

GUJJA RAJU 2403A52018 CSE AIML BATCH-02

Twitter Sentiment Analysis (Negative Sentiment TF-IDF)

This notebook loads the Twitter US Airline Sentiment dataset, preprocesses tweets, computes TF-IDF, and visualizes top negative sentiment terms.

```
# Install required libraries (run once)
!pip install pandas numpy scikit-learn nltk matplotlib wordcloud
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: wordcloud in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: cyclor>=0.10 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages
```

```
import pandas as pd
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

```

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab') # Added to resolve LookupError

```

```

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

```

```

# Load dataset (update path after downloading from Kaggle)
df = pd.read_csv("Tweets.csv")
df = df[['text', 'airline_sentiment']]
df.head()

```

| | text | airline_sentiment |
|---|---|-------------------|
| 0 | @VirginAmerica What @dhepburn said. | neutral |
| 1 | @VirginAmerica plus you've added commercials t... | positive |
| 2 | @VirginAmerica I didn't today... Must mean I n... | neutral |
| 3 | @VirginAmerica it's really aggressive to blast... | negative |
| 4 | @VirginAmerica and it's a really big bad thing... | negative |

```

# Text preprocessing
stop_words = set(stopwords.words('english'))

def clean_tweet(text):
    text = re.sub(r"http\S+|www\S+", "", text) # remove URLs
    text = re.sub(r"@[A-Za-z0-9_]+", "", text) # remove mentions
    text = re.sub(r"#\w+", "", text) # remove hashtags
    text = re.sub(r"[^a-zA-Z ]", "", text)
    tokens = word_tokenize(text.lower())
    tokens = [w for w in tokens if w not in stop_words]
    return " ".join(tokens)

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

```

| | text | airline_sentiment | clean_text |
|---|---|-------------------|---|
| 0 | @VirginAmerica What @dhepburn said. | neutral | said |
| 1 | @VirginAmerica plus you've added commercials t... | positive | plus youve added commercials experience tacky |
| 2 | @VirginAmerica I didn't today... Must mean I n... | neutral | didnt today must mean need take another trip |
| 3 | @VirainAmerica it's really .. | negative | really aggressive blast |

```
# Filter negative sentiment tweets
neg_df = df[df['airline_sentiment'] == 'negative']
```

```
# TF-IDF Vectorization
vectorizer = TfidfVectorizer(max_features=1000)
tfidf_matrix = vectorizer.fit_transform(neg_df['clean_text'])

tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=vectorizer.get_feature_names())
tfidf_df.head()
```

| | aa | able | absolute | absolutely | acceptable | access | accommodate | account | act |
|---|-----|------|----------|------------|------------|--------|-------------|---------|-----|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

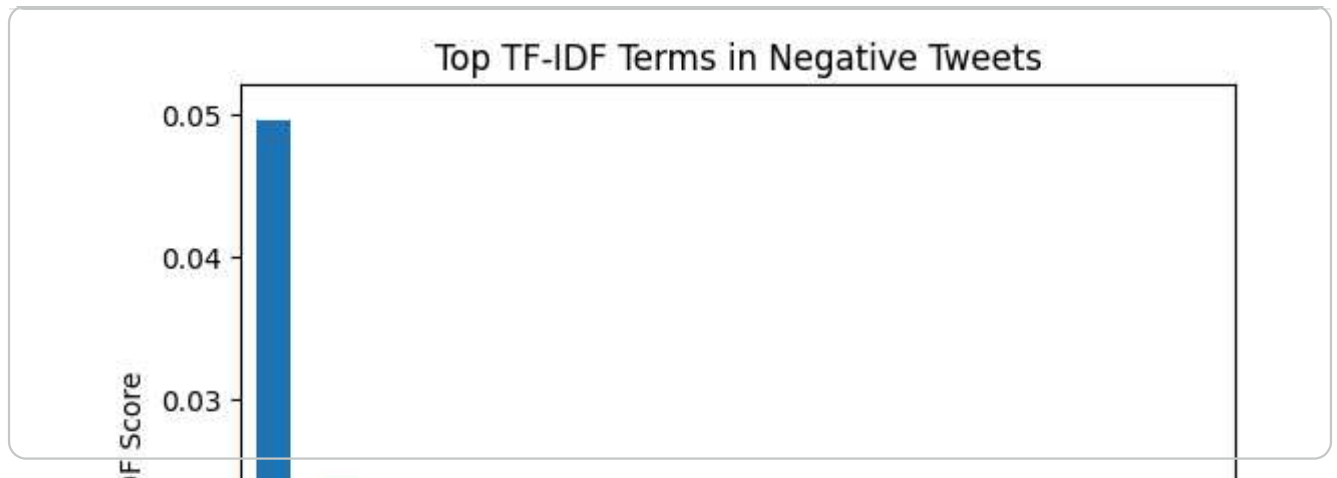
5 rows × 1000 columns

```
# Top TF-IDF terms (average score)
top_terms = tfidf_df.mean(axis=0).sort_values(ascending=False).head(15)
top_terms
```

| | 0 |
|------------------|----------|
| flight | 0.049564 |
| get | 0.023834 |
| cancelled | 0.021974 |
| service | 0.020094 |
| hours | 0.017978 |
| hold | 0.017853 |
| customer | 0.017246 |
| help | 0.016586 |
| time | 0.015804 |
| im | 0.015623 |
| plane | 0.014714 |
| delayed | 0.014674 |
| still | 0.014483 |
| cant | 0.014064 |
| us | 0.014027 |

dtype: float64

```
# Bar chart of top negative sentiment terms
plt.figure()
top_terms.plot(kind='bar')
plt.title("Top TF-IDF Terms in Negative Tweets")
plt.ylabel("TF-IDF Score")
plt.show()
```



```
# Word Cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_

plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

