```
import nltk
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from nltk.corpus import movie_reviews, stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
```

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

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

```
positive_reviews = [
    movie_reviews.raw(fileid)
    for fileid in movie_reviews.fileids('pos')
]

negative_reviews = [
    movie_reviews.raw(fileid)
    for fileid in movie_reviews.fileids('neg')
]

print("Positive reviews:", len(positive_reviews))
print("Negative reviews:", len(negative_reviews))
```

```
Positive reviews: 1000
Negative reviews: 1000
```

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

```
positive_corpus = positive_reviews
negative_corpus = negative_reviews
```

```
tfidf_pos = TfidfVectorizer(
    stop_words=stop_words,
    max_features=1000
)

tfidf_neg = TfidfVectorizer(
    stop_words=stop_words,
    max_features=1000
)

X_pos = tfidf_pos.fit_transform(positive_corpus)
X_neg = tfidf_neg.fit_transform(negative_corpus)
```

```
pos_scores = np.mean(X_pos.toarray(), axis=0)
pos_terms = tfidf_pos.get_feature_names_out()

pos_tfidf = pd.DataFrame({
    'term': pos_terms,
    'score': pos_scores
}).sort_values(by='score', ascending=False).head(15)

pos_tfidf
```

	term	score
312	film	0.099638
563	movie	0.062986
605	one	0.060080
492	like	0.042244
837	story	0.034709
368	good	0.033924
490	life	0.033608
894	time	0.032439
31	also	0.031933
963	well	0.031352
134	character	0.030747
265	even	0.030377
135	characters	0.029870
923	two	0.029204
567	much	0.028666

Next steps: Generate code with pos_tfidf New interactive sheet

```
neg_scores = np.mean(X_neg.toarray(), axis=0)
neg_terms = tfidf_neg.get_feature_names_out()

neg_tfidf = pd.DataFrame({
    'term': neg_terms,
    'score': neg_scores
}).sort_values(by='score', ascending=False).head(15)

neg_tfidf
```

	term	score
302	film	0.094042
559	movie	0.077448
600	one	0.061447
486	like	0.046121
254	even	0.037014
360	good	0.034473
883	time	0.033537
63	bad	0.033469
986	would	0.032474
823	story	0.032097
343	get	0.031457
563	much	0.030865
644	plot	0.029986
133	character	0.029938
134	characters	0.029323

Next steps: Generate code with neg_tfidf New interactive sheet

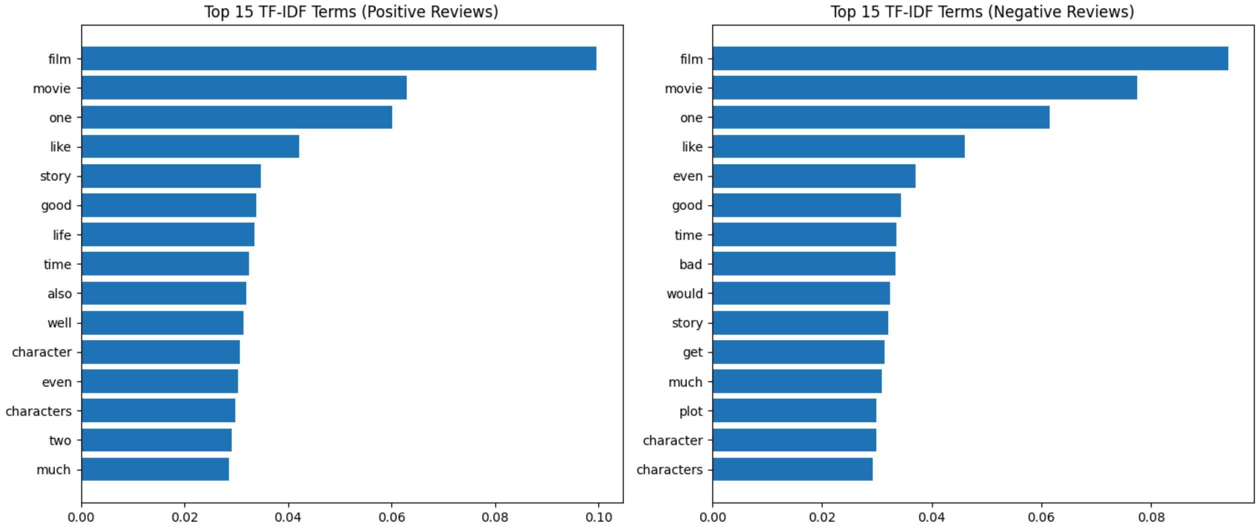
```
plt.figure(figsize=(14,6))

plt.subplot(1,2,1)
plt.barh(pos_tfidf['term'], pos_tfidf['score'])
plt.title("Top 15 TF-IDF Terms (Positive Reviews)")
plt.gca().invert_yaxis()

plt.subplot(1,2,2)
plt.barh(neg_tfidf['term'], neg_tfidf['score'])
plt.title("Top 15 TF-IDF Terms (Negative Reviews)")
```

```
    plt.gca().invert_yaxis()

    plt.tight_layout()
    plt.show()
```


Top 15 TF-IDF Terms (Positive Reviews)
film
movie
one
like
story
good
life
time
also
well
character
even
characters
two
much
Top 15 TF-IDF Terms (Negative Reviews)
film
movie
one
like
even
good
time
bad
would
story
get
much
plot
character
characters