### **Step 5: Train-Test Split**

#### **Objective:**

To divide the dataset into two parts:

* **Training set**: Used to train the machine learning model.
* **Testing set**: Used to evaluate the performance of the trained model.

#### **Steps:**

1. **Import necessary functions**: We need the train\_test\_split function from scikit-learn.
2. **Perform the split**: Use train\_test\_split to divide the data.

#### **Explanation of Code:**

**Import the Function**:  
python  
Copy code  
from sklearn.model\_selection import train\_test\_split

**Perform the Split**:  
python  
Copy code  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['message'], data['label'], test\_size=0.2, random\_state=42)

* + **data['message']**: The feature set (emails).
  + **data['label']**: The target set (labels: spam or ham).
  + **test\_size=0.2**: Specifies that 20% of the data should be used as the test set.
  + **random\_state=42**: Ensures reproducibility of the split. Any integer can be used here.

### **Running the Code:**

You can run this code in a Jupyter Notebook, Python script, or an interactive Python environment like IPython.

#### **Option 1: Jupyter Notebook**

1. **Open Jupyter Notebook**:
   * If you have Jupyter Notebook installed, you can start it by running jupyter notebook in your terminal or command prompt.
2. **Create a New Notebook**:
   * Create a new notebook and add the code cells.

**Code Cell 1**:  
python  
Copy code  
import pandas as pd

from sklearn.model\_selection import train\_test\_split

# Load the dataset

data = pd.read\_csv('spam.csv', encoding='latin-1')

data = data[['v1', 'v2']]

data.columns = ['label', 'message']

data['label'] = data['label'].map({'ham': 0, 'spam': 1})

**Code Cell 2**:  
python  
Copy code  
# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['message'], data['label'], test\_size=0.2, random\_state=42)

# Print the shapes to verify

print(f'Training set: {X\_train.shape}, {y\_train.shape}')

print(f'Test set: {X\_test.shape}, {y\_test.shape}')

#### **Option 2: Python Script**

1. **Create a Python Script**:
   * Open your text editor or IDE (e.g., VS Code, PyCharm).
   * Create a new file named spam\_detection.py.

**Add the Code**:  
python  
Copy code  
import pandas as pd

from sklearn.model\_selection import train\_test\_split

# Load the dataset

data = pd.read\_csv('spam.csv', encoding='latin-1')

data = data[['v1', 'v2']]

data.columns = ['label', 'message']

data['label'] = data['label'].map({'ham': 0, 'spam': 1})

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['message'], data['label'], test\_size=0.2, random\_state=42)

# Print the shapes to verify

print(f'Training set: {X\_train.shape}, {y\_train.shape}')

print(f'Test set: {X\_test.shape}, {y\_test.shape}')

1. **Run the Script**:
   * Open your terminal or command prompt.
   * Navigate to the directory where spam\_detection.py is saved.

Run the script using Python:  
bash  
Copy code  
python spam\_detection.py

#### **Option 3: Interactive Python Environment**

1. **Open IPython or Python Interactive Shell**:
   * Open your terminal or command prompt.
   * Type ipython or python to enter the interactive shell.

**Run the Code Step-by-Step**:  
python  
Copy code  
import pandas as pd

from sklearn.model\_selection import train\_test\_split

# Load the dataset

data = pd.read\_csv('spam.csv', encoding='latin-1')

data = data[['v1', 'v2']]

data.columns = ['label', 'message']

data['label'] = data['label'].map({'ham': 0, 'spam': 1})

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['message'], data['label'], test\_size=0.2, random\_state=42)

# Print the shapes to verify

print(f'Training set: {X\_train.shape}, {y\_train.shape}')

print(f'Test set: {X\_test.shape}, {y\_test.shape}')

### **Detailed Breakdown of train\_test\_split**

* **X\_train**: Training set features (80% of the original messages).
* **X\_test**: Test set features (20% of the original messages).
* **y\_train**: Training set labels (80% of the original labels).
* **y\_test**: Test set labels (20% of the original labels).

By running this code, you will have successfully split your dataset into training and testing sets, which is a crucial step in preparing your data for machine learning model training and evaluation.