupyter or Colab notebook, you may need to restart Python in order to load all the package's dependencies. You can do this by selecting the

'Restart kernel' or 'Restart runtime' option. [nltk\_data] Downloading package averaged\_perceptron\_tagger to [nltk\_data] /root/nltk\_data...

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

[nltk\_data] Downloading package punkt to /root/nltk\_data... [nltk\_data] Package punkt is already up-to-date!

/usr/local/lib/python3.10/dist-packages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based lem warnings.warn(Warnings.W108)

pos\_tags \ 7918 [(Knowledge, NNP), (distillation, NN), (descri...

12719 [(Though, IN), (languages, NNS), (can, MD), (e...

7502 [(The, DT), (Pointer-Generator, NNP), (archite...

980 [(The, DT), (availability, NN), (of, IN), (ope... 12367 [(Twitter, NN), (messages, NNS), ((, (), (twee...

entities 7918 [(Neural Machine Translation, ORG), (NMT, ORG)...

12719 [] 7502 [(over 30%, PERCENT), (two, CARDINAL), (ROUGE,...

980 [(Kaldi, PERSON), (Kaldi, PERSON), (PyTorch, P...

12367 [(NLP, ORG), (three, CARDINAL), (24 million, C...

import spacy import nltk import pandas as pd

# Download spaCy's small English model if not already downloaded

!python -m spacy download en\_core\_web\_sm

# Load the model, disabling unnecessary components nlp = spacy.load("en\_core\_web\_sm", disable=["tagger", "parser"])

# Download necessary NLTK resources if not already downloaded nltk.download("averaged\_perceptron\_tagger") nltk.download("punkt")

# Function for POS tagging def pos\_tagging(text):

tokens = nltk.word\_tokenize(text) return nltk.pos\_tag(tokens)

# Function to perform Named Entity Recognition def ner(text):

doc = nlp(text) # Process the text directly return [(ent.text, ent.label\_) for ent in doc.ents]

def process\_data(df): try:

# Sample the data (e.g., 10% of the original data)

sample\_df = df.sample(frac=0.1, random\_state=42) # Adjust 'frac' for desired sample size

# Apply NER and POS tagging to the sample sample\_df["entities"] = sample\_df["abstract"].apply(ner) sample\_df["pos\_tags"] = sample\_df["abstract"].apply(pos\_tagging)

# Display the results for the sample print(sample\_df[["pos\_tags", "entities"]].head()) except Exception as e:

print(f"Error encountered: {e}") print("Restarting the script...") return False # Indicate an error to trigger restart return True # Indicate successful processing

# Load the dataset (assuming it's a CSV file named 'your\_data.csv') df = pd.read\_csv('/content/drive/MyDrive/arxiv\_papers.csv') # Load your data into a DataFrame called 'df'

# Main execution loop with restart mechanism while True:

if process\_data(df): # Call your data processing function break # Exit the loop if processing was successful else:

print("Restarting...") # Optional message before restarting

# You can add any necessary cleanup or reset operations here before restarting

Requirement already satisfied: typer<1.0.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,

Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,

Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.

Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (

Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.2->en-c

Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.2->

Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,>

Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,>=0.3.

Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from wease

Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from weasel<

Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->spacy<3.

Requirement already satisfied: marisa-trie>=0.7.7 in /usr/local/lib/python3.10/dist-packages (from language-data

Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.1

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10

Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages (from smart-open<8.0.0,>=5.2.1->

Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0

✔

Download and installation successful

You can now load the package via spacy.load('en\_core\_web\_sm')

⚠

Restart to reload dependencies

If you are in a Jupyter or Colab notebook, you may need to restart Python in

order to load all the package's dependencies. You can do this by selecting the

'Restart kernel' or 'Restart runtime' option.

[

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

[

nltk\_data] /root/nltk\_data...

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

-

[

[

nltk\_data] date

!

[

nltk\_data] Downloading package punkt to /root/nltk\_data...

[

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

!

/usr/local/lib/python3.10/dist-packages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based lem

warnings.warn(Warnings.W108)

pos\_tags \

7918

[(Knowledge, NNP), (distillation, NN), (descri...

[(Though, IN), (languages, NNS), (can, MD), (e...

12719

[(The, DT), (Pointer-Generator, NNP), (archite...

7502

[(The, DT), (availability, NN), (of, IN), (ope...

980

[(Twitter, NN), (messages, NNS), ((, (), (twee...

12367

entities

[(Neural Machine Translation, ORG), (NMT, ORG)...

7918

[]

12719

7502

[(over 30%, PERCENT), (two, CARDINAL), (ROUGE,...

980

[(Kaldi, PERSON), (Kaldi, PERSON), (PyTorch, P...

12367

[(NLP, ORG), (three, CARDINAL), (24 million, C...

**Task 2: Sentiment Analysis (Using VADER and TextBlob)**

!

pip install vaderSentiment

!

pip install textblob

!

pip install spacy

!

python -m textblob.download\_corpora

!

python -m spacy download en\_core\_web\_sm

Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0, Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.

Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (

Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.2->en-c

Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.2-> Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.

Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.

Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.8.0,>=3.7.

Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.10/dist-packages (from langcodes<4.0 Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic!

Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (from pydantic!=

Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydanti

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from request Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.

Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from thinc<8.3.0,>

Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from thinc<8

Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,>=0.3.0

Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,> Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0.0,>=0.3.

Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from wease

Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from weasel< Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->spacy<3.

Requirement already satisfied: marisa-trie>=0.7.7 in /usr/local/lib/python3.10/dist-packages (from language-data

Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.1

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10

Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages (from smart-open<8.0.0,>=5.2.1-> Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0 ✔ Download and installation successful



You can now load the package via spacy.load('en\_core\_web\_sm')

⚠ Restart to reload dependencies

If you are in a Jupyter or Colab notebook, you may need to restart Python in order to load all the package's dependencies. You can do this by selecting the 'Restart kernel' or 'Restart runtime' option.

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# (i)VADER Sentiment Analysis

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

def vader\_sentiment(text):

analyzer = SentimentIntensityAnalyzer() return analyzer.polarity\_scores(text)

text\_sample = "The app is amazing but could improve in a few areas." vader\_result = vader\_sentiment(text\_sample) print("VADER Sentiment Result:", vader\_result)

VADER Sentiment Result: {'neg': 0.0, 'neu': 0.59, 'pos': 0.41, 'compound': 0.7391}

**(i)TextBlob Sentiment Analysis.**

from textblob import TextBlob

def textblob\_sentiment(text):

blob = TextBlob(text) return blob.sentiment

textblob\_result = textblob\_sentiment(text\_sample) print("TextBlob Sentiment Result:", textblob\_result) TextBlob Sentiment Result: Sentiment(polarity=0.20000000000000004, subjectivity=0.5)

!pip install import-ipynb import import\_ipynb from textblob import TextBlob

from nltk.sentiment.vader import SentimentIntensityAnalyzer import nltk

nltk.download('vader\_lexicon') nltk.download('punkt')

nltk.download('averaged\_perceptron\_tagger') nltk.download('maxent\_ne\_chunker') nltk.download('words')

# ... (Your other imports and function definitions) ...

def vader\_sentiment(text):

analyzer = SentimentIntensityAnalyzer() scores = analyzer.polarity\_scores(text) return scores

def textblob\_sentiment(text):

blob = TextBlob(text) return blob.sentiment

def perform\_ner(text):

# ... (Your NER logic) ... # Placeholder for demonstration ner\_results = [] # In a real scenario, replace with actual NER output return ner\_results

# Instead of importing from specific cell names, define the functions directly in this cell

# or within an external Python file that you import.

# Example: If your vader\_sentiment and textblob\_sentiment were in 'my\_functions.py',

# you would import them like this:

# from my\_functions import vader\_sentiment, textblob\_sentiment

# Your main code block # Load the corpus file

corpus\_path = '/content/drive/My Drive/vehicles\_data.csv' with open(corpus\_path, 'r') as file: corpus\_text = file.read()

# Perform sentiment analysis and NER on the corpus text vader\_result\_corpus = vader\_sentiment(corpus\_text[:1000]) textblob\_result\_corpus = textblob\_sentiment(corpus\_text[:1000]) ner\_result\_corpus = perform\_ner(corpus\_text[:1000])

# Print the results print("VADER Corpus Sentiment:", vader\_result\_corpus) print("TextBlob Corpus Sentiment:", textblob\_result\_corpus) print("NER Corpus Result:", ner\_result\_corpus)

Requirement already satisfied: import-ipynb in /usr/local/lib/python3.10/dist-packages (0.2)

Requirement already satisfied: IPython in /usr/local/lib/python3.10/dist-packages (from import-ipynb) (7.34.0)

Requirement already satisfied: nbformat in /usr/local/lib/python3.10/dist-packages (from import-ipynb) (5.10.4)

Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages (from IPython->import-i

Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-packages (from IPython->import-ipynb)

Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from IPython->import-ipynb) (

Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (from IPython->import-ipynb)

Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/dist-packages (from IPython->import-ipy

Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-pack Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from IPython->import-ipynb) (2

Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from IPython->import-ipynb) (0

Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages (from IPython->import-

Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (from IPython->import-ipynb)

Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.10/dist-packages (from nbformat->imp

Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.10/dist-packages (from nbformat->import-i

Requirement already satisfied: jupyter-core!=5.0.\*,>=4.12 in /usr/local/lib/python3.10/dist-packages (from nbforma

Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from jedi>=0.16->IP

Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbf

Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (fr Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.

Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nb Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from jupyter-core!=5.

Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages (from pexpect>4.3->IPyth Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0,!=3.

[nltk\_data] Downloading package vader\_lexicon to /root/nltk\_data...

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

[nltk\_data] Downloading package punkt to /root/nltk\_data...

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

[nltk\_data] Downloading package averaged\_perceptron\_tagger to [nltk\_data] /root/nltk\_data...

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

[nltk\_data] Downloading package maxent\_ne\_chunker to [nltk\_data] /root/nltk\_data...

[nltk\_data] Package maxent\_ne\_chunker is already up-to-date!

[nltk\_data] Downloading package words to /root/nltk\_data...

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

VADER Corpus Sentiment: {'neg': 0.035, 'neu': 0.893, 'pos': 0.072, 'compound': 0.6204}

TextBlob Corpus Sentiment: Sentiment(polarity=0.14829545454545454, subjectivity=0.34848484848484845)

NER Corpus Result: []

Double-click (or enter) to edit