

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
```

```
df = pd.read_csv("emails.csv")
```

```
df.head()
```

	Email No.	the	to	ect	and	for	of	a	you	hou	...	connevey
0	Email 1	0	0	1	0	0	0	2	0	0	...	0
1	Email 2	8	13	24	6	6	2	102	1	27	...	0
2	Email 3	0	0	1	0	0	0	8	0	0	...	0
3	Email 4	0	5	22	0	5	1	51	2	10	...	0
4	Email 5	7	6	17	1	5	2	57	0	9	...	0

	valued	lay	infrastructure	military	allowing	ff	dry
0	0	0		0	0	0	0
1	0	0		0	0	0	1
2	0	0		0	0	0	0
3	0	0		0	0	0	0
4	0	0		0	0	0	1

```
[5 rows x 3002 columns]
```

```
df = df.drop(columns=['Email No.'])
```

```
X = df.drop(columns='Prediction') # Features (word columns)
y = df['Prediction'] # Target (Spam or Not Spam)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

```
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)
knn_predictions = knn_model.predict(X_test)
```

```

svm_model = SVC(kernel='linear')
svm_model.fit(X_train, y_train)
svm_predictions = svm_model.predict(X_test)

def evaluate_model(y_test, predictions):
    print("Accuracy:", accuracy_score(y_test, predictions))
    print("Precision:", precision_score(y_test, predictions))
    print("Recall:", recall_score(y_test, predictions))
    print("F1 Score:", f1_score(y_test, predictions))

# Evaluate KNN
print("KNN Model Performance:")
evaluate_model(y_test, knn_predictions)

# Evaluate SVM
print("\nSVM Model Performance:")
evaluate_model(y_test, svm_predictions)

KNN Model Performance:
Accuracy: 0.8628019323671497
Precision: 0.7251461988304093
Recall: 0.8378378378378378
F1 Score: 0.7774294670846394

SVM Model Performance:
Accuracy: 0.9594202898550724
Precision: 0.9205298013245033
Recall: 0.9391891891891891
F1 Score: 0.9297658862876255

```