### **Step 8: Save the Model**

#### **What does saving a model mean?**

Saving a model means preserving the state of a trained machine learning model so that it can be used later without having to retrain it. This is useful for deploying the model in a production environment or for future use.

#### **Steps to Save the Model:**

1. **Import joblib**: joblib is a library that provides tools for saving Python objects, including machine learning models.
2. **Save the trained model to a file**: Use joblib's dump function to write the model to a file.
3. **Load the model from the file**: Use joblib's load function to read the model back into memory.

#### **Where to Run the Code:**

You can run this code in your development environment, such as a Jupyter notebook, VS Code, or any Python IDE. The specific location within your script to place this code is after you have trained and evaluated your model.

#### **Detailed Breakdown:**

1. **Import joblib**:

python

Copy code

import joblib

1. **Save the trained model to a file**:

Use the dump function from joblib to save the pipeline (which includes the trained model and the preprocessing steps).

python

Copy code

# Save the model

joblib.dump(pipeline, 'spam\_detector.pkl')

Here, 'spam\_detector.pkl' is the name of the file where the model will be saved. You can choose any name you prefer.

1. **Load the model from the file**:

This step is for when you want to use the saved model later. You would typically run this code in a separate script or at a different time in your workflow.

python

Copy code

# Load the model

pipeline = joblib.load('spam\_detector.pkl')

#### **Full Example:**

Here is how you can integrate these steps into your existing code:

1. **Training and Saving the Model**:

python

Copy code

import numpy as np

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer, TfidfTransformer

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

from sklearn.naive\_bayes import MultinomialNB

from sklearn.pipeline import Pipeline

from sklearn import metrics

import nltk

from nltk.corpus import stopwords

import string

import joblib

# Download stopwords

nltk.download('stopwords')

stop\_words = set(stopwords.words('english'))

# Preprocessing function

def preprocess\_text(text):

text = ''.join([char for char in text if char not in string.punctuation])

text = text.lower()

text = ' '.join([word for word in text.split() if word not in stop\_words])

return text

# 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})

# Preprocess messages

data['message'] = data['message'].apply(preprocess\_text)

# Train-test split

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

# Build and train the pipeline

pipeline = Pipeline([

('vect', CountVectorizer()),

('tfidf', TfidfTransformer()),

('clf', MultinomialNB())

])

pipeline.fit(X\_train, y\_train)

# Predict on test set

y\_pred = pipeline.predict(X\_test)

# Evaluate the model

print(f'Accuracy: {metrics.accuracy\_score(y\_test, y\_pred)}')

print(metrics.classification\_report(y\_test, y\_pred, target\_names=['ham', 'spam']))

# Save the model

joblib.dump(pipeline, 'spam\_detector.pkl')

1. **Loading and Using the Saved Model**:

In a separate script or later in your workflow, you can load the saved model and use it to make predictions:

python

Copy code

import joblib

# Load the model

pipeline = joblib.load('spam\_detector.pkl')

# Test with new data

new\_emails = ["Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005.",

"Hey Bob, can we schedule a meeting for tomorrow?"]

new\_emails = [preprocess\_text(email) for email in new\_emails]

predictions = pipeline.predict(new\_emails)

print(predictions) # Output: array([1, 0])

### **Where to Run the Code:**

* **Jupyter Notebook**: If you are using a Jupyter notebook, you can run each cell individually. After training the model, add a new cell for saving the model and run it.
* **VS Code or Other IDEs**: If you are using VS Code or another IDE, you can run the entire script at once. Ensure that the dataset is available in the working directory and execute the script.
* **Command Line**: You can also run the Python script from the command line. Save the script in a .py file and run it using python script\_name.py.