import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.model\_selection import train\_test\_split  
from sklearn.feature\_extraction.text import TfidfVectorizer  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import accuracy\_score, confusion\_matrix, precision\_score, recall\_score, f1\_score, classification\_report

data = pd.read\_csv('../data/spam.csv', encoding="ISO-8859-1")  
data.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'], axis=1, inplace=True)  
data.drop\_duplicates(inplace=True)  
  
data['v1'].replace(['ham', 'spam'], [0, 1], inplace=True)  
  
data.set\_axis(['Category', 'Message'], axis=1, inplace=True)  
  
data

/var/folders/15/hdxx982s3fj3bxc5v64cbpd00000gn/T/ipykernel\_6904/1066889766.py:7: FutureWarning: DataFrame.set\_axis 'inplace' keyword is deprecated and will be removed in a future version. Use `obj = obj.set\_axis(..., copy=False)` instead  
 data.set\_axis(['Category', 'Message'], axis=1, inplace=True)

Category Message  
0 0 Go until jurong point, crazy.. Available only ...  
1 0 Ok lar... Joking wif u oni...  
2 1 Free entry in 2 a wkly comp to win FA Cup fina...  
3 0 U dun say so early hor... U c already then say...  
4 0 Nah I don't think he goes to usf, he lives aro...  
... ... ...  
5567 1 This is the 2nd time we have tried 2 contact u...  
5568 0 Will Ì\_ b going to esplanade fr home?  
5569 0 Pity, \* was in mood for that. So...any other s...  
5570 0 The guy did some bitching but I acted like i'd...  
5571 0 Rofl. Its true to its name  
  
[5169 rows x 2 columns]

vectorizer = TfidfVectorizer()  
  
X = vectorizer.fit\_transform(data['Message'])  
y = data['Category']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

model = LogisticRegression()  
model.fit(X\_train, y\_train)

LogisticRegression()

y\_pred = model.predict(X\_test)  
  
accuracy = accuracy\_score(y\_test, y\_pred) \* 100  
precision = precision\_score(y\_test, y\_pred) \* 100  
recall = recall\_score(y\_test, y\_pred) \* 100  
f1 = f1\_score(y\_test, y\_pred) \* 100  
  
print(f"Accuracy: {accuracy:.2f}%")  
print(f"Precision: {precision:.2f}")  
print(f"Recall: {recall:.2f}")  
print(f"F1-score: {f1:.2f}")  
  
print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

Accuracy: 95.29%  
Precision: 98.04  
Recall: 68.18  
F1-score: 80.43  
  
Classification Report:  
 precision recall f1-score support  
  
 0 0.95 1.00 0.97 1331  
 1 0.98 0.68 0.80 220  
  
 accuracy 0.95 1551  
 macro avg 0.97 0.84 0.89 1551  
weighted avg 0.95 0.95 0.95 1551

plt.figure(figsize=(8, 6))  
  
cm = confusion\_matrix(y\_test, y\_pred)  
sns.heatmap(cm, annot=True, xticklabels=['Spam', 'Not Spam'], yticklabels=['Spam', 'Not Spam'])  
  
plt.xlabel('Predicted')  
plt.ylabel('Actual')  
plt.title('Confusion Matrix')  
plt.show()

