

AI Lab\Exp_12\spamDetection.py

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.feature_extraction.text import CountVectorizer
4 from sklearn.naive_bayes import MultinomialNB
5 from sklearn.metrics import accuracy_score, confusion_matrix
6
7 # 1. Load dataset from URL [cite: 59, 60]
8 # The manual uses the SMS Spam Collection dataset from a specific GitHub URL [cite:
9 url = "https://raw.githubusercontent.com/justmarkham/pycon-2016-
10 tutorial/master/data/sms.tsv"
11 data = pd.read_csv(url, sep='\t', names=['label', 'message'])
12
13 # 2. Encode labels: ham -> 0, spam -> 1 [cite: 62, 63]
14 data['label_num'] = data.label.map({'ham': 0, 'spam': 1})
15
16 # 3. Split the dataset [cite: 64]
17 # Splitting into training and testing sets (80% train, 20% test) [cite: 65]
18 X_train, X_test, y_train, y_test = train_test_split(
19     data['message'],
20     data['label_num'],
21     test_size=0.2,
22     random_state=42
23 )
24
25 # 4. Convert text messages into feature vectors
26 vectorizer = CountVectorizer()
27 X_train_counts = vectorizer.fit_transform(X_train)
28 X_test_counts = vectorizer.transform(X_test)
29
30 # 5. Train the Naïve Bayes classifier [cite: 68]
31 clf = MultinomialNB() # [cite: 70]
32 clf.fit(X_train_counts, y_train) # [cite: 71]
33
34 # 6. Make predictions on test data [cite: 72]
35 y_pred = clf.predict(X_test_counts) # [cite: 74]
36
37 # 7. Evaluate performance [cite: 75]
38 print("Accuracy:", accuracy_score(y_test, y_pred)) # [cite: 76]
39 print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred)) # [cite: 77]
```