OASIS INFOBYTE INTERNSHIP (Data Science)

Task 4: Email Spam Detection with Machine Learning

Name: Palak Parmar
College: Bansal Group Of Institute
Submission Date: 07/11/2025

Introduction:
This beginner-level project demonstrates how to detect spam and non-spam (ham) emails using a simple Machine Learning model. We use sample text data and apply a Naive Bayes classifier for text classification. This code is kept simple and self-contained, requiring no external CSV files.

Libraries to install (run once in terminal):

```
pip install pandas
pip install scikit-learn
```

Code (copy and run in Python):

```
# Beginner-friendly Email Spam Detection (no external files required)

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, confusion_matrix, classification_report

# Sample data
data = {
    'text': [
        'Win a free iPhone now',
        'Lowest price for car insurance',
        'Hey, are we still meeting today?',
        'Your OTP code is 123456',
        'Congratulations! You have won a lottery',
        'Let’s catch up for lunch tomorrow',
        'Claim your free gift voucher now',
        'Reminder: project meeting at 4 PM',
        'Get cheap loans easily',
        'See you soon my friend'
    ],
    'label': ['spam','spam','ham','ham','spam','ham','spam','ham','spam','ham']
}

df = pd.DataFrame(data)

# Split data
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.3, random_state=42)
```

```
# Convert text to numerical vectors
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Train Naive Bayes model
model = MultinomialNB()
model.fit(X_train_vec, y_train)

# Predictions
y_pred = model.predict(X_test_vec)

# Evaluate
print("Accuracy:", round(accuracy_score(y_test, y_pred)*100, 2), "%")
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Expected Output (example): - Accuracy: Around 80–100% - Confusion Matrix: Shows correct and incorrect predictions - Classification Report: Displays precision, recall, and F1-score This code uses a small internal dataset, so it runs easily for beginners without file handling issues.

Conclusion: This project demonstrates how to classify emails as spam or ham using the Naive Bayes algorithm. It provides a complete beginner-level implementation with text data processing, model training, and evaluation.