ABBOTTABAD UNIVERSITY OF SCIENCE & TECHNOLOGY

NAME: NEHA IRSHAD

ROLL NO: 10186

CLASS: BSSE 7(A)

SUBJECT: Natural Language Processing

SUBMITTED BY:SIR JAMAL ABDUL AHAD

DATE:31-OCT 024

ASSIGNMENT:1

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/vehicles_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/vehicles_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/vehicles_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!
     0    [first, present, view, detection, correction, ...
     1    [first, present, view, detection, correction, ...
     2    [choice, modeling, unit, critical, automatic, ...
     3    [computer, interpret, language, incrementally,...
     4    [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/vehicles_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.
0    [(We, PRP), (first, RB), (present, VBD), (our,...
1    [(We, PRP), (first, RB), (present, VBD), (our,...
2    [(The, DT), (choice, NN), (of, IN), (modeling,...
3    [(Why, WRB), (should, MD), (computers, NNS), (...
4    [(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...
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...
```

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