In [1]:

```python
# Import the necessary libraries
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
```

In [2]:

```python
#Import the preprocessed dataset
data = pd.read_csv('processed_spam_dataset.csv')
```

In [3]:

```python
data.head()
```

Out [3]:

	label	text	tokenized_text	text_features
0	ham	Go until jurong point, crazy.. Available only ...	['go', 'jurong', 'point', 'crazy', 'available'...	{'text_length': 111, 'num_words': 20, 'num_spe...
1	ham	Ok lar... Joking wif u oni...	['ok', 'lar', 'joking', 'wif', 'u', 'oni']	{'text_length': 29, 'num_words': 6, 'num_speci...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	['free', 'entry', 'wkly', 'comp', 'win', 'fa',...	{'text_length': 155, 'num_words': 28, 'num_spe...
3	ham	U dun say so early hor... U c already then say...	['u', 'dun', 'say', 'early', 'hor', 'u', 'c', ...	{'text_length': 49, 'num_words': 11, 'num_spec...
4	ham	Nah I don't think he goes to usf, he lives aro...	['nah', 'dont', 'think', 'goes', 'usf', 'lives...	{'text_length': 61, 'num_words': 13, 'num_spec...

In [4]:

```python
X = data['text']
y = data['label']
```

In [5]:

```python
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, r
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=
```

In [6]:

```python
vectorizer = TfidfVectorizer()
```

In [7]:

```python
X_train_tfidf = vectorizer.fit_transform(X_train)
X_val_tfidf = vectorizer.transform(X_val)
X_test_tfidf = vectorizer.transform(X_test)
```

In [9]:

```python
model = MultinomialNB()
model.fit(X_train_tfidf, y_train)
```

Out [9]:

```
▾ MultinomialNB
MultinomialNB()
```

In [10]:

```python
y_pred_val = model.predict(X_val_tfidf)
```

In [12]:

```
# Evaluate the model on the validation set
accuracy = accuracy_score(y_val, y_pred_val)
report = classification_report(y_val, y_pred_val)

print("Validation Accuracy:", accuracy)
print("Classification Report:\n", report)
```

```
Validation Accuracy: 0.9712918660287081
Classification Report:
               precision    recall  f1-score   support

         ham       0.97      1.00      0.98       729
        spam       1.00      0.78      0.87       107

    accuracy                           0.97       836
   macro avg       0.98      0.89      0.93       836
weighted avg       0.97      0.97      0.97       836
```

In [13]:

```
y_pred_test = model.predict(X_test_tfidf)
```

In [14]:

```
# Evaluate the model on the test set
accuracy_test = accuracy_score(y_test, y_pred_test)
report_test = classification_report(y_test, y_pred_test)

print("Test Accuracy:", accuracy_test)
print("Test Classification Report:\n", report_test)
```

```
Test Accuracy: 0.9485645933014354
Test Classification Report:
               precision    recall  f1-score   support

         ham       0.94      1.00      0.97       724
        spam       1.00      0.62      0.76       112

    accuracy                           0.95       836
   macro avg       0.97      0.81      0.87       836
weighted avg       0.95      0.95      0.94       836
```