


Assignment No.: 2

Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance. Dataset link: The emails.csv dataset on the Kaggle <https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv>

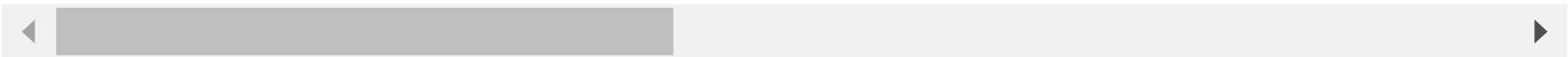
```
import pandas as pd
```

```
data= pd.read_csv("/content/emails.csv")
```


```
data.head()
```



	Email No.	the	to	ect	and	for	of	a	you	hou	...	connevey	jay	valued	lay	infrastr
0	Email 1	0	0	1	0	0	0	2	0	0	...	0	0	0	0	
1	Email 2	8	13	24	6	6	2	102	1	27	...	0	0	0	0	
2	Email 3	0	0	1	0	0	0	8	0	0	...	0	0	0	0	




```
data.isnull().sum()
```



	0
Email No.	0
the	0
to	0
ect	0
and	0
...	...
military	0
allowing	0
ff	0
dry	0
Prediction	0

3002 rows × 1 columns



KNN Model

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifierx = data.drop(["Prediction", "Emai
```

```
y = data["Prediction"]

xTrain , xTest , yTrain, yTest = train_test_split(x,y,test_size=0.2,random_state=1)

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(xTrain,yTrain)
y_pred_knn = knn.predict(xTest)
knn.score(xTest,yTest)

0.8647342995169082
```

Performance Evaluation of KNN

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,f1_score
print(f"Accuracy Score : {accuracy_score(yTest,y_pred_knn)}")
print(f"Precision Score : {precision_score(yTest,y_pred_knn)}")
print(f"Recall Score  :{recall_score(yTest,y_pred_knn)}")
print(f"F1 Score : {f1_score(yTest,y_pred_knn)}")

Accuracy Score : 0.8647342995169082
Precision Score : 0.75
Recall Score  :0.8354430379746836
F1 Score : 0.7904191616766467
```

SVM Model

```
from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(xTrain, yTrain)
y_pred_svm = svm_model.predict(xTest)
```

Performance Evaluation of KNN

```
print(f"Accuracy Score : {accuracy_score(yTest,y_pred_svm)}")
print(f"Precision Score : {precision_score(yTest,y_pred_svm)}")
print(f"Recall Score  :{recall_score(yTest,y_pred_svm)}")
print(f"F1 Score : {f1_score(yTest,y_pred_svm)}")

Accuracy Score : 0.7990338164251207
Precision Score : 0.8698630136986302
Recall Score  :0.40189873417721517
F1 Score : 0.5497835497835498
```