SPAM-CLASSIFICATION DEVELOPMENT PART 2

```
import nltk
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Download NLTK data (if not already downloaded)
nltk.download('stopwords')
nltk.download('punkt')

# Load and prepare your dataset (ensure you have labeled data with 'text' and 'label' columns)
# Example dataset:
# data = pd.read_csv('spam_dataset.csv')
# X = data['text']
# y = data['label']

# Tokenization and feature extraction
count_vectorizer = CountVectorizer()
tfidf_transformer = TfidfTransformer()

X_counts = count_vectorizer.fit_transform(X)
X_tfidf = tfidf_transformer.fit_transform(X_counts)
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2, random_state=42)

# Train a Naive Bayes classifier
clf = MultinomialNB()
clf.fit(X_train, y_train)

# Make predictions
y_pred = clf.predict(X_test)

# Evaluate the classifier
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")
print("Confusion Matrix:\n", confusion)
print("Classification Report:\n", report)
```