```
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]   Package movie_reviews is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
# Load file IDs
positive_ids = movie_reviews.fileids('pos')
negative_ids = movie_reviews.fileids('neg')

# Load raw text
positive_reviews = [
    movie_reviews.raw(fileid) for fileid in positive_ids
]

negative_reviews = [
    movie_reviews.raw(fileid) for fileid in negative_ids
]

print(f"Positive reviews: {len(positive_reviews)}")
print(f"Negative reviews: {len(negative_reviews)}")
```

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

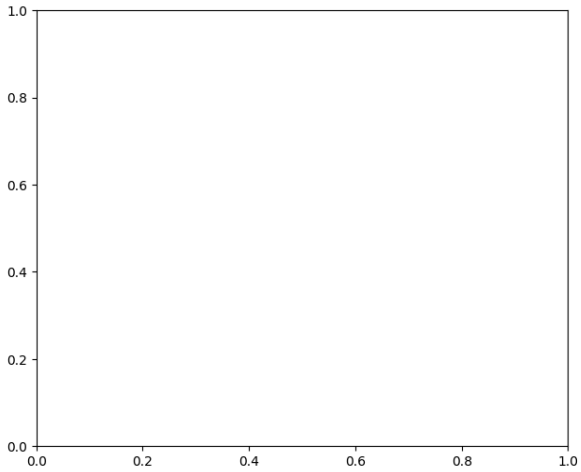
```
stop_words = stopwords.words('english')

vectorizer_pos = TfidfVectorizer(
    stop_words=stop_words,
    max_df=0.9,
    min_df=5
)

vectorizer_neg = TfidfVectorizer(
    stop_words=stop_words,
    max_df=0.9,
    min_df=5
)
```

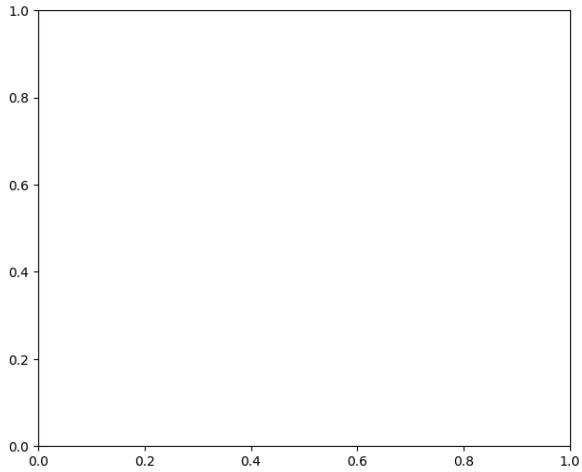
```
# TF-IDF matrices
tfidf_pos = vectorizer_pos.fit_transform(positive_reviews)
tfidf_neg = vectorizer_neg.fit_transform(negative_reviews)
```

```
def get_top_terms(tfidf_matrix, feature_names, top_n=15):
    mean_tfidf = np.asarray(tfidf_matrix.mean(axis=0)).ravel()
    top_indices = mean_tfidf.argsort()[::-1][:top_n]
    return pd.DataFrame({
        'term': feature_names[top_indices],
        'tfidf_score': mean_tfidf[top_indices]
    })
```

```
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# Positive reviews
axes[0].barh(
    top_pos_terms['term'],
    top_pos_terms['tfidf_score'],
    color='green'
)
axes[0].set_title("Top 15 TF-IDF Terms (Positive Reviews)")
axes[0].invert_yaxis()

# Negative reviews
axes[1].barh(
    top_neg_terms['term'],
    top_neg_terms['tfidf_score'],
    color='red'
)
axes[1].set_title("Top 15 TF-IDF Terms (Negative Reviews)")
axes[1].invert_yaxis()

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-3678735816.py in <cell line: 0>()
      3 # Positive reviews
      4 axes[0].barh(
----> 5     top_pos_terms['term'],
      6     top_pos_terms['tfidf_score'],
      7     color='green'

NameError: name 'top_pos_terms' is not defined
```





Next steps: Explain error

Gemini

```
# Get feature names
pos_feature_names = np.array(vectori
neg_feature_names = np.array(vectori

# Get top terms for positive and neg
top_pos_terms = get_top_terms(tfidf_
top_neg_terms = get_top_terms(tfidf_

fig, axes = plt.subplots(1, 2, figsi

# Positive reviews
axes[0].barh(
    top_pos_terms['term'],
```

```
    top_pos_terms['tfidf_score'],
    color='green'
)
axes[0].set_title("Top 15 TF-IDF Ter
axes[0].invert_yaxis()

# Negative reviews
axes[1].barh(
    top_neg_terms['term'],
    top_neg_terms['tfidf_score'],
    color='red'
)
axes[1].set_title("Top 15 TF-IDF Ter
axes[1].invert_yaxis()

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