

# codsoft-ml-3ipynb

August 5, 2024

## TASK-3 SPAM DETECTION MAIL

```
[1]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, \
    confusion_matrix
```

```
[2]: # Step 1: Load the dataset
df = pd.read_csv("/content/spam (1).csv", encoding='latin-1')
```

```
[3]: df.shape
```

```
[3]: (5572, 5)
```

```
[4]: df.head()
```

```
[4]:      v1                                     v2 Unnamed: 2  \
0  ham  Go until jurong point, crazy.. Available only ...      NaN
1  ham                                     Ok lar... Joking wif u oni...      NaN
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...      NaN
3  ham  U dun say so early hor... U c already then say...      NaN
4  ham  Nah I don't think he goes to usf, he lives aro...      NaN

      Unnamed: 3  Unnamed: 4
0           NaN           NaN
1           NaN           NaN
2           NaN           NaN
3           NaN           NaN
4           NaN           NaN
```

```
[6]: # Check the actual column names in your DataFrame
print(df.columns)
```

```
Index(['v1', 'v2', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')
```

```
[7]: # Step 2: Preprocessing
# Assuming the dataset has 'Category' and 'Message' columns
X = df['v2']
y = df['v1']
```

```
[8]: #Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↳random_state=42)
```

```
[9]: # Step 3: Feature Engineering
# Convert text data into numerical features
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

```
[10]: # Step 4: Model Building
# Initialize SVM model
svm_model = SVC(kernel='linear')
```

```
[11]: #Train the model
svm_model.fit(X_train, y_train)
```

```
[11]: SVC(kernel='linear')
```

```
[12]: # Model evaluation
y_pred = svm_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Accuracy: 0.979372197309417

```
[13]: # Classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Classification Report:

	precision	recall	f1-score	support
ham	0.98	1.00	0.99	965
spam	0.97	0.87	0.92	150
accuracy			0.98	1115
macro avg	0.98	0.93	0.95	1115
weighted avg	0.98	0.98	0.98	1115