In [1]:
```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
import joblib
```

In [ ]:

In [ ]:

Predefined Input

In [ ]:

In [ ]:

In [2]:
```
# Load the new dataset
new_data = pd.read_csv('spam.csv', encoding= 'latin1')
```

In [3]:
```
# Load the saved model
classifier = joblib.load('spam_classifier_model.joblib')
```

In [4]:
```
# Preprocess the new dataset
vectorizer = CountVectorizer()
vectorizer.fit_transform(new_data['v2'])  # Fit the vectorizer with the training data vocabulary
X_new = vectorizer.transform(new_data['v2'])  # Transform the new data
```

In [5]:
```
# Use the loaded model for predictions
y_pred = classifier.predict(X_new)
```

In [6]:
```
print(X_new)
```

```
  (0, 1070)     1
  (0, 1304)     1
  (0, 1749)     1
  (0, 1751)     1
  (0, 2044)     1
  (0, 2321)     1
  (0, 3539)     1
  (0, 3583)     1
  (0, 3623)     1
  (0, 4074)     1
  (0, 4330)     1
  (0, 4456)     1
  (0, 5517)     1
  (0, 5900)     1
  (0, 7622)     1
  (0, 8006)     1
  (0, 8242)     1
  (0, 8463)     1
  (1, 4298)     1
  (1, 4492)     1
  (1, 5484)     1
  (1, 5513)     1
  (1, 8366)     1
  (2, 77)       1
  (2, 402)      1
  :     :
  (5570, 1778)  1
  (5570, 1786)  1
  (5570, 2586)  1
  (5570, 2883)  1
  (5570, 3298)  1
  (5570, 3347)  1
  (5570, 3459)  1
  (5570, 3676)  1
  (5570, 3769)  1
  (5570, 4074)  1
  (5570, 4148)  1
  (5570, 4204)  1
  (5570, 4596)  1
  (5570, 5316)  1
  (5570, 7018)  1
  (5570, 7028)  1
```

```
  (5570, 7604)  1
  (5570, 7733)  1
  (5570, 8041)  1
  (5570, 8288)  1
  (5571, 4211)  2
  (5571, 5226)  1
  (5571, 6483)  1
  (5571, 7733)  1
  (5571, 7862)  1
```

In [7]:
```
print(y_pred)
```

```
['ham' 'ham' 'spam' ... 'ham' 'ham' 'ham']
```

In [ ]:

In [ ]:

User Input

In [8]:
```
# Load the saved model
classifier = joblib.load('spam_classifier_model.joblib')
```

In [9]:
```
# Load the vocabulary used by the CountVectorizer
vectorizer = joblib.load('count_vectorizer.joblib')
```

In [13]:
```
# Prompt the user to enter an email text
user_input = input("Enter the email text: ")
```

```
Enter the email text: Yo come over carlos will be here soon
```

In [14]:
```
# Preprocess the user input
X_input = vectorizer.transform([user_input])

# Use the loaded model for prediction
prediction = classifier.predict(X_input)[0]
```

In [15]:
```
# Print the prediction
if prediction == 'spam':
```

```
    print("The email is classified as spam.")
else:
    print("The email is not spam.")
```

```
The email is not spam.
```

In [ ]:

In [ ]: