Step 1: Load the Dataset

Assuming the dataset is a CSV with two columns like label and message:

```
import pandas as pd
# Load dataset
df = pd.read_csv('spam.csv', encoding='latin-1')[['v1', 'v2']]
df.columns = ['label', 'message']
```

Step 2: Preprocessing

Clean the text data to make it ready for vectorization

```
import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()

def preprocess(text):
    text = text.lower()
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    tokens = text.split()
    tokens = [stemmer.stem(word) for word in tokens if word not in stop_words]
    return " ".join(tokens)

df['cleaned'] = df['message'].apply(preprocess)
```

Step 3: Encode Labels

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['label_num'] = le.fit_transform(df['label'])  # ham: 0, spam: 1
```

Step 4: Feature Extraction (TF-IDF)

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vectorizer = TfidfVectorizer(max_features=3000)
X = vectorizer.fit_transform(df['cleaned'])
y = df['label_num']
```

Step 5: Train a Classifier

You can start with a simple and effective model like **Naive Bayes**.

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train model
model = MultinomialNB()
model.fit(X_train, y_train)
# Predictions
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```