In [1]:
```
import pandas as pd
import numpy as np

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, confusion_mat

import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data = pd.read_csv(
    "https://raw.githubusercontent.com/justmarkham/pycon-2016-tutorial/master/da
    sep="\t",
    names=["label", "message"]
)

data.head()
```

Out[3]:

	label	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 [4]:
```
data['label_num'] = data.label.map({'ham': 0, 'spam': 1})
data.head()
```

Out[4]:

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

In [5]:
```
X = data['message']
y = data['label_num']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)
```

In [6]:
```
vectorizer = TfidfVectorizer(stop_words='english')

X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

In [7]:
```
model = MultinomialNB()
model.fit(X_train_tfidf, y_train)
```

Out[7]:
```
MultinomialNB()
```

In [8]:
```
y_pred = model.predict(X_test_tfidf)
```

In [9]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9741564967695621
```

In [10]:
```
print(classification_report(y_test, y_pred))
```

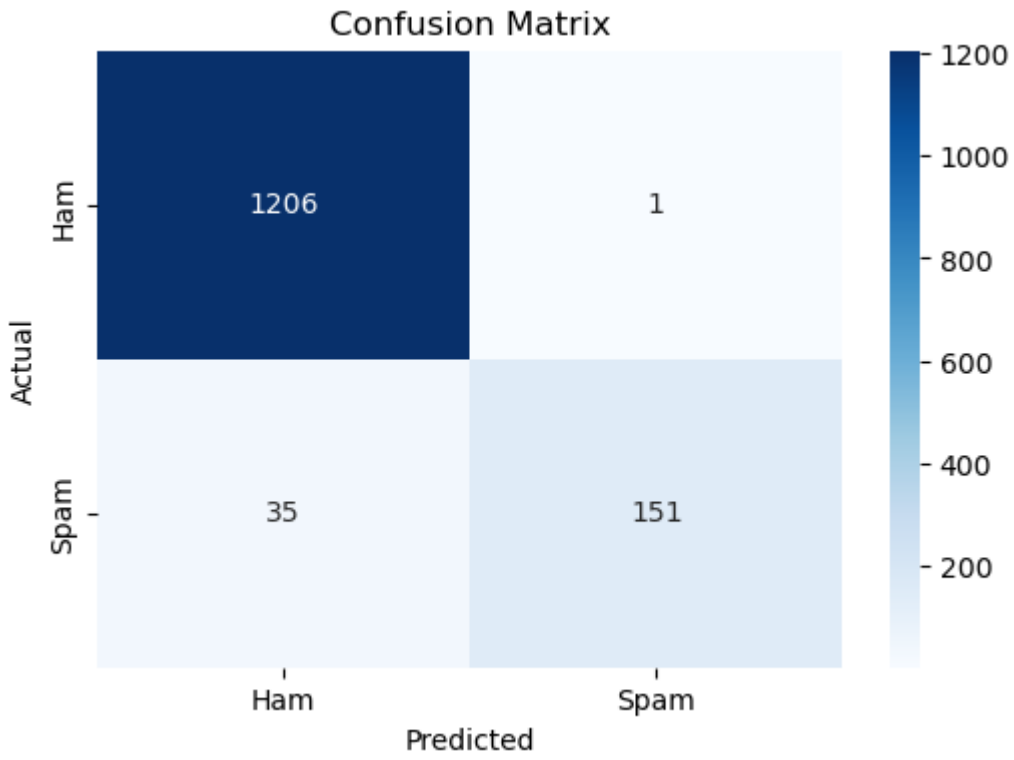
```
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      1207
           1       0.99      0.81      0.89       186

    accuracy                           0.97      1393
   macro avg       0.98      0.91      0.94      1393
weighted avg       0.97      0.97      0.97      1393
```

In [11]:
```
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Ham", "Spam"],
            yticklabels=["Ham", "Spam"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```



In [12]:
```
sample_emails = [
    "Congratulations! You have won a free gift card",
    "Are we still meeting tomorrow for the project discussion?"
]

sample_tfidf = vectorizer.transform(sample_emails)
predictions = model.predict(sample_tfidf)

for email, pred in zip(sample_emails, predictions):
    print(f"Email: {email}")
    print("Prediction:", "Spam" if pred == 1 else "Not Spam")
    print()
```

```
Email: Congratulations! You have won a free gift card
Prediction: Spam

Email: Are we still meeting tomorrow for the project discussion?
Prediction: Not Spam
```

In [ ]: