

Spam

January 27, 2024

```
[11]: # Import necessary libraries
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
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[12]: # Load dataset
df = pd.read_csv('mail_data.csv')
```

```
[13]: df.head()
```

```
[13]:
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

```
[15]: # Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['Message'], \
    df['Category'], test_size=0.2, random_state=42)
```

```
[16]: # Vectorize the text data
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

```
[17]: # Train a Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train_vectorized, y_train)
```

```
[17]: MultinomialNB()
```

```
[18]: # Make predictions on the test set
predictions = classifier.predict(X_test_vectorized)
```

```
[19]: # Evaluate the model
accuracy = accuracy_score(y_test, predictions)
conf_matrix = confusion_matrix(y_test, predictions)
classification_rep = classification_report(y_test, predictions)
```

```
[20]: # Display the results
print(f'Accuracy: {accuracy:.2f}')
print('\nConfusion Matrix:')
print(conf_matrix)
print('\nClassification Report:')
print(classification_rep)
```

Accuracy: 0.99

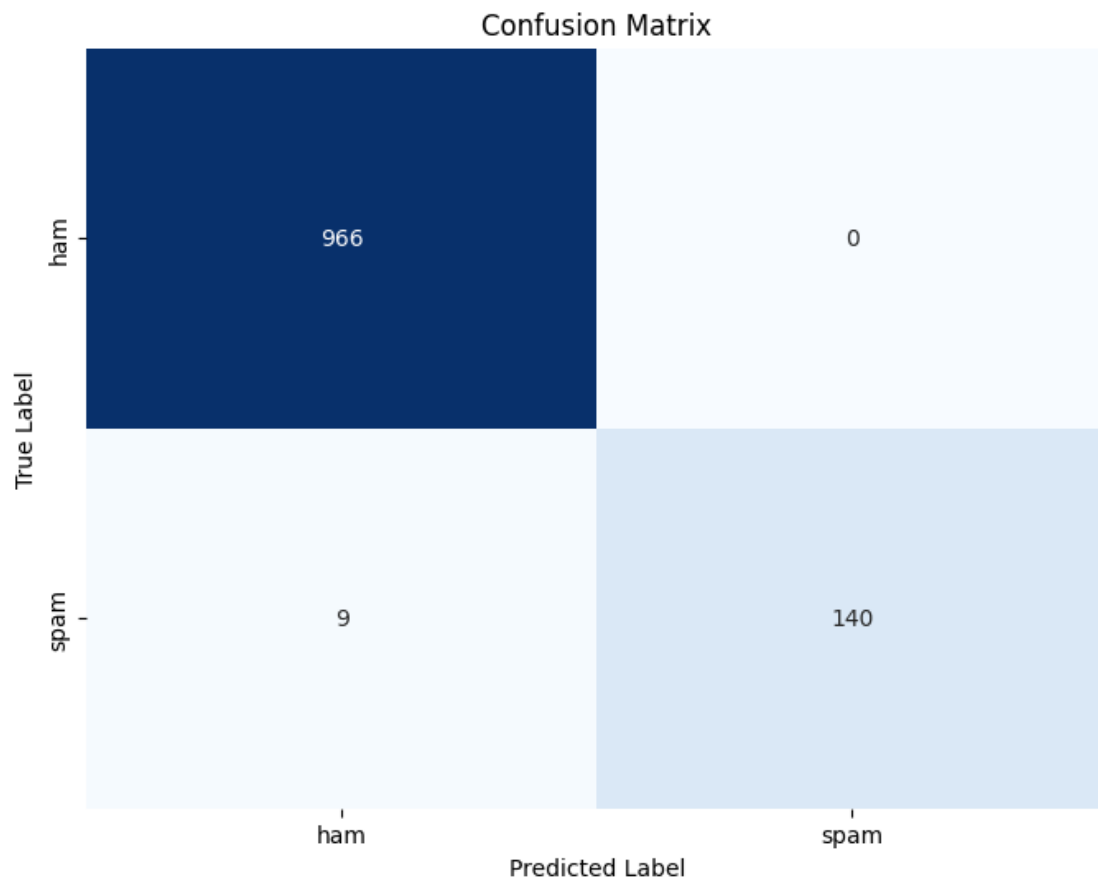
Confusion Matrix:

```
[[966  0]
 [  9 140]]
```

Classification Report:

	precision	recall	f1-score	support
ham	0.99	1.00	1.00	966
spam	1.00	0.94	0.97	149
accuracy			0.99	1115
macro avg	1.00	0.97	0.98	1115
weighted avg	0.99	0.99	0.99	1115

```
[21]: # Plotting Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=classifier.classes_, yticklabels=classifier.classes_)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```



[]: