In [53]:
```
# Import necessary libraries
import pandas as pd
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, classification_report
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

In [42]:
```
# data

df  = pd.read_csv('/Users/user/Downloads/spam_not_spam.csv')
df.shape
```

Out[42]:
```
(5572, 2)
```

In [43]:
```
df.head()
```

Out[43]:

	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...

In [44]:
```
# Split 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
```

In [45]:
```
# Vectorize the text data
vectorizer = TfidfVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

In [46]:

```
# Train the Naive Bayes model
model = MultinomialNB()
model.fit(X_train_vectorized, y_train)
```

Out[46]:

```
MultinomialNB()
```

In [47]:

```
# Make predictions on test data
y_pred = model.predict(X_test_vectorized)
```

In [48]:

```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", classification_report(y_test, y_pred))
```

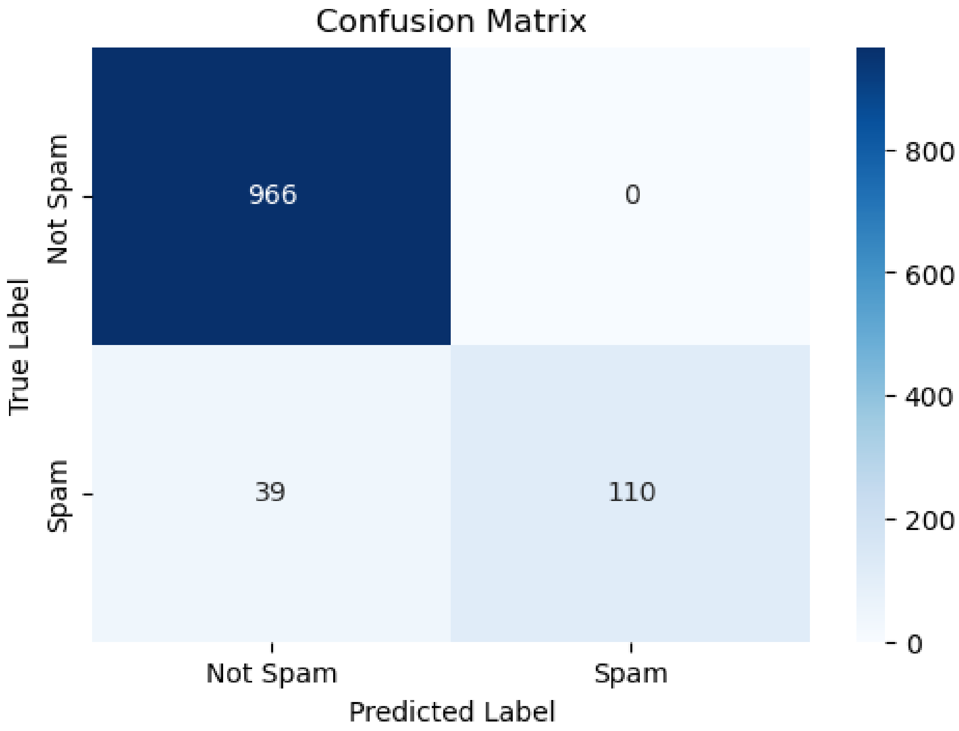
```
Accuracy: 0.9650224215246637
Classification Report:
               precision    recall  f1-score   support

         ham       0.96      1.00      0.98       966
        spam       1.00      0.74      0.85       149

    accuracy                           0.97      1115
   macro avg       0.98      0.87      0.91      1115
weighted avg       0.97      0.97      0.96      1115
```

In [52]:

```
# Generate and print the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)
```

```
Confusion Matrix:
 [[966   0]
 [ 39 110]]
```

```
In [54]: # Plotting the confusion matrix
         plt.figure(figsize=(6,4))
         sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=['Not Spam', 'Spam'], yticklabels=['Not S
         plt.xlabel("Predicted Label")
         plt.ylabel("True Label")
         plt.title("Confusion Matrix")
         plt.show()
```



```
In [ ]:
```