```
import os
os.environ['KAGGLE_USERNAME'] = 'sreeyadaripelli'
os.environ['KAGGLE_KEY'] = '5e0752429f497134d5284d136fd300fb'
```

```
import kaggle
kaggle.api.dataset_download_files('crowdflower/twitter-airline-sentiment', path
```

```
Dataset URL: https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentime
```

```
import pandas as pd

tweets_df = pd.read_csv('Tweets.csv')
display(tweets_df.head())
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negative
0	570306133677760513	neutral	1.0000	
1	570301130888122368	positive	0.3486	
2	570301083672813571	neutral	0.6837	
3	570301031407624196	negative	1.0000	Ba
4	570300817074462722	negative	1.0000	C

```
print(tweets_df['airline_sentiment'].value_counts())
```

```
airline_sentiment
negative    9178
neutral     3099
positive    2363
Name: count, dtype: int64
```

```
negative_tweets_df = tweets_df[tweets_df['airline_sentiment'] == 'negative']

print(f"Number of negative sentiment tweets: {len(negative_tweets_df)}")
display(negative_tweets_df.head())
```

```
ment tweets: 9178
```

airline_sentiment	airline_sentiment_confidence	negativ
negative	1.0000	B
negative	1.0000	C
negative	1.0000	C
negative	0.6842	La
negative	1.0000	B

```python
from sklearn.feature_extraction.text import TfidfVectorizer

# Assuming 'text_for_tfidf' column exists from a previous preprocessing step
# If it doesn't, this step would need to be preceded by text cleaning and tokeniz
vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')
tfidf_matrix = vectorizer.fit_transform(tweets_df['text_for_tfidf'])
feature_names = vectorizer.get_feature_names_out()

print("TF-IDF matrix created with shape:", tfidf_matrix.shape)
print("Number of features (terms):"), len(feature_names)
```

```
TF-IDF matrix created with shape: (14640, 5000)
Number of features (terms):
(None, 5000)
```

```python
negative_tweet_indices = negative_tweets_df.index

print(f"Indices of negative tweets (first 5): {negative_tweet_indices[:5].tolist(
print(f"Total number of negative tweet indices: {len(negative_tweet_indices)}")
```

```
Indices of negative tweets (first 5): [3, 4, 5, 15, 17]
Total number of negative tweet indices: 9178
```

```python
negative_tfidf_matrix = tfidf_matrix[negative_tweet_indices]

# Calculate the sum of TF-IDF scores for each term across all negative tweets
sum_tfidf_negative = negative_tfidf_matrix.sum(axis=0)

print(f"Shape of negative TF-IDF matrix: {negative_tfidf_matrix.shape}")
print(f"Shape of sum of TF-IDF scores for negative terms: {sum_tfidf_negative.sha
[illegible]
```

```
Shape of negative TF-IDF matrix: (9178, 5000)
Shape of sum of TF-IDF scores for negative terms: (1, 5000)
```

```
import pandas as pd

# Convert the sparse matrix row vector to a dense NumPy array
sum_tfidf_negative_array = sum_tfidf_negative.A.flatten()

# Create a Pandas Series with feature names as index and TF-IDF scores as value
tfidf_scores_negative = pd.Series(sum_tfidf_negative_array, index=feature_names

# Sort the Series in descending order
tfidf_scores_negative = tfidf_scores_negative.sort_values(ascending=False)

# Display the top 20 terms and their scores
print("Top 20 terms with highest TF-IDF scores in negative tweets:")
display(tfidf_scores_negative.head(20))
```

```
Top 20 terms with highest TF-IDF scores in negative tweets:
```

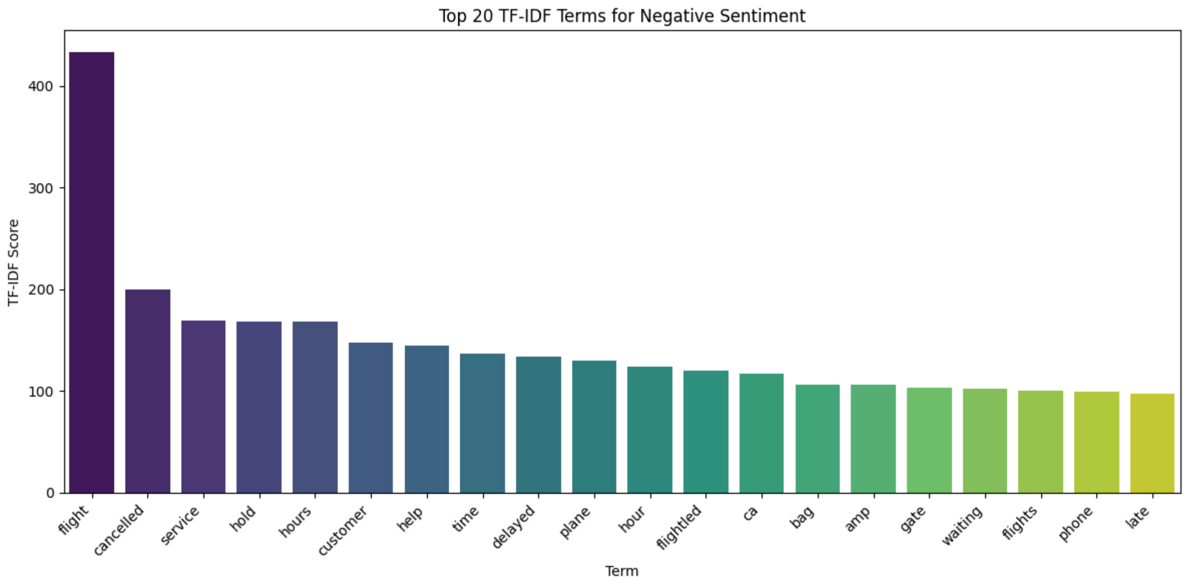
	0
flight	433.050935
cancelled	199.923383
service	169.135761
hold	168.683727
hours	168.320074
customer	147.458089
help	144.685472
time	136.412815
delayed	133.534531
plane	129.621976
hour	124.038666
flightled	119.962833
ca	117.141493
bag	106.595413
amp	106.444929
gate	103.755449
waiting	102.625705
flights	100.759696
phone	99.908555
late	97.032667

dtype: float64

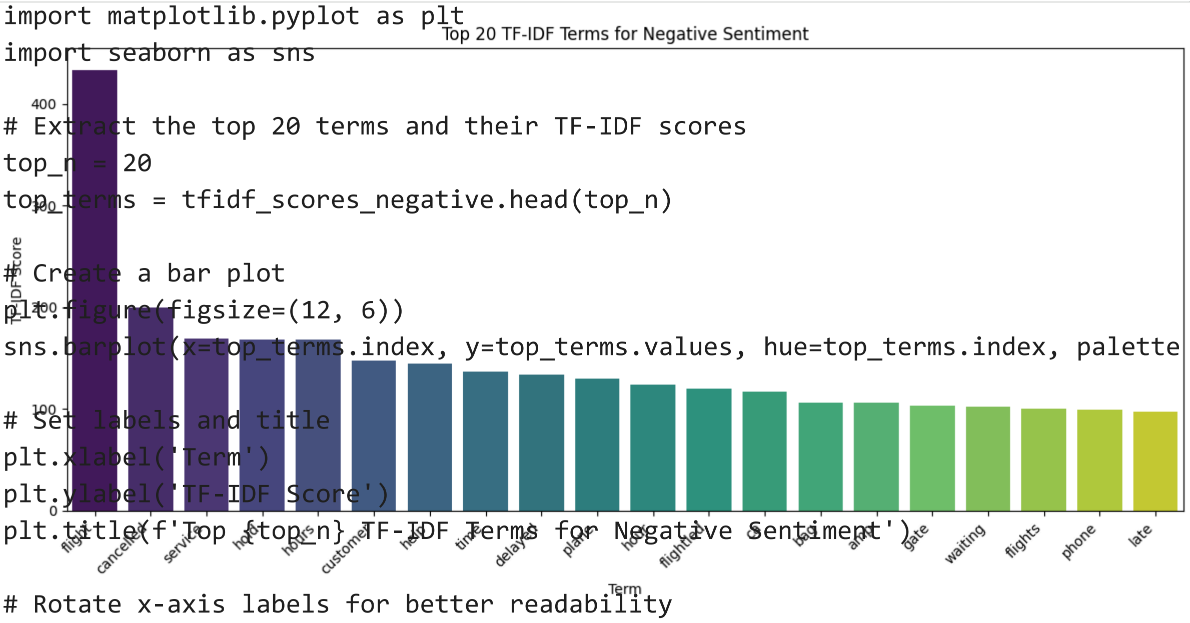
```
import matplotlib.pyplot as plt
import seaborn as sns

# Extract the top 20 terms and their TF-IDF scores
top_n = 20
top_terms = tfidf_scores_negative.head(top_n)

# Create a bar plot
plt.figure(figsize=(12, 6))
sns.barplot(x=top_terms.index, y=top_terms.values, palette='viridis')

# Set labels and title
plt.xlabel('Term')
plt.ylabel('TF-IDF Score')
plt.title(f'Top {top_n} TF-IDF Terms for Negative Sentiment')

# Rotate x-axis labels for better readability
```

```
plt.xticks(rotation=45, ha='right')

# Adjust plot parameters for a tight layout
plt.tight_layout()

# Display the plot
plt.show()
```

```
/tmp/ipython-input-1961476244.py:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v

  sns.barplot(x=top_terms.index, y=top_terms.values, palette='viridis')
```



```
import matplotlib.pyplot as plt
import seaborn as sns

# Extract the top 20 terms and their TF-IDF scores
top_n = 20
top_terms = tfidf_scores_negative.head(top_n)

# Create a bar plot
plt.figure(figsize=(12, 6))
sns.barplot(x=top_terms.index, y=top_terms.values, hue=top_terms.index, palette

# Set labels and title
plt.xlabel('Term')
plt.ylabel('TF-IDF Score')
plt.title(f'Top {top_n} TF-IDF Terms for Negative Sentiment')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')
```



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

print("WordCloud library imported successfully.")
```

```
WordCloud library imported successfully.
```

```
wordcloud_dict = tfidf_scores_negative.to_dict()

wordcloud = WordCloud(width=800, height=400, background_color='white', colormap=

plt.figure(figsize=(15, 7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Top TF-IDF Terms for Negative Sentiment')
plt.show()
```

Word Cloud of Top TF-IDF Terms for Negative Sentiment
customer
hours
hour
customers
terrible
getting
weather
yes
care
talk
sent
mins
week
took
aa
voucher
mechanical
use
understand
contact
worse
dfw