```
Start coding or generate with AI.
```

NAME : NITHIN.V

ROLL.NO : 2403A52355

LAB : 05

1.Install required libraries & load spaCy English model

```
!pip install pandas spacy matplotlib emoji
!python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: emoji in /usr/local/lib/python3.12/dist-packages (2.15.0)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.3)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.21.1)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.61.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.3.1)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spac
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spacy
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0
Collecting en-core-web-sm==3.8.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0-py3-none
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 112.3 MB/s eta 0:00:00
✔ 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.
```

2.Load the Twitter US Airline Sentiment dataset

```
import pandas as pd

df = pd.read_csv("Tweets.csv")
```

3.Select tweet text & sentiment columns and remove missing values

```
df = df[["text", "airline_sentiment"]]
df.dropna(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 2 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   text               14640 non-null  object
 1   airline_sentiment  14640 non-null  object
dtypes: object(2)
memory usage: 228.9+ KB
```

4.Clean tweets by removing URLs, mentions, emojis, special characters, and converting text to lowercase.

```
import re
import emoji

def clean_tweet(text):
    text = re.sub(r"http\S+|www\S+", "", text)   # Remove URLs
    text = re.sub(r"@\w+", "", text)             # Remove mentions
    text = emoji.replace_emoji(text, replace="")# Remove emojis
    text = re.sub(r"[^a-zA-Z# ]+", "", text)     # Remove special chars
    text = text.lower().strip()
    return text
```

5.Create a cleaned tweet corpus after preprocessing.

```
df["cleaned_text"] = df["text"].apply(clean_tweet)
display(df.head())
```

	text	airline_sentiment	cleaned_text
0	@VirginAmerica What @dhepburn said.	neutral	what said
1	@VirginAmerica plus you've added commercials t...	positive	plus youve added commercials to the experience...
2	@VirginAmerica I didn't today... Must mean I n...	neutral	i didnt today must mean i need to take another...
3	@VirginAmerica it's really aggressive to blast...	negative	its really aggressive to blast obnoxious enter...
4	@VirginAmerica and it's a really big bad thing...	negative	and its a really big bad thing about it

6.Initialize the spaCy NLP pipeline.

```
import spacy

nlp = spacy.load("en_core_web_sm")
```

7.Create and add a custom spaCy pipeline component to detect hashtags.

```
from spacy.language import Language
from spacy.tokens import Doc

Doc.set_extension("hashtags", default=[], force=True)

@Language.component("hashtag_detector")
def hashtag_detector(doc):
    doc._.hashtags = [token.text for token in doc if token.text.startswith("#")]
    return doc

if "hashtag_detector" not in nlp.pipe_names:
    nlp.add_pipe("hashtag_detector", last=True)
nlp.pipe_names
```

```
['tok2vec',
 'tagger',
 'parser',
 'attribute_ruler',
 'lemmatizer',
 'ner',
 'hashtag_detector']
```

8.Process the cleaned tweets using the customized spaCy pipeline.

```
docs = list(nlp.pipe(cleaned_corpus))
```

9.Extract lemmas and part-of-speech tags from processed tweets.

```
lemmatized_pos = []

for doc in docs:
    tokens = [(token.lemma_, token.pos_)
              for token in doc
              if not token.is_stop and not token.is_punct]
    lemmatized_pos.append(tokens)

lemmatized_pos[:2]
```

```
[[(' ', 'SPACE'), ('say', 'VERB')],
 [('plus', 'CCONJ'),
  ('ve', 'AUX'),
  ('add', 'VERB'),
  ('commercial', 'NOUN'),
  ('experience', 'NOUN'),
  ('tacky', 'ADV')]]
```

10.Extract hashtags from original tweets and compute their frequencies.

```
from collections import Counter

hashtags = []
for text in df["text"]:
    hashtags.extend(re.findall(r"#\w+", text.lower()))

hashtag_freq = Counter(hashtags)
hashtag_freq.most_common(10)
```

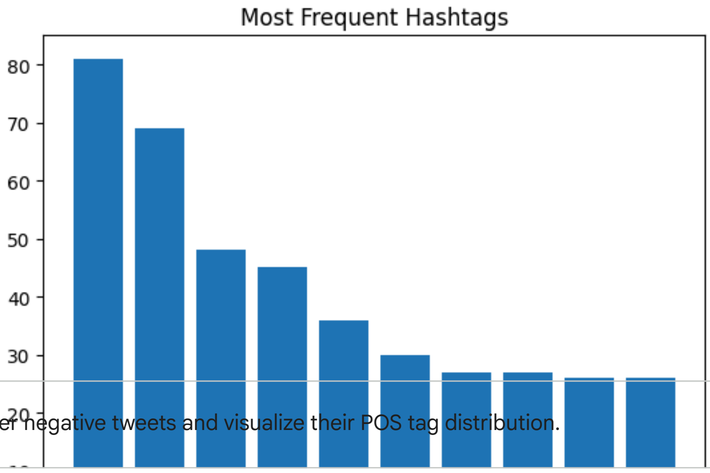
```
[('#destinationdragons', 81),
 ('#fail', 69),
 ('#jetblue', 48),
 ('#unitedairlines', 45),
 ('#customerservice', 36),
 ('#usairways', 30),
 ('#americanairlines', 27),
 ('#neveragain', 27),
 ('#united', 26),
 ('#usairwaysfail', 26)]
```

11.Visualize the most frequent hashtags.

```
import matplotlib.pyplot as plt

top_hashtags = hashtag_freq.most_common(10)
labels, values = zip(*top_hashtags)

plt.figure()
plt.bar(labels, values)
plt.title("Most Frequent Hashtags")
plt.xticks(rotation=45)
plt.show()
```



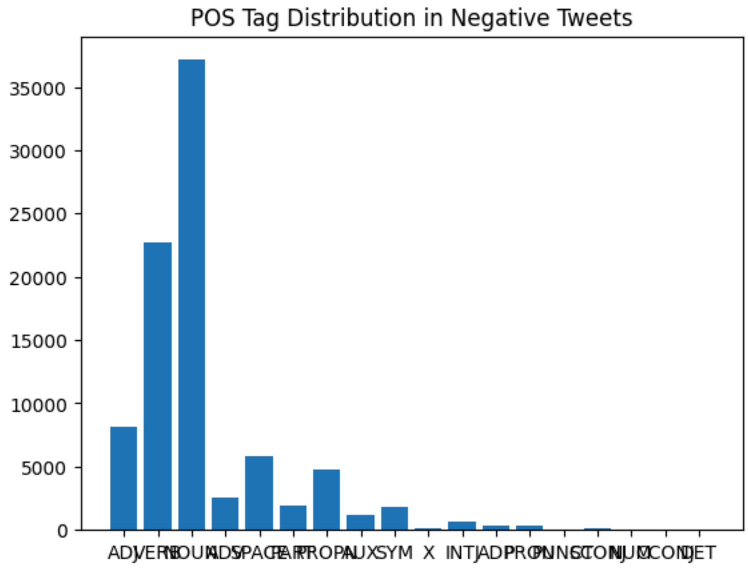
12.Filter negative tweets and visualize their POS tag distribution.

```
negative_df = df[df["airline_sentiment"] == "negative"]

neg_pos_tags = [
    pos for tags in negative_df["pos_tags"] for pos in tags
]

pos_freq = Counter(neg_pos_tags)

plt.figure()
plt.bar(pos_freq.keys(), pos_freq.values())
plt.title("POS Tag Distribution in Negative Tweets")
plt.show()
```


POS Tag Distribution in Negative Tweets
ADJ
VERB
NOUN
ADV
SPACE
PART
PROPN
AUX
SYM
X
INTJ
ADP
PRON
PUNCT
SCONJ
NUM
CCONJ
DET