

1. Install required libraries and load the spaCy English model

◆ Gemini

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

```
  Downloading emoji-2.15.0-py3-none-any.whl.metadata (5./ KB)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: pydantic!=1.8.,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from -r /tmp/ipykernel_1030/1030/requirements.txt)
```

```
ecting en-core-web-sm==3.8.0
wnloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0.tar.gz
12.8/12.8 MB 83.0 MB/s eta 0:00:00
ownload and installation successful
can now load the package via spacy.load('en_core_web_sm')
estart to reload dependencies
ou are in a Jupyter or Colab notebook, you may need to restart Python in
r to load all the package's dependencies. You can do this by selecting the
'tart kernel' or 'Restart runtime' option.
```

```
import re
import pandas as pd
import spacy
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import emoji
```

2. Load the Twitter US Airline Sentiment dataset

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

tweet_id	airline_sentiment	airline_sentiment_confidence	negative
----------	-------------------	------------------------------	----------

570306133677760513	neutral	1.0000
--------------------	---------	--------

570301130888122368	positive	0.3486
--------------------	----------	--------

570301083672813571	neutral	0.6837
--------------------	---------	--------

570301031407624196	negative	1.0000
--------------------	----------	--------

570300817074462722	negative	1.0000
--------------------	----------	--------

Next steps:

[Generate code with df](#)

[New interactive sheet](#)

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

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

df.shape
```

(14640, 2)

4. Clean tweets

```
def clean_tweet(text):
    text = text.lower()
    text = re.sub(r"http\S+|www\S+", "", text)      # remove URLs
    text = re.sub(r"@[\w+]", "", text)                # remove mentions
    text = re.sub(r"#", "", text)                     # remove hashtag symbol
    text = emoji.replace_emoji(text, replace="")
    text = re.sub(r"[^a-z\s]", "", text)              # remove emojis
    text = re.sub(r"\s+", " ", text).strip()           # remove special characters
    return text
```

```
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 you've 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](#)

5. Create a cleaned tweet corpus

```
corpus = df['clean_text'].tolist()
corpus[:5]
```

```
[ 'what said',
  'plus youve added commercials to the experience tacky',
  'i didnt today must mean i need to take another trip',
  'its really aggressive to blast obnoxious entertainment in your guests
faces amp they have little recourse',
  'and its a really big bad thing about it']
```

6. Initialize the spaCy NLP pipeline

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

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

```
from spacy.language import Language

@Language.component("hashtag_detector")
def hashtag_detector(doc):
    hashtags = re.findall(r"#\w+", doc.text)
    doc._.hashtags = hashtags
    return doc
```

```
from spacy.tokens import Doc

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

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(corpus))
```

9. Extract lemmas and part-of-speech tags

```

lemmatized_pos = []

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

lemmatized_pos[:2]

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

```

10. Extract hashtags from original tweets and compute frequencies

```

hashtag_counter = Counter()

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

hashtag_counter.most_common(10)

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

```

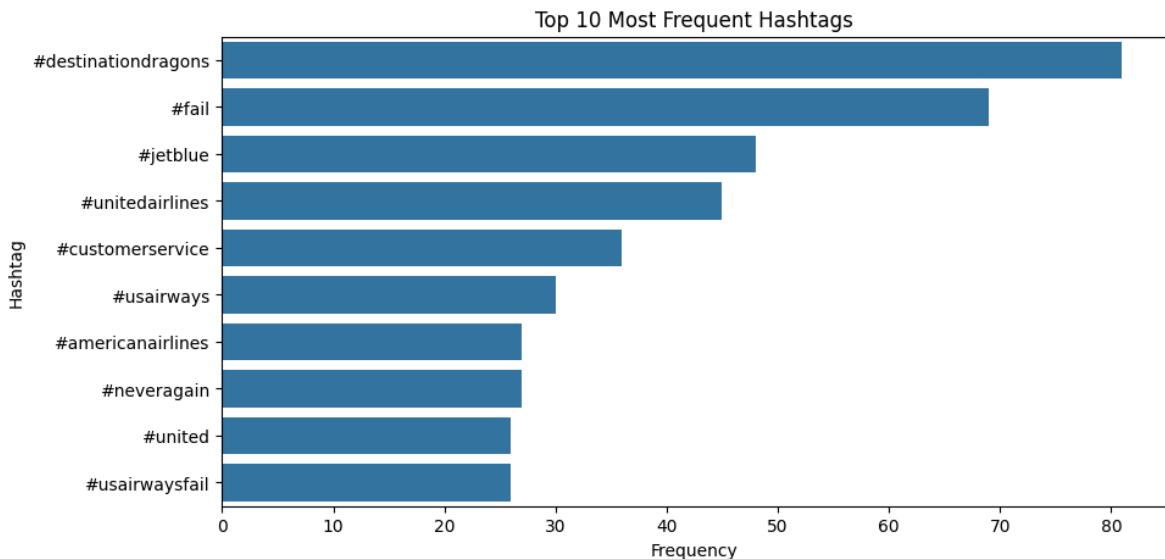
11. Visualize the most frequent hashtags

```

top_hashtags = hashtag_counter.most_common(10)
hashtags, counts = zip(*top_hashtags)

plt.figure(figsize=(10,5))
sns.barplot(x=list(counts), y=list(hashtags))
plt.title("Top 10 Most Frequent Hashtags")
plt.xlabel("Frequency")
plt.ylabel("Hashtag")
plt.show()

```



12. Filter negative tweets and visualize their POS tag distribution

```
negative_df = df[df['airline_sentiment'] == 'negative']
negative_docs = list(nlp.pipe(negative_df['clean_text']))
```

```
pos_counter = Counter()

for doc in negative_docs:
    for token in doc:
        if token.is_alpha and not token.is_stop:
            pos_counter[token.pos_] += 1

pos_counter
```

```
Counter({'ADJ': 8272,
         'VERB': 22633,
         'NOUN': 36596,
         'ADV': 2633,
         'PART': 1974,
         'PROPN': 4918,
         'AUX': 1185,
         'ADP': 344,
         'INTJ': 676,
         'PRON': 347,
         'X': 165,
         'PUNCT': 22,
         'SCONJ': 125,
         'NUM': 58,
         'CCONJ': 53,
         'DET': 60,
         'SYM': 2})
```

```
plt.figure(figsize=(8,5))
sns.barplot(
    x=list(pos_counter.values()),
    y=list(pos_counter.keys())
)
```

```
plt.title("POS Tag Distribution in Negative Tweets")
plt.xlabel("Count")
plt.ylabel("POS Tag")
plt.show()
```

