

# Building the core components of the spam classifier(using Kaggle dataset).

## Program:

```
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, classification_report, confusion_matrix

# Step 1: Data Preprocessing
df = pd.read_csv('spam_dataset.csv') # Replace 'spam_dataset.csv' with the actual dataset filename
# Perform any necessary data cleaning or preprocessing steps (e.g., handling missing values)

# Step 2: Data Split
X = df['message']
y = df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 3: Feature Extraction
vectorizer = CountVectorizer()
X_train_count = vectorizer.fit_transform(X_train)
X_test_count = vectorizer.transform(X_test)

# Step 4: Model Building
model = MultinomialNB()
model.fit(X_train_count, y_train)

# Step 5: Model Evaluation
y_pred = model.predict(X_test_count)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

# Step 6: Model Deployment
# Use the trained model to make predictions on new, unseen SMS messages
```

```
new_messages = ['Free offer!', 'Let\'s meet for lunch tomorrow', 'Congratulations, you won a prize!']  
new_messages_count = vectorizer.transform(new_messages)  
new_messages_pred = model.predict(new_messages_count)  
print(new_messages_pred)
```