

TASK 3:EMAIL SPAM DETECTION WITH ML

- **Program:**

```
# Import necessary libraries

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix


# Load the dataset

file_path = 'C:\\Users\\krith\\Downloads\\Krisha \\spam.csv'

dataset = pd.read_csv(file_path, encoding='ISO-8859-1')


# Clean the dataset

dataset_cleaned = dataset.drop(columns=['Unnamed: 2', 'Unnamed: 3',
'Unnamed: 4'])

dataset_cleaned = dataset_cleaned.rename(columns={'v1': 'label', 'v2':
'message'})


# Convert labels to numerical format

label_encoder = LabelEncoder()

dataset_cleaned['label'] =
label_encoder.fit_transform(dataset_cleaned['label']) # 0 for ham, 1 for spam


# Split the data into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(dataset_cleaned['message'],
dataset_cleaned['label'], test_size=0.2, random_state=42)
```

```
# Convert text data to numerical data using CountVectorizer
```

```
vectorizer = CountVectorizer(stop_words='english')
```

```
X_train_vectorized = vectorizer.fit_transform(X_train)
```

```
X_test_vectorized = vectorizer.transform(X_test)
```

```
# Train the Naive Bayes model
```

```
model = MultinomialNB()
```

```
model.fit(X_train_vectorized, y_train)
```

```
# Make predictions on the test data
```

```
y_pred = model.predict(X_test_vectorized)
```

```
# Evaluate the model
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print(f"Accuracy: {accuracy * 100:.2f}%")
```

```
# Detailed classification report
```

```
print("\nClassification Report:")
```

```
print(classification_report(y_test, y_pred, target_names=['Ham', 'Spam']))
```

```
# Confusion matrix
```

```
print("\nConfusion Matrix:")
```

```
print(confusion_matrix(y_test, y_pred))
```

- **Output:**

Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\krith\untitled0.py

```
1 # Import necessary libraries
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import LabelEncoder
5 from sklearn.feature_extraction.text import CountVectorizer
6 from sklearn.naive_bayes import MultinomialNB
7 from sklearn.metrics import accuracy_score, classification_report
8
9 # Load the dataset
10 file_path = 'C:\\Users\\krith\\Downloads\\Kri
11 dataset = pd.read_csv(file_path, encoding='I
12
13 # Clean the dataset
14 dataset_cleaned = dataset.drop(columns=['unn
15 dataset_cleaned = dataset_cleaned.rename(col
16
17 # Convert labels to numerical format
18 label_encoder = LabelEncoder()
19 dataset_cleaned['label'] = label_encoder.fit_
20
21 # Split the data into training and testing sets
22 X_train, X_test, y_train, y_test = train_test
23
24 # Convert text data to numerical data using C
25 vectorizer = CountVectorizer(stop_words='eng
26 X_train_vectorized = vectorizer.fit_transform
27 X_test_vectorized = vectorizer.transform(X_te
28
29 # Train the Naive Bayes model
30 model = MultinomialNB()
31 model.fit(X_train_vectorized, y_train)
32
33 # Make predictions on the test data
34 y_pred = model.predict(X_test_vectorized)
35
36 # Evaluate the model
37 accuracy = accuracy_score(y_test, y_pred)
38 print(f'Accuracy: {accuracy * 100:.2f}%')
39
40 # Detailed classification report
41 print("\nClassification Report:")
42 print(classification_report(y_test, y_pred, 1
```

Console v1.0

Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

[Python 8.15.0 -- An enhanced Interactive Python.]

In [1]: runfile('C:/Users/krith/untitled0.py', wdir='C:/Users/krith')
Accuracy: 98.39%

Classification Report:

	precision	recall	f1-score	support
Ham	0.99	0.99	0.99	965
Spam	0.96	0.92	0.94	150
accuracy			0.98	1115
macro avg	0.97	0.96	0.96	1115
weighted avg	0.98	0.98	0.98	1115

Confusion Matrix:

```
[[959  6]
 [ 12 138]]
```

In [2]:

Python Console History

conda (Python 3.11.5) | Completions: conda | LSP: Python | Line 8, Col 1 | UTF-8 | CRLF | RW | Mem 80%