SMS CLASSIFIER

Develop a text classification model to classify SMS as either spam or non-spam using data science techniques in Python.

In [1]:
```
#libraries imported
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
```

In [2]:
```
#reading data from dataset
df = pd.read_csv('SMSSpamCollection.csv', sep='\t', names=['label', 'message'])
```

These sets can be used to train and evaluate a machine learning model, such as a classifier, to predict whether a message is 'ham' (0) or 'spam' (1) based on its content.

In [3]:
```
# Data Preprocessing
#'label' column,It assigns the value 0 to 'ham' and 1 to 'spam'.
#This is often done when you want to convert categorical labels into numerical
#where numerical inputs are usually preferred.
df['label'] = df['label'].map({'ham': 0, 'spam': 1})
X_train, X_test, y_train, y_test = train_test_split(df['message'], df['label'],
```

These matrices can be used as input features for machine learning models, such as classifiers, to train and make predictions based on the token counts of words in the text data.

In [4]:
```
# Text Vectorization
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

In [5]:
```
# Train Model
classifier = MultinomialNB()
classifier.fit(X_train_vectorized, y_train)
```

Out[5]:
```
MultinomialNB()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [6]:
```
# Make Predictions
predictions = classifier.predict(X_test_vectorized)
```

In [7]:
```
# Evaluate Model
accuracy = accuracy_score(y_test, predictions)
report = classification_report(y_test, predictions)
print(f'Accuracy: {accuracy}')
print(f'Classification Report:\n{report}')
```

```
Accuracy: 0.9919282511210762
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       966
           1       1.00      0.94      0.97       149

    accuracy                           0.99      1115
   macro avg       1.00      0.97      0.98      1115
weighted avg       0.99      0.99      0.99      1115
```

machine learning model (e.g., from scikit-learn) capable of predicting spam or non-spam based on the vectorized input.

In [9]:
```
# Take User Input
user_input = input('Enter SMS Message: ')
user_input_vectorized = vectorizer.transform([user_input])
prediction = classifier.predict(user_input_vectorized)

if prediction[0] == 1:
    print(' IT IS A SPAM SMS..!!')
else:
    print('DONT WORRY IT IS NOT A SPAM SMS..!')
```

```
Enter SMS Message: ham  I've been searching for the right words to thank you
for this breather. I promise i wont take your help for granted and will fulfi
l my promise. You have been wonderful and a blessing at all times.
DONT WORRY IT IS NOT A SPAM SMS..!
```

In [ ]:
```
```