Name: Akhil.B

Hallticket:2403a52344

lab:05

step 1: Install required libraries and load the spaCy English model

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

```
Collecting emoji
  Downloading emoji-2.15.0-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: wordcloud in /usr/local/lib/python3.12/dist-packages (1.9.5)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post
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
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spac
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->sp
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
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spa
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=
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.
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
Downloading emoji-2.15.0-py3-none-any.whl (608 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 608.4/608.4 kB 10.5 MB/s eta 0:00:00
Installing collected packages: emoji
Successfully installed emoji-2.15.0
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-no
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 69.0 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.
```

step 2: Load the Twitter US Airline Sentiment dataset

```
import pandas as pd
df = pd.read_csv("Tweets.csv")

df.head()
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America

Next steps: Generate code with df New interactive sheet

step 3:Select tweet text and sentiment columns and remove missing values

```
df = df[['text', 'airline_sentiment']]
df = df.dropna()

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
```

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

```
import re
import emoji

def clean_tweet(text):
    text = text.lower()
    text = re.sub(r"http\S+|www\S+", "", text)       # URLs
    text = re.sub(r"@\w+", "", text)                  # Mentions
    text = re.sub(r"#\w+", "", text)                  # Hashtags (text only)
    text = emoji.replace_emoji(text, replace="")     # Emojis
    text = re.sub(r"[^a-z\s]", "", text)              # Special characters
    text = re.sub(r"\s+", " ", text).strip()
    return text
```

step 5: Create a cleaned tweet corpus after preprocessing

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

	text	airline_sentiment	clean_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

Next steps: Generate code with df New interactive sheet

step 6:Initialize the spaCy NLP pipeline.

```
import spacy

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

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

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

# Define custom extension
Doc.set_extension("hashtags", default=[])

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

# Add component to pipeline
nlp.add_pipe("hashtag_detector", last=True)

nlp.pipe_names
```

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

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

```
df['cleaned_text'] = df['text'].apply(clean_tweet)
docs = list(nlp.pipe(df["cleaned_text"]))
```

step 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]
```

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

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

```
from collections import Counter

def extract_hashtags(text):
    return re.findall(r"#\w+", text.lower())

all_hashtags = []

for tweet in df["text"]:
    all_hashtags.extend(extract_hashtags(tweet))

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

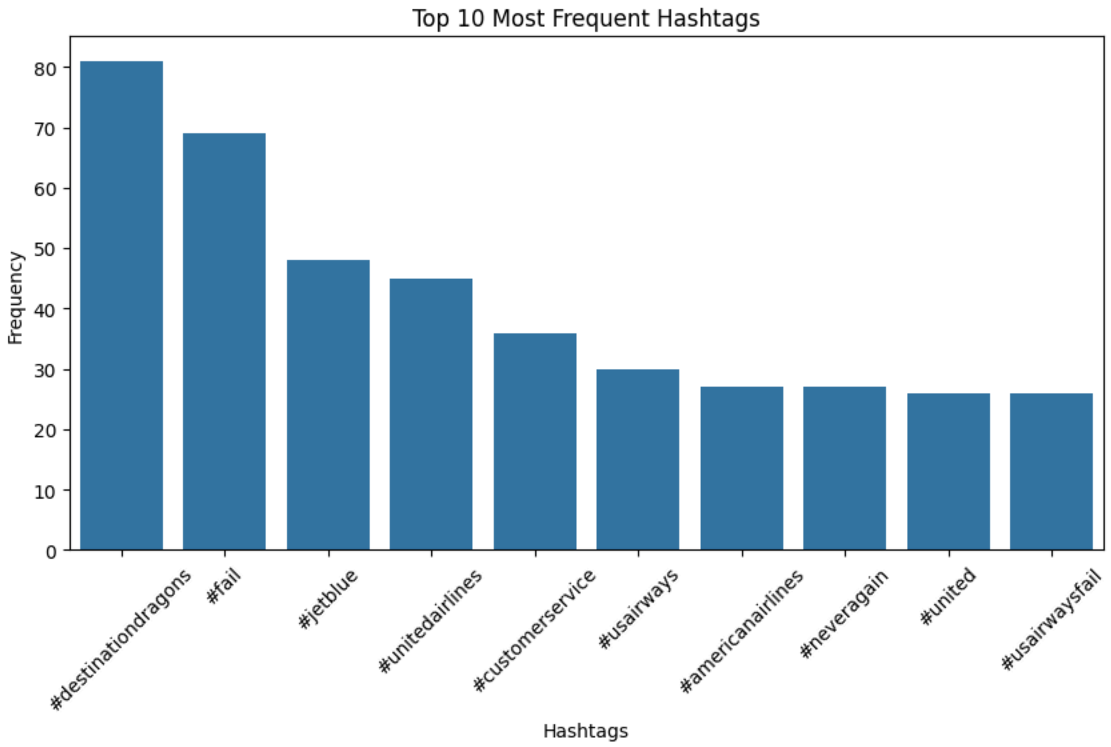
```
[('#destinationdragons', 81),
 ('#fail', 69),
 ('#jetblue', 48),
 ('#unitedairlines', 45),
 ('#customerservice', 36),
 ('#usairways', 30),
 ('#americanairlines', 27),
 ('#neveragain', 27),
 ('#united', 26),
 ('#usairwaysfail', 26)]
```

step 11:Visualize the most frequent hashtags

```
import matplotlib.pyplot as plt
import seaborn as sns

top_hashtags = hashtag_freq.most_common(10)
tags, counts = zip(*top_hashtags)

plt.figure(figsize=(10,5))
sns.barplot(x=list(tags), y=list(counts))
plt.title("Top 10 Most Frequent Hashtags")
plt.xlabel("Hashtags")
plt.ylabel("Frequency")
plt.xticks(rotation=45)
plt.show()
```



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

```
# Filter negative tweets
negative_df = df[df["airline_sentiment"] == "negative"]

negative_docs = list(nlp.pipe(negative_df["clean_text"], disable=["ner"]))
```

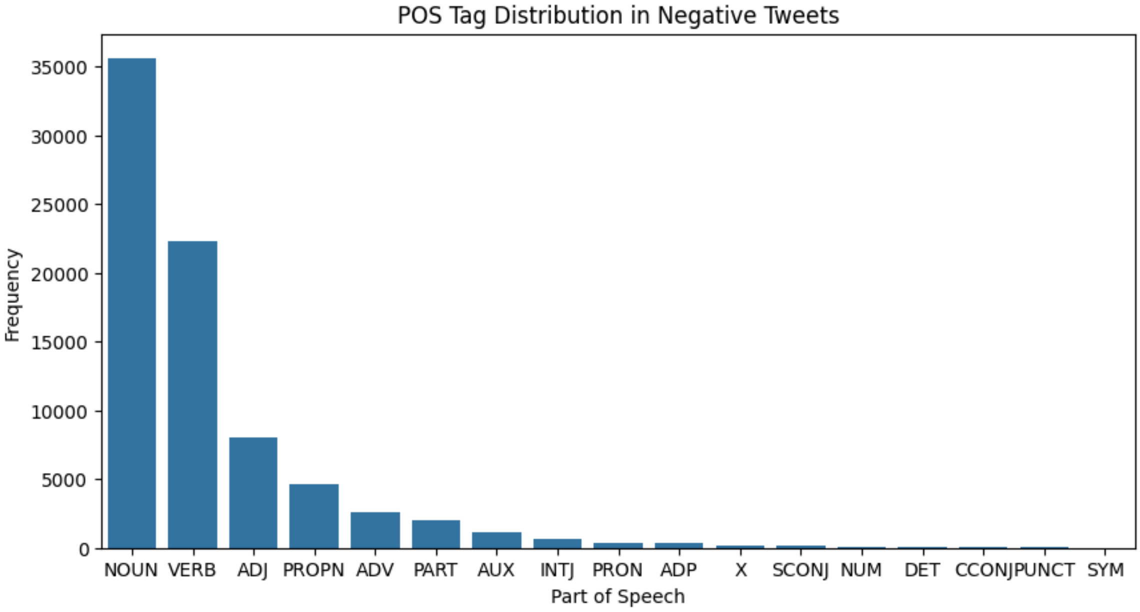
```
# Collect POS tags
pos_tags = []

for doc in negative_docs:
    pos_tags.extend([token.pos_ for token in doc
                     if not token.is_stop and not token.is_punct])

pos_freq = Counter(pos_tags)

# Convert to DataFrame
pos_df = pd.DataFrame(pos_freq.items(), columns=["POS", "Frequency"])
pos_df = pos_df.sort_values(by="Frequency", ascending=False)

# Plot
plt.figure(figsize=(10,5))
sns.barplot(data=pos_df, x="POS", y="Frequency")
plt.title("POS Tag Distribution in Negative Tweets")
plt.xlabel("Part of Speech")
plt.ylabel("Frequency")
plt.show()
```



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