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

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

```
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
df = pd.read_csv('/content/twitter_sentiment_sample.csv')
df.head()
```

	text	airline_sentiment
0	The flight was delayed for hours and customer ...	negative
1	Worst airline experience ever lost my baggage	negative
2	Seats were uncomfortable and the staff was rude	negative
3	Flight cancellation without proper notificatio...	negative
4	Poor service and long waiting time at the airport	negative

Next steps: Generate code with df New interactive sheet

```
df.shape
```

```
(15, 2)
```

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

	text	airline_sentiment
0	The flight was delayed for hours and customer ...	negative
1	Worst airline experience ever lost my baggage	negative
2	Seats were uncomfortable and the staff was rude	negative
3	Flight cancellation without proper notificatio...	negative
4	Poor service and long waiting time at the airport	negative

Next steps: Generate code with df New interactive sheet

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

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 = re.sub(r'[^a-z\s]', '', text)         # remove special characters

    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
```

```
    return " ".join(tokens)
```

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

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

	text	airline_sentiment	clean_text
0	The flight was delayed for hours and customer ...	negative	flight delayed hours customer service terrible
1	Worst airline experience ever lost my baggage	negative	worst airline experience ever lost baggage
2	Seats were uncomfortable and the staff was rude	negative	seats uncomfortable staff rude
3	Flight cancellation without proper notificatio...	negative	flight cancellation without proper notificatio...
4	Poor service and long waiting time at the airport	negative	poor service long waiting time airport

Next steps: Generate code with df New interactive sheet

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

```
(10, 3)
```

```
vectorizer = TfidfVectorizer(max_features=1000)
tfidf_matrix = vectorizer.fit_transform(negative_tweets['clean_text'])
```

```
tfidf_matrix.shape
```

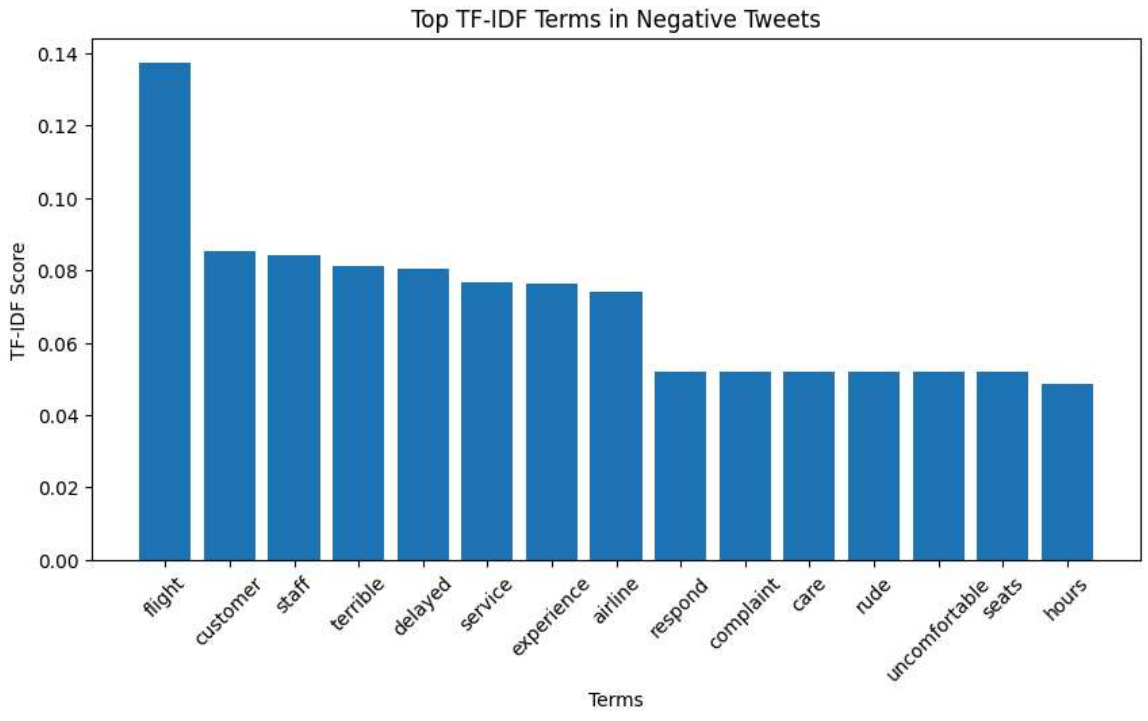
```
(10, 43)
```

```
feature_names = vectorizer.get_feature_names_out()
tfidf_scores = np.mean(tfidf_matrix.toarray(), axis=0)

tfidf_df = pd.DataFrame({
    'term': feature_names,
    'score': tfidf_scores
})

top_terms = tfidf_df.sort_values(by='score', ascending=False).head(15)
top_terms
```

	term	score
16	flight	0.137245
9	customer	0.085328
33	staff	0.084042
36	terrible	0.081251
11	delayed	0.080386
32	service	0.076804
15	experience	0.076412
0	airline	0.074164
28	respond	0.051829
7	complaint	0.051829
6	care	0.051829
30	rude	0.051829
38	uncomfortable	0.051829
31	seats	0.051829
18	hours	0.048546

Next steps: Generate code with top_terms | New interactive sheet

```
plt.figure(figsize=(10,5))
plt.bar(top_terms['term'], top_terms['score'])
plt.xticks(rotation=45)
plt.title("Top TF-IDF Terms in Negative Tweets")
plt.xlabel("Terms")
plt.ylabel("TF-IDF Score")
plt.show()
```




```
wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white'
).generate_from_frequencies(
    dict(zip(tfidf_df['term'], tfidf_df['score']))
)
```

```
)

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