

```
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: cyemem2<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,!<1.8.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.11.10)
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,!<1.8.0) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!<1.8.0) (2.21.1)
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!<1.8.0) (4.14.1)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!<1.8.0) (0.4.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0) (3.10.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0) (2025.11.11)
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) (1.3.0)
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) (0.0.4)
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) (0.19.0)
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) (7.0.5)
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) (1.3.0)
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) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>wordcloud) (1.4.7)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>wordcloud) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib>wordcloud) (2.9.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7) (1.17.0)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1) (3.13.1)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1) (0.4.3)
```

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



```
import pandas as pd

data = {
    "text": [
        "Flight delayed again very frustrating experience",
        "Worst airline service ever never booking again",
        "Seats were uncomfortable and staff was rude",
        "Customer support did not respond for hours",
        "Flight got cancelled without proper notice",
    ]
}
```

```


    "The airline experience was amazing and smooth",
    "Staff were friendly and helpful",
    "Average flight nothing special",
    "Food quality was okay"
],
"airline_sentiment": [
    "negative",
    "negative",
    "negative",
    "negative",
    "negative",
    "positive",
    "positive",
    "neutral",
    "neutral"
]
}

df = pd.DataFrame(data)
df
```

| | text | airline_sentiment |  |
|---|--|-------------------|---|
| 0 | Flight delayed again very frustrating experience | negative |  |
| 1 | Worst airline service ever never booking again | negative | |
| 2 | Seats were uncomfortable and staff was rude | negative | |
| 3 | Customer support did not respond for hours | negative | |
| 4 | Flight got cancelled without proper notice | negative | |
| 5 | The airline experience was amazing and smooth | positive | |
| 6 | Staff were friendly and helpful | positive | |
| 7 | Average flight nothing special | neutral | |
| 8 | Food quality was okay | neutral | |


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

```
df.head()
```

| | text | airline_sentiment |  |
|---|--|-------------------|---|
| 0 | Flight delayed again very frustrating experience | negative | |
| 1 | Worst airline service ever never booking again | negative | |
| 2 | Seats were uncomfortable and staff was rude | negative | |
| 3 | Customer support did not respond for hours | negative | |
| 4 | Flight got cancelled without proper notice | negative | |

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

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

| | text | airline_sentiment |  |
|---|--|-------------------|---|
| 0 | Flight delayed again very frustrating experience | negative | |
| 1 | Worst airline service ever never booking again | negative | |
| 2 | Seats were uncomfortable and staff was rude | negative | |
| 3 | Customer support did not respond for hours | negative | |
| 4 | Flight got cancelled without proper notice | negative | |

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

```
def clean_text(text):
    text = re.sub(r"http\S+", "", text)
    text = re.sub(r"@w+", "", text)
    text = re.sub(r"#w+", "", text)
```

```
text = re.sub(r"[^a-zA-Z]", " ", text)
text = text.lower()
return text
```

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

| | text | airline_sentiment | clean_text |
|---|--|-------------------|--|
| 0 | Flight delayed again very frustrating experience | negative | flight delayed again very frustrating experience |
| 1 | Worst airline service ever never booking again | negative | worst airline service ever never booking again |
| 2 | Seats were uncomfortable and staff was rude | negative | seats were uncomfortable and staff was rude |
| 3 | Customer support did not respond for hours | negative | customer support did not respond for hours |
| 4 | Flight got cancelled without proper notice | negative | flight got cancelled without proper notice |

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

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

def tokenize_remove_stopwords(text):
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
    return " ".join(tokens)
```

```
!pip install -q nltk
```

```
import nltk
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
```

```
import pandas as pd
import re

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

```
data = {
    "text": [
        "Flight delayed again very frustrating experience",
        "Worst airline service ever never booking again",
        "Seats were uncomfortable and staff was rude",
        "Customer support did not respond for hours",
        "Flight got cancelled without proper notice",
        "The airline experience was amazing and smooth",
        "Staff were friendly and helpful",
        "Average flight nothing special",
        "Food quality was okay"
    ],
    "airline_sentiment": [
        "negative","negative","negative","negative","negative",
        "positive","positive","neutral","neutral"
    ]
}


df = pd.DataFrame(data)
df
```

| | text | airline_sentiment |  |
|---|--|-------------------|---|
| 0 | Flight delayed again very frustrating experience | negative |  |
| 1 | Worst airline service ever never booking again | negative | |
| 2 | Seats were uncomfortable and staff was rude | negative | |
| 3 | Customer support did not respond for hours | negative | |
| 4 | Flight got cancelled without proper notice | negative | |
| 5 | The airline experience was amazing and smooth | positive | |
| 6 | Staff were friendly and helpful | positive | |
| 7 | Average flight nothing special | neutral | |
| 8 | Food quality was okay | neutral | |


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

```
def clean_text(text):
    text = str(text)
    text = re.sub(r'^a-zA-Z ]', '', text)
    return text.lower()

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

| | text | clean_text |  |
|---|--|--|---|
| 0 | Flight delayed again very frustrating experience | flight delayed again very frustrating experience | |
| 1 | Worst airline service ever never booking again | worst airline service ever never booking again | |
| 2 | Seats were uncomfortable and staff was rude | seats were uncomfortable and staff was rude | |
| 3 | Customer support did not respond for hours | customer support did not respond for hours | |
| 4 | Flight got cancelled without proper notice | flight got cancelled without proper notice | |
| 5 | The airline experience was amazing and smooth | the airline experience was amazing and smooth | |
| 6 | Staff were friendly and helpful | staff were friendly and helpful | |
| 7 | Average flight nothing special | average flight nothing special | |
| 8 | Food quality was okay | food quality was okay | |

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

def tokenize_remove_stopwords(text):
    text = str(text)
    tokens = text.split() #  NO NLTK tokenizer (avoids errors)
    tokens = [word for word in tokens if word not in stop_words]
    return " ".join(tokens)
```

```
df['processed_text'] = df['clean_text'].apply(tokenize_remove_stopwords)
df
```

| | text | airline_sentiment | clean_text | processed_text |
|---|--|-------------------|--|--|
| 0 | Flight delayed again very frustrating experience | negative | flight delayed again very frustrating experience | flight delayed frustrating experience |
| 1 | Worst airline service ever never booking again | negative | worst airline service ever never booking again | worst airline service ever never booking |
| 2 | Seats were uncomfortable and staff was rude | negative | seats were uncomfortable and staff was rude | seats uncomfortable staff rude |
| 3 | Customer support did not respond for hours | negative | customer support did not respond for hours | customer support respond hours |
| 4 | Flight got cancelled without proper notice | negative | flight got cancelled without proper notice | flight got cancelled without proper notice |
| 5 | The airline experience was amazing and smooth | positive | the airline experience was amazing and smooth | airline experience amazing smooth |
| 6 | Staff were friendly and helpful | positive | staff were friendly and helpful | staff friendly helpful |
| 7 | Average flight nothing special | neutral | average flight nothing special | average flight nothing special |
| 8 | Food quality was okay | neutral | food quality was okay | food quality okay |

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

```
negative_df = df[df['airline_sentiment'] == 'negative']
negative_df.head()
```

| | text | airline_sentiment | clean_text | processed_text |
|---|--|-------------------|--|--|
| 0 | Flight delayed again very frustrating experience | negative | flight delayed again very frustrating experience | flight delayed frustrating experience |
| 1 | Worst airline service ever never booking again | negative | worst airline service ever never booking again | worst airline service ever never booking |
| 2 | Seats were uncomfortable and staff was rude | negative | seats were uncomfortable and staff was rude | seats uncomfortable staff rude |
| 3 | Customer support did not respond for hours | negative | customer support did not respond for hours | customer support respond hours |
| 4 | Flight got cancelled without proper notice | negative | flight got cancelled without proper notice | flight got cancelled without proper notice |

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

```
vectorizer = TfidfVectorizer(max_features=20)
tfidf_matrix = vectorizer.fit_transform(negative_df['processed_text'])

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

tfidf_df.head()
```

| | airline | booking | cancelled | customer | delayed | ever | experience | flight | frustrating | got | hours | never | n |
|---|----------|----------|-----------|----------|----------|----------|------------|----------|-------------|----------|-------|----------|-----|
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.523358 | 0.000000 | 0.523358 | 0.422242 | 0.523358 | 0.000000 | 0.0 | 0.000000 | 0.0 |
| 1 | 0.447214 | 0.447214 | 0.000000 | 0.0 | 0.000000 | 0.447214 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.447214 | 0.0 |
| 2 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.000000 | 0.0 |
| 3 | 0.000000 | 0.000000 | 0.000000 | 0.5 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.5 | 0.000000 | 0.0 |
| 4 | 0.000000 | 0.000000 | 0.463693 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.374105 | 0.000000 | 0.463693 | 0.0 | 0.000000 | 0.4 |

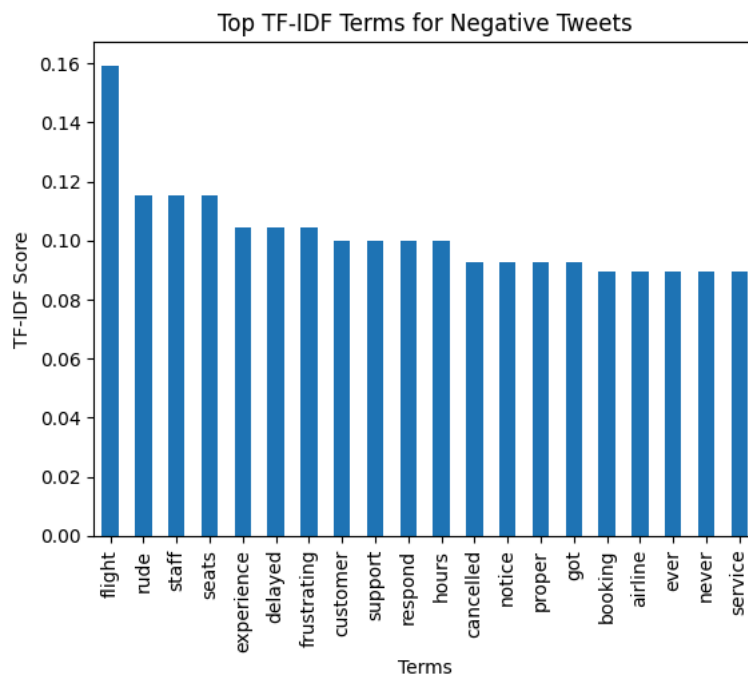
Next steps: [Generate code with tfidf_df](#) [New interactive sheet](#)

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

| | θ |
|-------------|----------|
| flight | 0.159269 |
| rude | 0.115470 |
| staff | 0.115470 |
| seats | 0.115470 |
| experience | 0.104672 |
| delayed | 0.104672 |
| frustrating | 0.104672 |
| customer | 0.100000 |
| support | 0.100000 |
| respond | 0.100000 |
| hours | 0.100000 |
| cancelled | 0.092739 |
| notice | 0.092739 |
| proper | 0.092739 |
| got | 0.092739 |
| booking | 0.089443 |
| airline | 0.089443 |
| ever | 0.089443 |
| never | 0.089443 |
| service | 0.089443 |

dtype: float64

```
plt.figure()
tfidf_scores.plot(kind='bar')
plt.title("Top TF-IDF Terms for Negative Tweets")
plt.xlabel("Terms")
plt.ylabel("TF-IDF Score")
plt.show()
```



```
import matplotlib.pyplot as plt

# Convert TF-IDF scores to sorted values
tfidf_scores = tfidf_scores.sort_values(ascending=False)

# Plot BAR GRAPH
plt.figure(figsize=(10,5))
```

```
plt.bar(tfidf_scores.index, tfidf_scores.values)
plt.xlabel("Terms")
plt.ylabel("TF-IDF Score")
plt.title("Top TF-IDF Terms for Negative Sentiment")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

