SMS CLASSIFIER

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

In [8]:
```
# 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 [9]:
```
# Reading data from dataset
df = pd.read_csv('SMSSpamCollection.csv', sep='\t', names=['label', 'message']
```

In [10]:
```
df.head(5)
```

Out[10]:

	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 [11]:
```
#Date Preprocessing

#Label coloumn, It assigns the value 0 to hum and I to 'spum.

#This is often done when you want to convert categorical Labels into numertcal
# Label column: Assign the value 0 to 'ham' and 1 to 'spam'
df['label'] = df['label'].map({'ham': 0, 'spam': 1})
X_train, X_test, y_train, y_test = train_test_split(df['message'], df['label']
```

In [12]:
```
# Text Vectorization
vectorizer = CountVectorizer()

X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

In [13]:
```
# Train Model
classifier = MultinomialNB()
classifier.fit(X_train_vectorized, y_train)
```

Out[13]:
```
MultinomialNB
MultinomialNB()
```

In [14]:
```
# Make predictions
predictions = classifier.predict(X_test_vectorized)
```

In [15]:
```
# 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: 0.9913854989231874
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      1202
           1       0.98      0.96      0.97       191

    accuracy                           0.99      1393
   macro avg       0.99      0.98      0.98      1393
weighted avg       0.99      0.99      0.99      1393
```

In [17]:
```
# Take User Input
def sms():
    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("DON'T WORRY IT IS NOT A SPAM SMS..!")
```

In [18]:
```
sms()
```

```
Enter SMS Message: Congratulations..you won a bonus
IT IS A SPAM SMS..!
```

In [19]:
```
sms()
```

```
Enter SMS Message: I love dancing
DON'T WORRY IT IS NOT A SPAM SMS..!
```

In [ ]: