

```
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('C:/Users/HP/Downloads/mail_data.csv')
print(df)

data = df.where(pd.notnull(df), '')

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

X = data['Message']
Y = 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_test.shape)
print(X_train.shape)

print(Y.shape)
print(Y_train.shape)
print(Y_test.shape)

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

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

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

```

Y_test = Y_test.astype('int')

print(X_train_features)

model=LogisticRegression()
model.fit(X_train_features,Y_train)
predicted_data=model.predict(X_train_features)
Accuracy_of_data=accuracy_score(Y_train,predicted_data)
print("Accuracy:",Accuracy_of_data)

predicted_test=model.predict(X_test_features)
Accuracy_of_test=accuracy_score(Y_test,predicted_test)

print("Accuracy of test:",Accuracy_of_test)
input_your_mail=[" I'm 17 sharing a room with my 15-year-old brother. What can I
do?"]
input_data_feature=feature_extraction.transform(input_your_mail)
prediction=model.predict(input_data_feature)
if(prediction[0]==1):
    print("Ham mail")

else:
    print("Spam mail")

plt.figure(figsize=(6,4))
sns.countplot(x='Category', data=data)
plt.title("Distribution of Ham and Spam")
plt.xlabel("Category (1 = Ham, 0 = Spam)")
plt.ylabel("Count")
plt.show()

accuracies = [Accuracy_of_data, Accuracy_of_test]
labels = ['Train Accuracy', 'Test Accuracy']

plt.figure(figsize=(5,4))
plt.bar(labels, accuracies, color=['skyblue', 'orange'])
plt.title('Model Accuracy')
plt.ylim(0.8, 1.0)
for i, acc in enumerate(accuracies):
    plt.text(i, acc + 0.01, f"{acc:.2f}", ha='center')
plt.show()

```