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

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

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
df = pd.read_csv("/content/twitter_datas
```

```
df.head()
```

	Tweet_ID	Username	Text	Retweets	Likes	Timestamp
0	1	julie81	Party least receive say or single. Prevent pre...	2	25	2023-01-30 11:00:51
1	2	richardhester	Hotel still Congress may member staff. Media d...	35	29	2023-01-02 22:45:58
2	3	williamsjoseph	Nice be her debate industry that year. Film wh...	51	25	2023-01-18 11:25:19
3	4	danielsmary	Laugh explain situation career occur serious. ...	37	18	2023-04-10 22:06:29
4	5	carlwarren	Involve sense former often approach government...	27	80	2023-01-24 07:12:21

```
def clean_tweet(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\s]", "", text)
    text = text.lower().strip()
    return text

df['clean_text'] = df['Text'].apply(clean_tweet)
```

```
stop_words = set(stopwords.words('english'))
df['clean_text'] = df['clean_text'].apply(
    lambda x: " ".join([word for word in x.split() if word not in stop_words])
)

df[['Text','clean_text']].head()
```

	Text	clean_text
0	Party least receive say or single. Prevent pre...	party least receive say single prevent prevent...
1	Hotel still Congress may member staff. Media d...	hotel still congress may member staff media dr...
2	Nice be her debate industry that year. Film wh...	nice debate industry year film generation push...
3	Laugh explain situation career occur serious. ...	laugh explain situation career occur serious f...
4	Involve sense former often approach government...	involve sense former often approach government...

```
vectorizer = TfidfVectorizer(max_features=5000)
tfidf_matrix = vectorizer.fit_transform(df['clean_text'])
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=vectorizer.get_feature_names_out())
tfidf_df.head()
```

	ability	able	accept	according	account	across	act	action	activity	actually	...	would	write	writer	wrong
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	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	0.0	...	0.198325	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	0.0	...	0.000000	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	0.0	...	0.000000	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	0.0	...	0.000000	0.0	0.0	0.0

5 rows × 869 columns

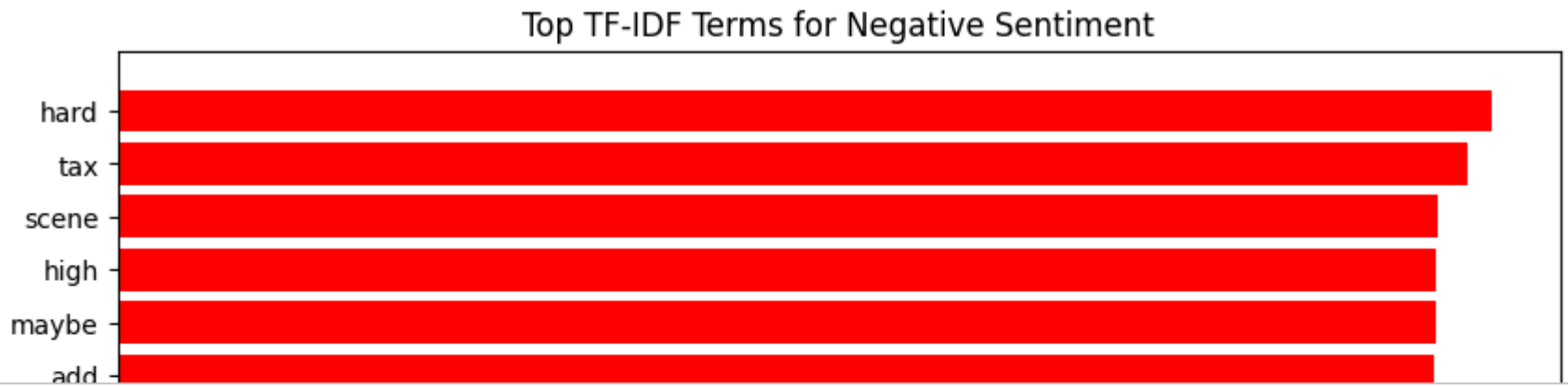
```
all_tweets_clean = df['clean_text']
neg_vectorizer = TfidfVectorizer(max_features=5000)
neg_tfidf_matrix = neg_vectorizer.fit_transform(all_tweets_clean)
neg_scores = np.sum(neg_tfidf_matrix.toarray(), axis=0)
terms = neg_vectorizer.get_feature_names_out()
top_indices = neg_scores.argsort()[-15:][::-1]

top_terms = [(terms[i], neg_scores[i]) for i in top_indices]
top_terms
```

```
[('hard', np.float64(71.41534929291903)),
 ('tax', np.float64(70.14749331144758)),
 ('scene', np.float64(68.64151786374853)),
 ('high', np.float64(68.55526056548001)),
 ('maybe', np.float64(68.52335127253716)),
 ('add', np.float64(68.4827306318259)),
 ('yard', np.float64(68.45616720332833)),
 ('senior', np.float64(68.44784533370833)),
 ('forget', np.float64(68.37092712543082)),
 ('food', np.float64(68.36181096915784)),
 ('success', np.float64(68.31790823032573)),
 ('job', np.float64(68.27858869587938)),
 ('young', np.float64(68.02581644631354)),
```

```
 ('man', np.float64(67.79450279096052)),
 ('agree', np.float64(67.7487058985991))]
```

```
terms, scores = zip(*top_terms)
plt.figure(figsize=(10,6))
plt.barh(terms, scores, color='red')
plt.xlabel("TF-IDF Score")
plt.title("Top TF-IDF Terms for Negative Sentiment")
plt.gca().invert_yaxis()
plt.show()
```




```
wordcloud = WordCloud(width=800, height=
plt.figure(figsize=(12,6))
plt.imshow(wordcloud, interpolation='bil
plt.axis("off")
plt.title("Word Cloud - All Tweets Vocab
plt.show()
```

Word Cloud - All Tweets Vocabulary