In [1]:
```
pip install pandas scikit-learn nltk
```

```
Requirement already satisfied: pandas in ./anaconda3/lib/python3.11/sit
e-packages (1.5.3)
Requirement already satisfied: scikit-learn in ./anaconda3/lib/python3.
11/site-packages (1.3.2)
Requirement already satisfied: nltk in ./anaconda3/lib/python3.11/site-
packages (3.8.1)
Requirement already satisfied: python-dateutil>=2.8.1 in ./anaconda3/li
b/python3.11/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in ./anaconda3/lib/python3.
11/site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in ./anaconda3/lib/python
3.11/site-packages (from pandas) (1.24.3)
Requirement already satisfied: scipy>=1.5.0 in ./anaconda3/lib/python3.
11/site-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in ./anaconda3/lib/python
3.11/site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in ./anaconda3/lib/
python3.11/site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: click in ./anaconda3/lib/python3.11/site
-packages (from nltk) (8.0.4)
Requirement already satisfied: regex>=2021.8.3 in ./anaconda3/lib/pytho
n3.11/site-packages (from nltk) (2022.7.9)
Requirement already satisfied: tqdm in ./anaconda3/lib/python3.11/site-
packages (from nltk) (4.65.0)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.11/s
ite-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score
import nltk #Natural Langunage Tool Kit
from nltk.corpus import stopwords #StopWords are commonly used words
from nltk.stem import SnowballStemmer
import re
```

In [3]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/venkatasrideepthisrikotapeetamabaram/nltk_data.
[nltk_data]     ..
[nltk_data]   Package stopwords is already up-to-date!
```

Out[3]:
```
True
```

In [4]:
```
df = pd.read_csv("spam_ham_dataset.csv")
df
```

Out[4]:

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0
...	...	...	...	...
5166	1518	ham	Subject: put the 10 on the ft\r\nthe transport...	0
5167	404	ham	Subject: 3 / 4 / 2000 and following noms\r\nhp...	0
5168	2933	ham	Subject: calpine daily gas nomination\r\n>\r\n...	0
5169	1409	ham	Subject: industrial worksheets for august 2000...	0
5170	4807	spam	Subject: important online banking alert\r\ndea...	1

5171 rows × 4 columns

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5171 entries, 0 to 5170
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  5171 non-null   int64 
 1   label       5171 non-null   object
 2   text        5171 non-null   object
 3   label_num   5171 non-null   int64 
dtypes: int64(2), object(2)
memory usage: 161.7+ KB
```

In [6]:
```
#Data Preprocessing

def preprocessingText(text):
    text = re.sub('[^a-zA-Z]',' ',text)
    text.lower()
    words = text.split()
    words = [word for word in words if word not in stopwords.words('engl
    stemmer = SnowballStemmer('english')
    words = [stemmer.stem(word) for word in words]
    return ''.join(words)
```

In [7]:
```
# Apply preprocessing to the text data
df['text'] = df['text'].apply(preprocessingText)
df['text']
```

Out[7]:
```
0       subjectenronmethanolmeterfollownotegavemondayp...
1       subjecthplnomjanuariseeattachfilehplnolxlshpln...
2       subjectneonretreathohohoaroundwondertimeyearne...
3       subjectphotoshopwindowofficcheapmaintrendabasd...
4       subjectindianspringdealbooktecopvrrevenuunders...
                              ...
5166    subjectputfttransportvolumdecreascontractthank...
5167    subjectfollownomhpltakeextrammcfweekendtrinext...
5168    subjectcalpindailigasnominjulimentionearlierho...
5169    subjectindustriworksheetaugustactivattachworks...
5170    subjectimportonlinbankalertdearvalucitizensrba...
Name: text, Length: 5171, dtype: object
```

In [8]:
```
# Feature extraction using TF-IDF
vectorizer = TfidfVectorizer(max_features=3000)
X = vectorizer.fit_transform(df['text'])
X
```

Out[8]:
```
<5171x3000 sparse matrix of type '<class 'numpy.float64'>'
        with 3554 stored elements in Compressed Sparse Row format>
```

In [9]:
```
# Convert labels to binary (0 for ham, 1 for spam)
y = df['label'].apply(lambda x: 1 if x == 'spam' else 0)
y
```

Out[9]:
```
0       0
1       0
2       0
3       1
4       0
       ..
5166    0
5167    0
5168    0
5169    0
5170    1
Name: label, Length: 5171, dtype: int64
```

In [10]:
```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,

# Train SVM model
svm = SVC(kernel='linear')
svm.fit(X_train, y_train)
```

Out[10]:
```
▾        SVC

SVC(kernel='linear')
```

In [11]:
```
# Make predictions
y_pred = svm.predict(X_test)
y_pred
```

Out[11]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

In [12]:
```
# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.7265700483091787
Classification Report:
              precision    recall  f1-score   support

           0       0.72      1.00      0.84       742
           1       1.00      0.03      0.07       293

    accuracy                           0.73      1035
   macro avg       0.86      0.52      0.45      1035
weighted avg       0.80      0.73      0.62      1035
```

In [ ]:
```

```