2403A52031

AKULA MANOJ

B-02

Step 1: Install & Import Libraries

```
!pip install nltk spacy wordcloud

Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-
packages (3.9.1)
Requirement already satisfied: spacy in
/usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: wordcloud in
/usr/local/lib/python3.12/dist-packages (1.9.5)
Requirement already satisfied: click in
/usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in
/usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-
packages (from nltk) (4.67.1)
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: numpy>=1.19.0 in
/usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
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) (2.12.3)
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: pillow in
/usr/local/lib/python3.12/dist-packages (from wordcloud) (11.3.0)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.12/dist-packages (from wordcloud) (3.10.0)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!
=1.8.1,<3.0.0,>=1.7.4->spacy) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in
/usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!
=1.8.1,<3.0.0,>=1.7.4->spacy) (2.41.4)
Requirement already satisfied: typing-extensions>=4.14.1 in
/usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!
=1.8.1,<3.0.0,>=1.7.4->spacy) (4.15.0)
Requirement already satisfied: typing-inspection>=0.4.2 in
/usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!
=1.8.1,<3.0.0,>=1.7.4->spacy) (0.4.2)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
>spacy) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
>spacy) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
>spacy) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
>spacy) (2026.1.4)
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-
>spacy) (1.3.3)
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-
>spacy) (0.1.5)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in
/usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2-
>spacy) (0.23.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in
/usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2-
>spacy) (7.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in
```

```
/usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib->wordcloud)
(1.3.3)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.12/dist-packages (from matplotlib->wordcloud)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.12/dist-packages (from matplotlib->wordcloud)
(4.61.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib->wordcloud)
(1.4.9)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib->wordcloud)
(3.3.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.12/dist-packages (from matplotlib->wordcloud)
(2.9.0.post0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7-
>matplotlib->wordcloud) (1.17.0)
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,>=0.4.2->spacy) (2.0.1)
```

```python
import pandas as pd
import re
import nltk
import matplotlib.pyplot as plt

from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from wordcloud import WordCloud

nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
True
```

Step 2: Load Dataset

```python
df = pd.read_csv("/content/Tweets.csv")
df = df[['text', 'airline_sentiment']]
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 14640,\n  \"fields\":
[\n    {\n      \"column\": \"text\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 14427,\n
\"samples\": [\n          \"@JetBlue so technically I could drive to
JFK now and put in. Request for tomorrow's flight?\",\n
\"@united why I won't check my carry on. Watched a handler throw this
bag -- miss the conveyer belt -- sat there 10 min
http://t.co/lyoocx5mSH\",\n          \"@SouthwestAir you guys are so
clever \\ud83d\\ude03 http://t.co/qn5odUGFqK\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"airline_sentiment\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"neutral\",\n          \"positive\",\n          \"negative\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

Step 3: Text Preprocessing

```python
stop_words = set(stopwords.words('english'))

def clean_tweet(text):
    text = text.lower()
    text = re.sub(r'http\S+', '', text)      # remove URLs
    text = re.sub(r'@\w+', '', text)          # remove mentions
    text = re.sub(r'#\w+', '', text)          # remove hashtags
    text = re.sub(r'[^a-z\s]', '', text)      # remove special chars
    tokens = text.split()
    tokens = [word for word in tokens if word not in stop_words]
    return " ".join(tokens)

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

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 14640,\n  \"fields\":
[\n    {\n      \"column\": \"text\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 14427,\n
\"samples\": [\n          \"@JetBlue so technically I could drive to
JFK now and put in. Request for tomorrow's flight?\",\n
\"@united why I won't check my carry on. Watched a handler throw this
bag -- miss the conveyer belt -- sat there 10 min
http://t.co/lyoocx5mSH\",\n          \"@SouthwestAir you guys are so
clever \\ud83d\\ude03 http://t.co/qn5odUGFqK\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"airline_sentiment\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"neutral\",\n          \"positive\",\n          \"negative\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"clean_text\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 14046,\n        \"samples\": [\n
\"rude grouchy agent dulles checkin got hour trip lousy start needs
nap something\",\n          \"smart makes angry mt weve partnered
bring free digital access onboard\",\n          \"really drunk someone
stop serving\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

Step 4: Filter Negative Tweets

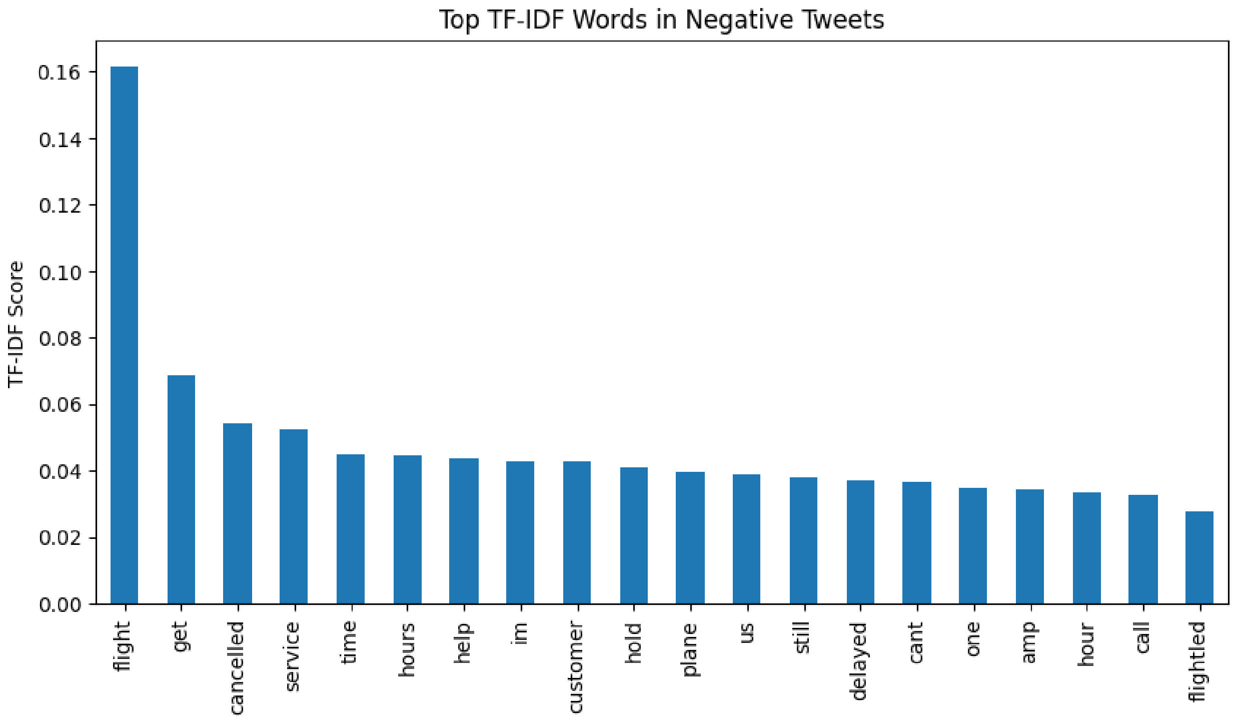
```
negative_tweets = df[df['airline_sentiment'] == 'negative']
negative_tweets.shape

(9178, 3)
```

Step 5: TF-IDF Vectorization

```
vectorizer = TfidfVectorizer(max_features=20)
tfidf_matrix = vectorizer.fit_transform(negative_tweets['clean_text'])

tfidf_df = pd.DataFrame(
    tfidf_matrix.toarray(),
    columns=vectorizer.get_feature_names_out()
)

tfidf_df.mean().sort_values(ascending=False)

flight       0.161364
get          0.068515
cancelled    0.053937
service      0.052331
time         0.044687
hours        0.044553
help         0.043687
im           0.042881
customer     0.042700
hold         0.040962
plane        0.039804
us           0.038649
still        0.037706
delayed      0.037145
cant         0.036345
one          0.034583
amp          0.034461
hour         0.033592
call         0.032677
flightled    0.027828
dtype: float64
```

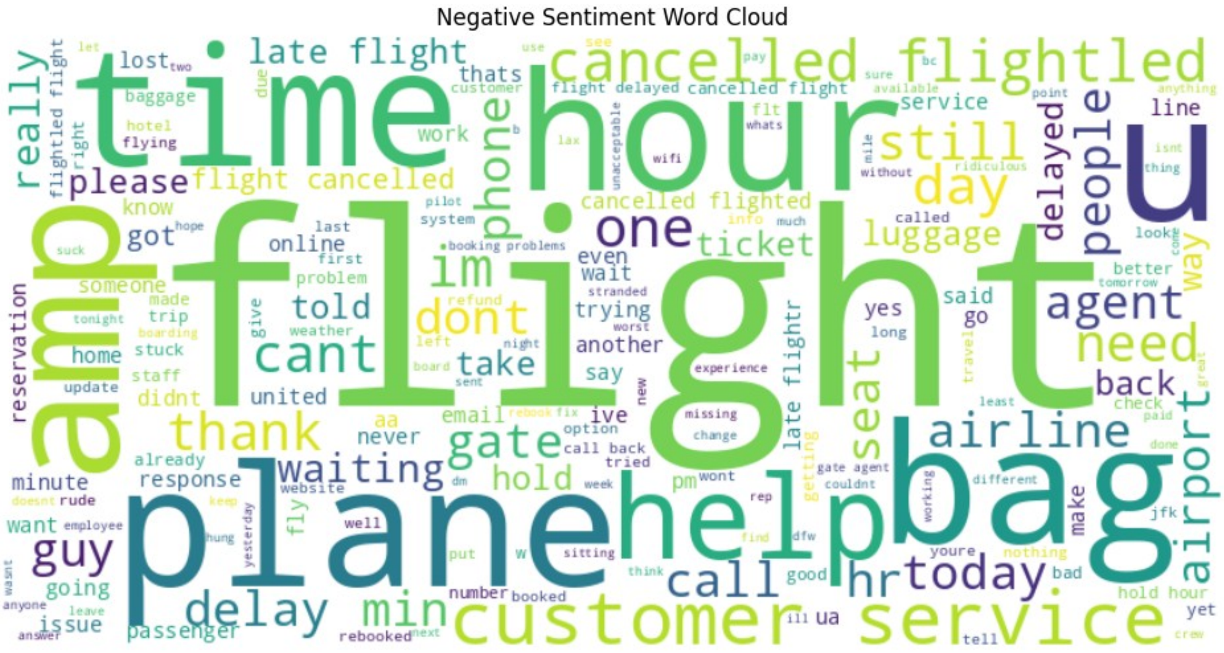
Step 6: Bar Chart (Top Negative Words)

```
top_words = tfidf_df.mean().sort_values(ascending=False)

plt.figure(figsize=(10,5))
top_words.plot(kind='bar')
plt.title("Top TF-IDF Words in Negative Tweets")
plt.ylabel("TF-IDF Score")
plt.show()
```



Step 7: Word Cloud

```
wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white'
).generate(" ".join(negative_tweets['clean_text']))

plt.figure(figsize=(12,6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Negative Sentiment Word Cloud")
plt.show()
```



Discussion

This assignment applied NLP preprocessing techniques to clean and prepare Twitter data for analysis. TF-IDF was used to identify important words contributing to negative airline sentiment. The TF-IDF matrix enabled quantitative comparison of terms across tweets. Bar chart and word cloud visualizations clearly highlighted frequent negative sentiment keywords. The results demonstrate the effectiveness of TF-IDF in extracting meaningful insights from social media text.