Spam_Detection

February 3, 2025

[4]:
```
import pandas as pd
import numpy as np
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report
```

[5]:
```
df=pd.read_csv("/home/comp56/spam.csv",encoding='ISO-8859-1')
```

[6]:
```
def clean_text(text):
    text = text.lower()  # Convert text to lowercase
    text = re.sub(r'\W', ' ', text)  # Remove non-word characters
    text = re.sub(r'\s+', ' ', text)  # Remove extra spaces
    return text
```

[7]:
```
df['cleaned_text'] = df['v2'].apply(clean_text)  # Assuming 'v2' column␣
 ↪contains email content
```

[8]:
```
X = df['cleaned_text']
y = df['v1'].apply(lambda x: 1 if x == 'spam' else 0)  # Convert labels to␣
 ↪binary (1 for spam, 0 for ham)
```

[9]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)
```

[10]:
```
vectorizer = TfidfVectorizer(max_features=5000)  # You can adjust the number of␣
 ↪features
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

[11]:
```
model = MultinomialNB()
model.fit(X_train_tfidf, y_train)
```

[11]:
```
MultinomialNB()
```

[12]:
```
y_pred = model.predict(X_test_tfidf)
```

[13]:
```
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
```

[14]:
```
print(f"Accuracy: {accuracy*100:.2f}%")
print("Confusion Matrix:")
print(conf_matrix)
print("Classification Report:")
print(class_report)
```

```
Accuracy: 96.47%
Confusion Matrix:
[[1453    0]
 [  59  160]]
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1453
           1       1.00      0.73      0.84       219

    accuracy                           0.96      1672
   macro avg       0.98      0.87      0.91      1672
weighted avg       0.97      0.96      0.96      1672
```

[15]:
```
import joblib
```

[18]:
```
# Save model and vectorizer
joblib.dump(model, 'spam_model.pkl')
joblib.dump(vectorizer, 'vectorizer.pkl')
```

[18]:
```
['vectorizer.pkl']
```

[ ]:
```
```