

AICS4

```
pip install nltk
```

```
import pandas as pd
import string
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
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
```

```
# Download stopwords if needed
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()
```

```
# Load dataset
df = pd.read_csv('spam.csv')
df.columns = ['label', 'message']
df['label'] = df['label'].map({'ham': 0, 'spam': 1})
```

```
def preprocess(text):
    text = text.lower()
    text = text.translate(str.maketrans("", "", string.punctuation))
    words = text.split()
    filtered = [stemmer.stem(word) for word in words if word not in stop_words]
    return " ".join(filtered)
```

```
df['cleaned_message'] = df['message'].apply(preprocess)
```

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['cleaned_message'])
y = df['label']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
model = MultinomialNB()
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
def predict_spam(message):
    message_clean = preprocess(message)
    msg_vec = vectorizer.transform([message_clean])
    prediction = model.predict(msg_vec)
    return "spam" if prediction[0] == 1 else "ham"

print(predict_spam("Claim your free gift now!"))
print(predict_spam("I'll see you at the meeting tomorrow."))
```