```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.feature_extraction.text import TfidfVectorizer

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

from wordcloud import WordCloud
```

```
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
```

```
url = "/content/twitter_dataset.csv"
df = pd.read_csv(url)

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

Next steps: Generate code with df New interactive sheet

```
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'@\w+', '', text)            # remove mentions
    text = re.sub(r'#\w+', '', text)            # remove hashtags
    text = re.sub(r'[^a-zA-Z\s]', '', text)     # remove punctuation
    text = text.lower().strip()                 # lowercase

    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]

    return " ".join(tokens)
```

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

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

	Text	cleaned_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...

```
tfidf = TfidfVectorizer(max_features=1000)

tfidf_matrix = tfidf.fit_transform(df['cleaned_text'])

print("TF-IDF Matrix Shape:", tfidf_matrix.shape)
```

```
TF-IDF Matrix Shape: (10000, 869)
```

```
feature_names = tfidf.get_feature_names_out()
tfidf_scores = tfidf_matrix.mean(axis=0).A1

tfidf_df = pd.DataFrame({
    'term': feature_names,
    'score': tfidf_scores
})

top_terms = tfidf_df.sort_values(by='score', ascending=False).head(15)
top_terms
```

	term	score
332	hard	0.007142
763	tax	0.007015
652	scene	0.006864
339	high	0.006856
444	maybe	0.006852
10	add	0.006848
863	yard	0.006846
668	senior	0.006845
294	forget	0.006837
290	food	0.006836
749	success	0.006832
381	job	0.006828
868	young	0.006803
434	man	0.006779
20	agree	0.006775

Next steps: Generate code with top_terms New interactive sheet

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

tfidf_df.head()
```

	ability	able	accept	according	account	across	act	action	activity	actually	...	would	write	writer	wrong	yar
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	0.00000
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	0.00000
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	0.18348
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	0.00000
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	0.00000

5 rows × 869 columns

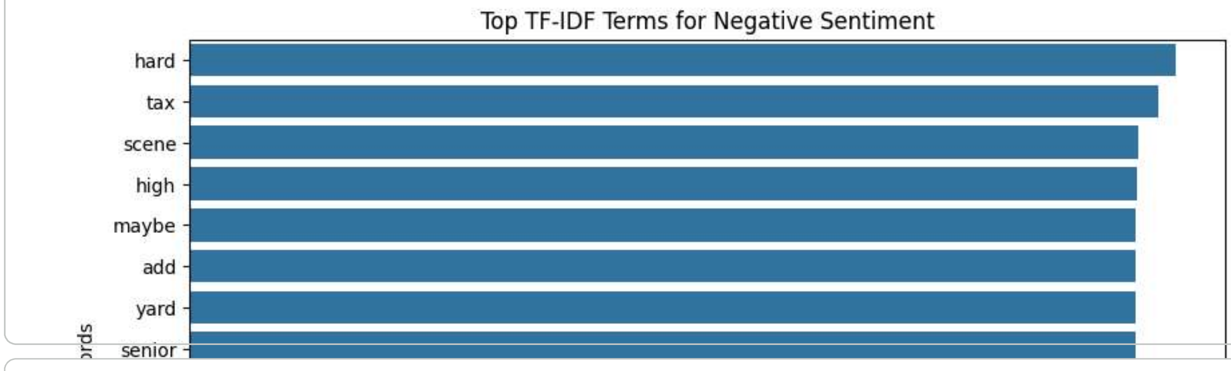
```
tfidf_scores = tfidf_df.mean(axis=0)
top_terms = tfidf_scores.sort_values(ascending=False).head(15)

top_terms
```

	0
hard	0.007142
tax	0.007015
scene	0.006864
high	0.006856
maybe	0.006852
add	0.006848
yard	0.006846
senior	0.006845
forget	0.006837
food	0.006836
success	0.006832
job	0.006828
young	0.006803
man	0.006779
agree	0.006775

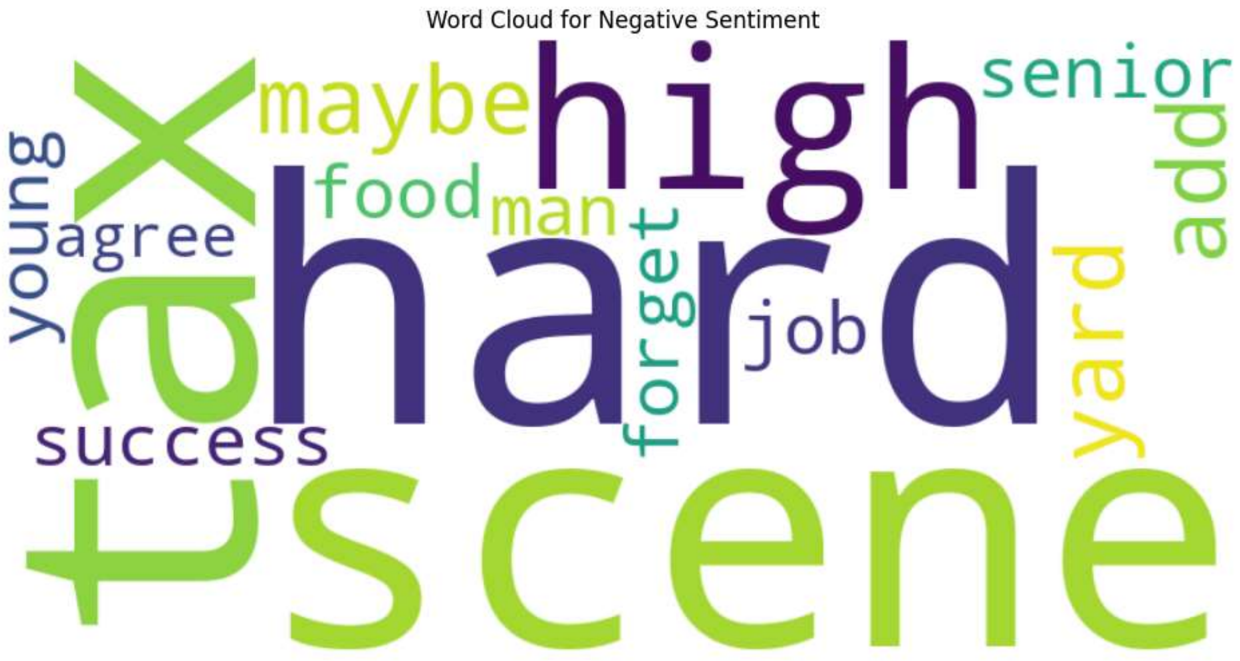
dtype: float64

```
average_tfidf = tfidf_df.mean(axis=0)
```

```
plt.figure(figsize=(10,6))
sns.barplot(x=top_terms.values, y=top_terms.index)
plt.title("Top TF-IDF Terms for Negative Sentiment")
plt.xlabel("TF-IDF Score")
plt.ylabel("Words")
plt.show()
```



```
wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white'
).generate_from_frequencies(top_terms)

plt.figure(figsize=(12,6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Word Cloud for Negative Sentiment")
plt.show()
```


Word Cloud for Negative Sentiment
hard
tax
scene
high
maybe
senior
add
young
agree
food
man
forget
job
yard
success