

```
In [7]: # Import necessary Libraries
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

In [8]: # Load your dataset (replace 'your_dataset.csv' with your actual dataset)
# The dataset should have columns 'text' for email content and 'label' for spam or not
df = pd.read_csv(r"C:\Users\warul\Downloads\archive (3).zip", encoding='latin1')

# Explore the dataset
print(df.head())
```

	v1	v2	Unnamed: 2	\
0	ham	Go until jurong point, crazy.. Available only ...		NaN
1	ham	Ok lar... Joking wif u oni...		NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...		NaN
3	ham	U dun say so early hor... U c already then say...		NaN
4	ham	Nah I don't think he goes to usf, he lives aro...		NaN

  

	Unnamed: 3	Unnamed: 4
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN

```
In [9]: # Drop unnecessary columns
df = df[['v1', 'v2']]

# Rename columns for clarity
df.columns = ['label', 'message']

# Display basic information about the dataset
print(df.info())

# Display the first few rows of the dataset
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
#   Column   Non-Null Count  Dtype
---  -
0    label    5572 non-null   object
1    message  5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
None
```

	label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

```
In [14]: X = df['message']
        y = df['label']
```

```
In [17]: # Split the dataset into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

```
In [18]: # Feature extraction using CountVectorizer
        vectorizer = CountVectorizer()
        X_train_vectorized = vectorizer.fit_transform(X_train)
        X_test_vectorized = vectorizer.transform(X_test)

        # Train a Naïve Bayes classifier
        classifier = MultinomialNB()
        classifier.fit(X_train_vectorized, y_train)

        # Make predictions on the test set
        y_pred = classifier.predict(X_test_vectorized)

        # Evaluate the model
        accuracy = accuracy_score(y_test, y_pred)
        conf_matrix = confusion_matrix(y_test, y_pred)
        classification_rep = classification_report(y_test, y_pred)

        print(f'Accuracy: {accuracy:.2f}')
        print('Confusion Matrix:')
        print(conf_matrix)
        print('Classification Report:')
        print(classification_rep)
```

Accuracy: 0.98

Confusion Matrix:

```
[[963  2]
 [ 16 134]]
```

Classification Report:

	precision	recall	f1-score	support
ham	0.98	1.00	0.99	965
spam	0.99	0.89	0.94	150
accuracy			0.98	1115
macro avg	0.98	0.95	0.96	1115
weighted avg	0.98	0.98	0.98	1115

```
In [ ]:
```