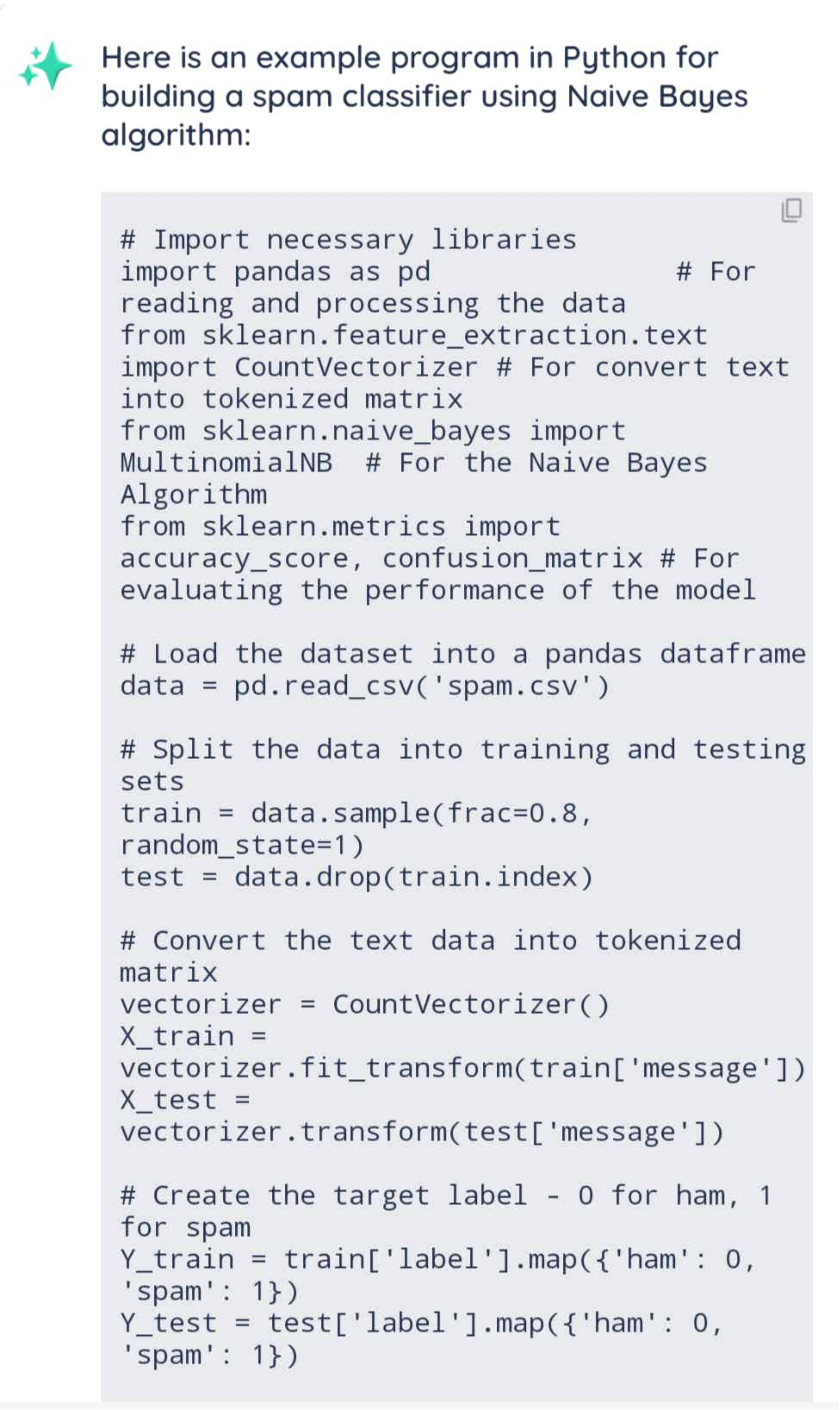


Here is an example program in Python for building a spam classifier using Naive Bayes algorithm:

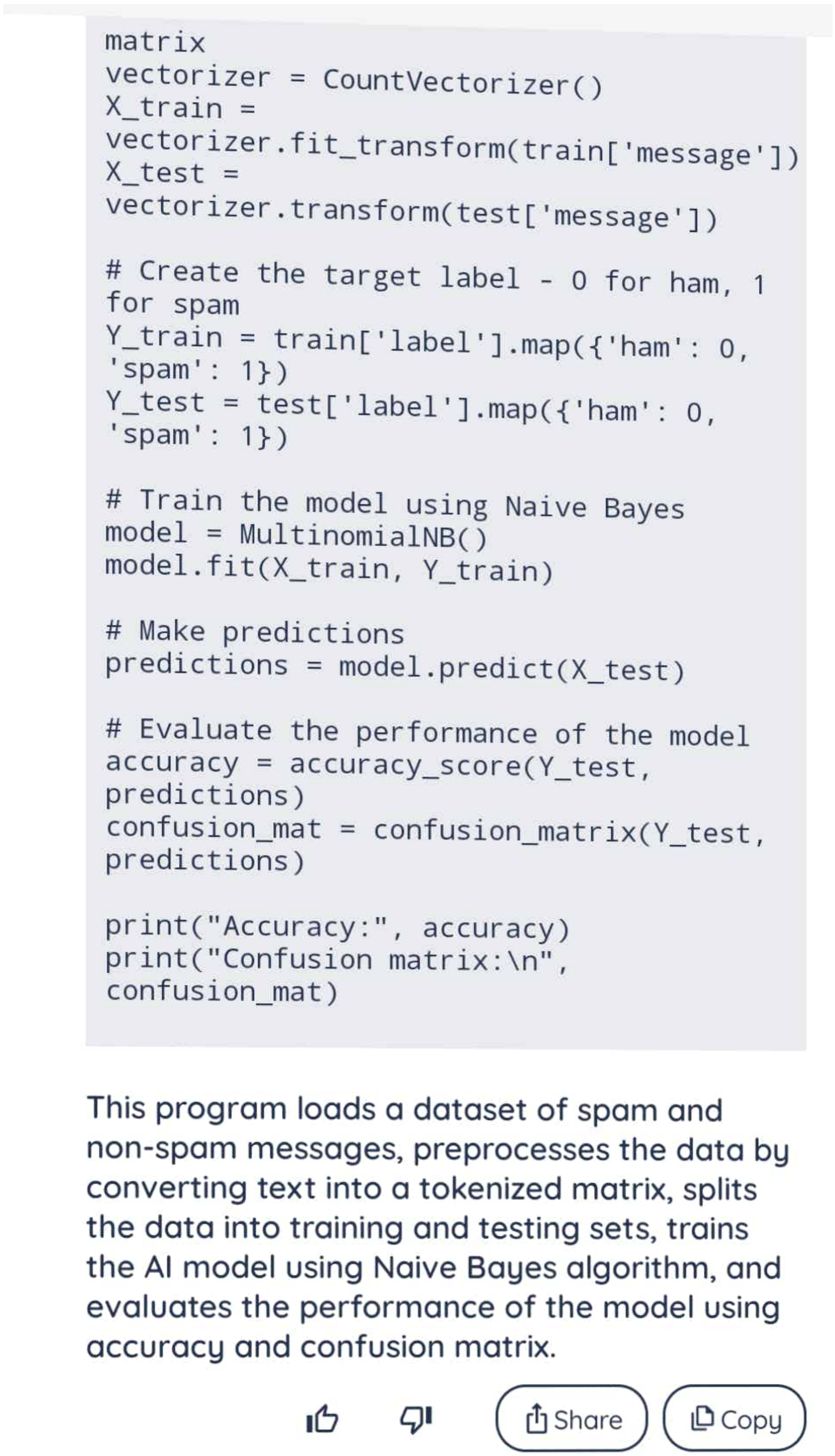
```
# Import necessary libraries
import pandas as pd                # For
reading and processing the data
from sklearn.feature_extraction.text
import CountVectorizer # For convert text
into tokenized matrix
from sklearn.naive_bayes import
MultinomialNB  # For the Naive Bayes
Algorithm
from sklearn.metrics import
accuracy_score, confusion_matrix # For
evaluating the performance of the model

# Load the dataset into a pandas dataframe
data = pd.read_csv('spam.csv')

# Split the data into training and testing
sets
train = data.sample(frac=0.8,
random_state=1)
test = data.drop(train.index)

# Convert the text data into tokenized
matrix
vectorizer = CountVectorizer()
X_train =
vectorizer.fit_transform(train['message'])
X_test =
vectorizer.transform(test['message'])

# Create the target label - 0 for ham, 1
for spam
Y_train = train['label'].map({'ham': 0,
'spam': 1})
Y_test = test['label'].map({'ham': 0,
'spam': 1})
```



```
matrix
vectorizer = CountVectorizer()
X_train =
vectorizer.fit_transform(train['message'])
X_test =
vectorizer.transform(test['message'])

# Create the target label - 0 for ham, 1
for spam
Y_train = train['label'].map({'ham': 0,
'spam': 1})
Y_test = test['label'].map({'ham': 0,
'spam': 1})

# Train the model using Naive Bayes
model = MultinomialNB()
model.fit(X_train, Y_train)

# Make predictions
predictions = model.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(Y_test,
predictions)
confusion_mat = confusion_matrix(Y_test,
predictions)

print("Accuracy:", accuracy)
print("Confusion matrix:\n",
confusion_mat)
```

This program loads a dataset of spam and non-spam messages, preprocesses the data by converting text into a tokenized matrix, splits the data into training and testing sets, trains the AI model using Naive Bayes algorithm, and evaluates the performance of the model using accuracy and confusion matrix.