

```
# Data handling
import pandas as pd
import numpy as np

# Text preprocessing
import re
import nltk
nltk.download('punkt')
nltk.download('stopwords')

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

# TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

# Visualization
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

```
[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.
```

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

	text	airline_sentiment	grid
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(tweet):
    tweet = tweet.lower()                      # lowercase
    tweet = re.sub(r'http\S+', '', tweet)       # remove URLs
    tweet = re.sub(r'@\w+', '', tweet)          # remove mentions
    tweet = re.sub(r'\#\w+', '', tweet)          # remove hashtags
    tweet = re.sub(r'[^a-z\s]', '', tweet)        # remove symbols
    tokens = word_tokenize(tweet)                # tokenize
    tokens = [w for w in tokens if w not in stop_words] # remove stopwords
    return ' '.join(tokens)
```

```
import nltk
nltk.download('punkt_tab')
df['cleaned_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	cleaned_text	grid
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_df = df[df['airline_sentiment'] == 'negative']
negative_df
```

	text	airline_sentiment	cleaned_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	
5	Delayed boarding and no response from support ...	negative	delayed boarding response support team	
6	Food quality was bad and the flight was overcr...	negative	food quality bad flight overcrowded	
7	Customer care did not respond to my complaint	negative	customer care respond complaint	
8	Missed my connecting flight due to airline delay	negative	missed connecting flight due airline delay	
9	Terrible experience the flight was late and st...	negative	terrible experience flight late staff unhelpful	

Next steps: [Generate code with negative_df](#) [New interactive sheet](#)

```
vectorizer = TfidfVectorizer(max_features=1000)
tfidf_matrix = vectorizer.fit_transform(negative_df['cleaned_text'])

tfidf_matrix.shape
```

(10, 43)

```
terms = vectorizer.get_feature_names_out()
scores = tfidf_matrix.mean(axis=0).A1

tfidf_df = pd.DataFrame({
    'Term': terms,
    'Score': scores
})

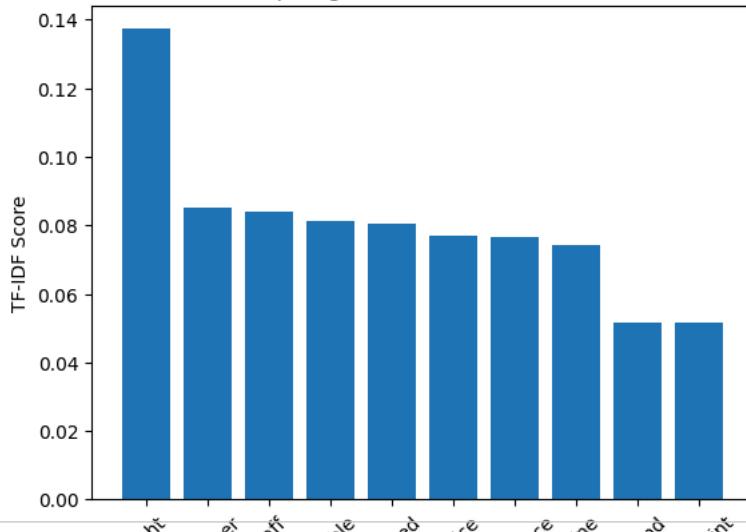
top_terms = tfidf_df.sort_values(by='Score', ascending=False).head(10)
top_terms
```

	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	

Next steps: [Generate code with top_terms](#) [New interactive sheet](#)

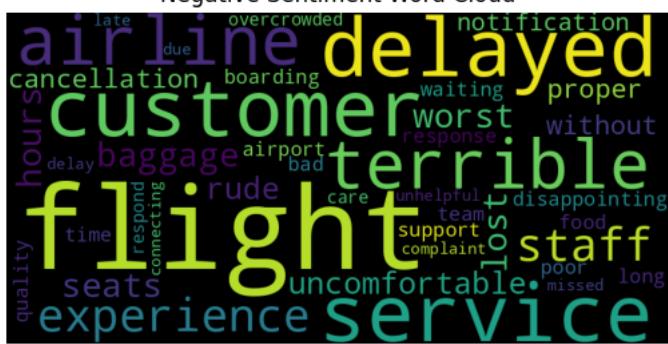
```
plt.figure()
plt.bar(top_terms['Term'], top_terms['Score'])
plt.xticks(rotation=45)
plt.xlabel("Terms")
plt.ylabel("TF-IDF Score")
plt.title("Top Negative Sentiment Terms")
plt.show()
```

Top Negative Sentiment Terms



```
text = ".join(negative_df['cleaned_text'])\n\nwc = WordCloud(width=800, height=400).generate(text)\n\nplt.figure()\nplt.imshow(wc)\nplt.axis("off")\nplt.title("Negative Sentiment Word Cloud")\nplt.show()
```

Negative Sentiment Word Cloud



Start coding or generate with AI.