```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
import seaborn as sns

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

from sklearn.feature_extraction.text import TfidfVectorizer
from wordcloud import WordCloud
```

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

	airline_sentiment	text
0	negative	@united Flight delayed again! Very disappointed.
1	negative	@americanair Worst service ever. No response f...
2	positive	@delta Thanks for the quick support! Great ser...
3	neutral	@southwest Flight was okay, nothing special.
4	negative	@united Lost my baggage and no help from staff.

Next steps: Generate code with df New interactive sheet

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

	text	airline_sentiment
0	@united Flight delayed again! Very disappointed.	negative
1	@americanair Worst service ever. No response f...	negative
2	@delta Thanks for the quick support! Great ser...	positive
3	@southwest Flight was okay, nothing special.	neutral
4	@united Lost my baggage and no help from staff.	negative

Next steps: Generate code with df New interactive sheet

```
nltk.download('punkt')
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

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

```
def clean_tweet(text):
    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-zA-Z\s]", "", text)     # remove special chars
    text = text.lower()
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
    return " ".join(tokens)
```

```
nltk.download('punkt_tab', quiet=True)
df['clean_text'] = df['text'].apply(clean_tweet)
df.head()
```

	text	airline_sentiment	clean_text
0	@united Flight delayed again! Very disappointed.	negative	flight delayed disappointed
1	@americanair Worst service ever. No response f...	negative	worst service ever response hours
2	@delta Thanks for the quick support! Great ser...	positive	thanks quick support great service
3	@southwest Flight was okay, nothing special.	neutral	flight okay nothing special
4	@united Lost my baggage and no help from staff.	negative	lost baggage help staff

Next steps: Generate code with df New interactive sheet

```
df[['text', 'clean_text']].sample(5)
```

	text	clean_text
0	@united Flight delayed again! Very disappointed.	flight delayed disappointed
8	@united Appreciate the friendly crew!	appreciate friendly crew
4	@united Lost my baggage and no help from staff.	lost baggage help staff
2	@delta Thanks for the quick support! Great ser...	thanks quick support great service
5	@delta Loved the smooth flight experience.	loved smooth flight experience

```
negative_df = df[df['airline_sentiment'] == 'negative']
negative_texts = negative_df['clean_text']
```

```
tfidf = TfidfVectorizer(max_features=1000)
tfidf_matrix = tfidf.fit_transform(negative_texts)

tfidf_matrix
```

```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 20 stored elements and shape (5, 19)>
```

```
tfidf_df = pd.DataFrame(
    tfidf_matrix.toarray(),
    columns=tfidf.get_feature_names_out()
)
tfidf_df.head()
```

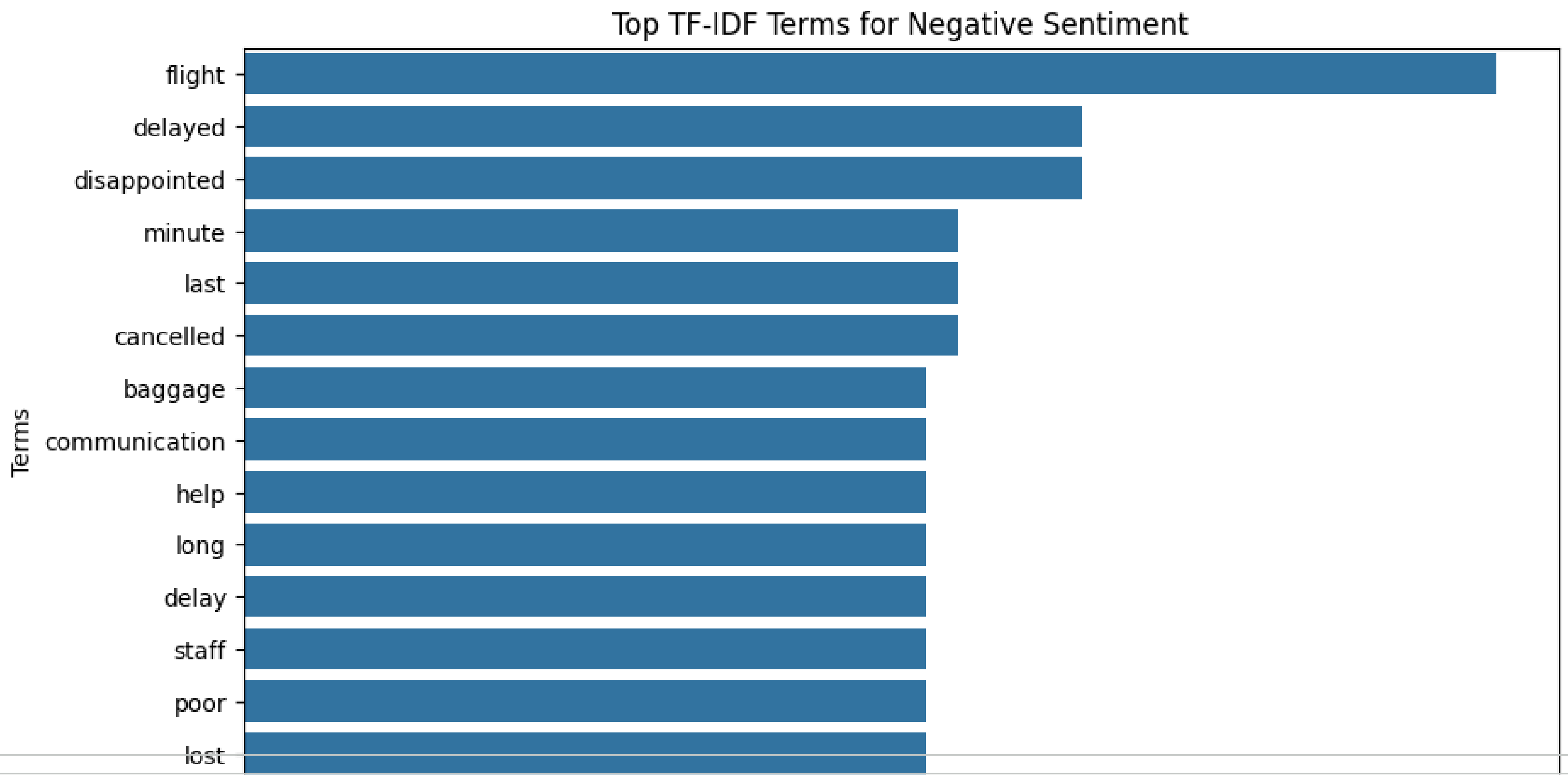
	baggage	cancelled	communication	delay	delayed	disappointed	ever	flight	help	hours	last	long	lost	m
0	0.0	0.000000	0.0	0.0	0.614189	0.614189	0.000000	0.495524	0.0	0.000000	0.000000	0.0	0.0	0.0
1	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.447214	0.000000	0.0	0.447214	0.000000	0.0	0.0	0.0
2	0.5	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.5	0.000000	0.000000	0.0	0.5	0.0
3	0.0	0.523358	0.0	0.0	0.000000	0.000000	0.000000	0.422242	0.0	0.000000	0.523358	0.0	0.0	0.5
4	0.0	0.000000	0.5	0.5	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.5	0.0	0.0

Next steps: Generate code with tfidf_df New interactive sheet

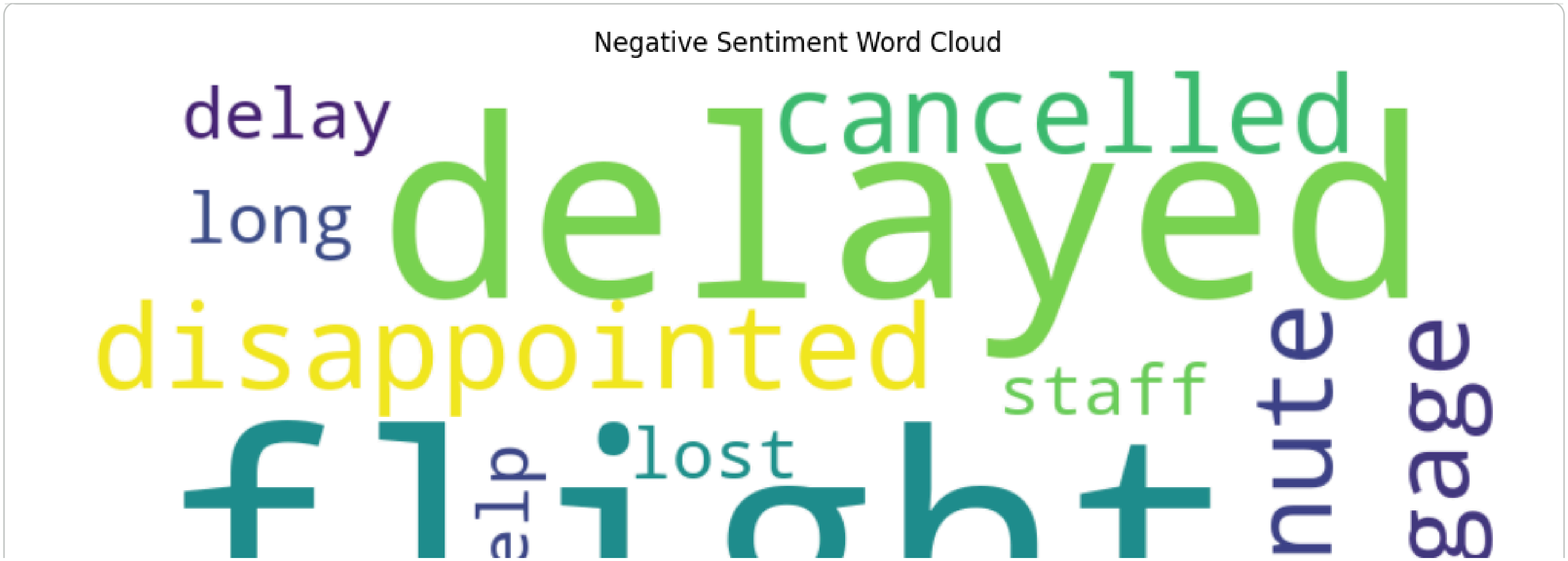
```
mean_tfidf = tfidf_df.mean().sort_values(ascending=False)
top_terms = mean_tfidf.head(15)

top_terms
```

```
                      0
flight         0.183553
delayed        0.122838
disappointed   0.122838
minute         0.104672
last           0.104672
cancelled      0.104672
baggage        0.100000
communication  0.100000
help           0.100000
long           0.100000
delay          0.100000
staff          0.100000
poor           0.100000
lost           0.100000
ever           0.089443

dtype: float64
```

```
plt.figure(figsize=(10,6))
sns.barplot(x=top_terms.values, y=top_terms.index)
plt.title("Top TF-IDF Terms for Negative Sentiment")
plt.xlabel("TF-IDF Score")
plt.ylabel("Terms")
plt.show()
```



```
wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white'
).generate_from_frequencies(top_terms)

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


Negative Sentiment Word Cloud
delayed
flight
cancelled
delay
long
disappointed
staff
lost
nute
gage
elp