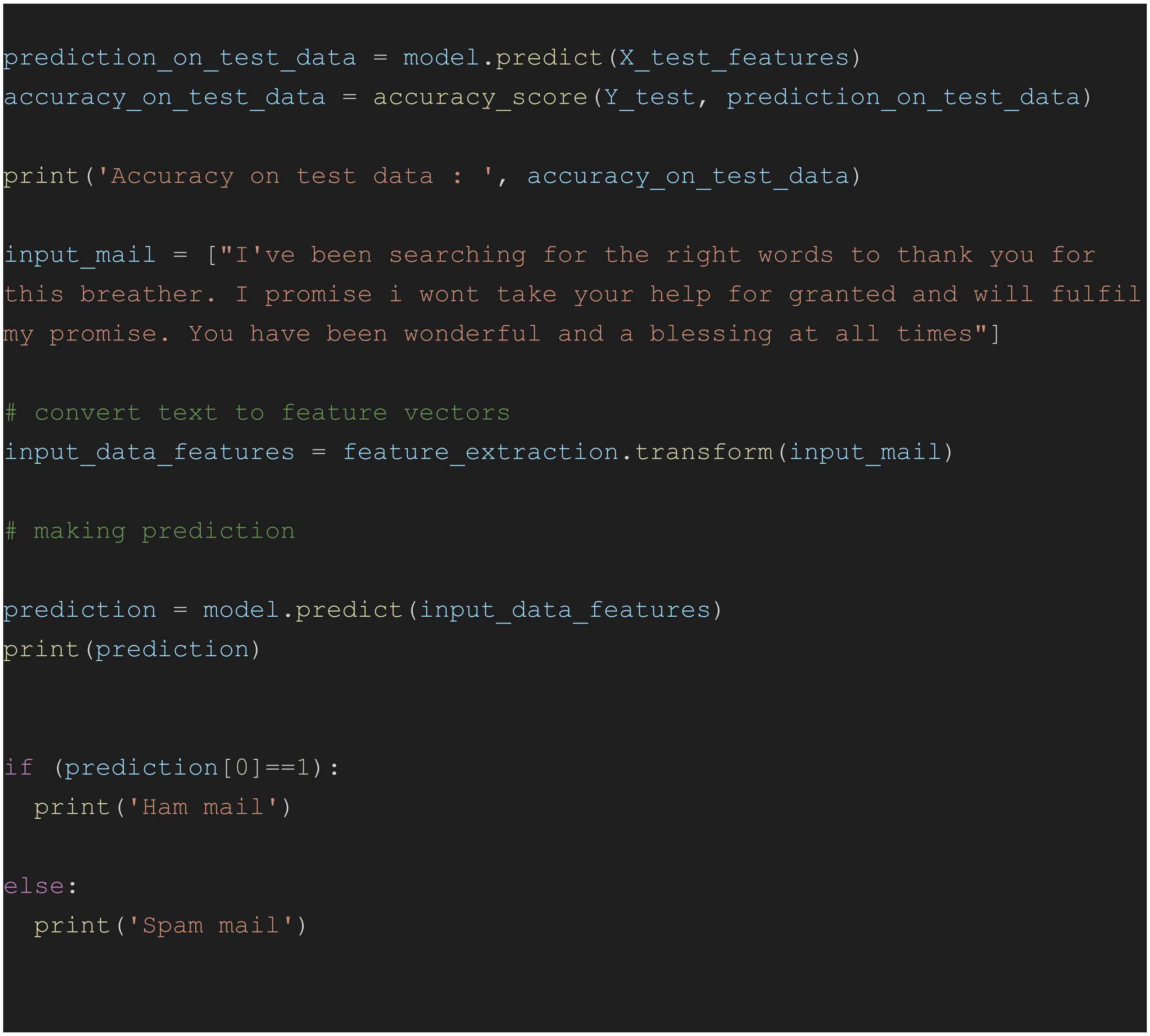
Source Code:

```
import numpy as np
import pandas as pd
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


# loading the data from csv file to a pandas Dataframe
raw_mail_data = pd.read_csv('C:/Users/DELL/Downloads/mail_data.csv')

print(raw_mail_data)

# replace the null values with a null string
mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)),'')

# printing the first 5 rows of the dataframe
mail_data.head()

# checking the number of rows and columns in the dataframe
mail_data.shape

# label spam mail as 0;  ham mail as 1;

mail_data.loc[mail_data['Category'] == 'spam', 'Category',] = 0
mail_data.loc[mail_data['Category'] == 'ham', 'Category',] = 1

# separating the data as texts and label

X = mail_data['Message']

Y = mail_data['Category']

print(X)

print(Y)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=3)

print(X.shape)
print(X_train.shape)
print(X_test.shape)

# transform the text data to feature vectors that can be used as input to
the Logistic regression

feature_extraction = TfidfVectorizer(min_df = 1, stop_words='english',
lowercase=True)

feature_extraction.fit(X_test)
X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)

# convert Y_train and Y_test values as integers

Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')

print(X_train)

print(X_train_features)

model = LogisticRegression()

# training the Logistic Regression model with the training data
model.fit(X_train_features, Y_train)

# prediction on training data

prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train,
prediction_on_training_data)

print('Accuracy on training data : ', accuracy_on_training_data)

# prediction on test data
```



```
prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

print('Accuracy on test data : ', accuracy_on_test_data)

input_mail = ["I've been searching for the right words to thank you for
this breather. I promise i wont take your help for granted and will fulfil
my promise. You have been wonderful and a blessing at all times"]

# convert text to feature vectors
input_data_features = feature_extraction.transform(input_mail)

# making prediction

prediction = model.predict(input_data_features)
print(prediction)


if (prediction[0]==1):
  print('Ham mail')

else:
  print('Spam mail')
```