ModelDevelopmentPhaseTemplate

Date	06-06-2024
TeamID	740055
Project Title	DETECTION OF PHISHING WEBSITE FROM URLS
Maximum Marks	4 Marks

InitialModelTrainingCode,ModelValidationandEvaluationReport

Theinitialmodeltrainingcodewillbeshowcasedinthefuturethroughascreenshot.Themodel validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

InitialModelTrainingCode:

[35]:
```
#Splitting data as independent and dependent
#removing index column in independent dataset
x=ds.iloc[:,1:31].values
y=ds.iloc[:, -1].values
print(x,y)
```

```
[[ 37.  19.   0. ...   0.   0.   0.]
 [ 77.  23.   1. ...   0.   0.   0.]
 [126.  50.   1. ...   0.   0.   0.]
 ...
 [105.  16.   1. ...   0.   0.   0.]
 [ 38.  30.   0. ...   0.   0.   0.]
 [477.  14.   1. ...   0.   0.   1.]] ['legitimate' 'phishing' 'phishing' ... 'legitimate' 'legitimate'
 'phishing']
```

[36]:
```
y=ds.iloc[:, -1].values
y
```

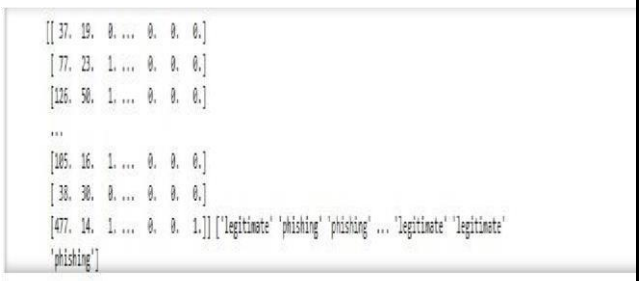
[36]:
```
array(['legitimate', 'phishing', 'phishing', ..., 'legitimate',
       'legitimate', 'phishing'], dtype=object)
```

[37]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
# Split data after x and y are defined
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

```
[38]: from sklearn.linear_model import LogisticRegression
      model = LogisticRegression(max_iter=10000)
      model.fit(x_train, y_train)  # Use model to fit the training data

[38]: LogisticRegression
      LogisticRegression(max_iter=10000)

[39]: model.fit(x_train,y_train)

[39]: LogisticRegression
      LogisticRegression(max_iter=10000)

[40]: y_pred1=model.predict(x_test)

[41]: y_pred1 = model.predict(x_test)  # Predict labels (if not already done)

      from sklearn.metrics import accuracy_score

      accuracy = accuracy_score(y_test, y_pred1)
      print(f"Logistic Regression Accuracy: {accuracy:.4f}")  # Print accuracy with formatting

      Logistic Regression Accuracy: 0.8027
```

ModelValidationandEvaluationReport:

Model	Classification Report	F1 Score	Confusion Matrix

Logistic Regression		80%	---

Decision Tree	-	79%	`confusion_matrix(y_test,ypred)`
			
			`array([[62, 13],`
			`[23, 71]])`

KNN	`print(classification_report(y_test,ypred))`
	
	precision recall f1-score support
	
	Loan will be Approved 0.60 0.57 0.59 75
	Loan will not be Approved 0.67 0.69 0.68 94
	
	accuracy 0.64 169
	macro avg 0.63 0.63 0.63 169
	weighted avg 0.64 0.64 0.64 169
		64%	`confusion_matrix(y_test,ypred)`
			
			`array([[43, 32],`
			`[29, 65]])`

Gradient Boosting	`print(classification_report(y_test,ypred))`
	
	precision recall f1-score support
	
	Loan will be Approved 0.71 0.84 0.77 75
	Loan will not be Approved 0.85 0.72 0.78 94
	
	accuracy 0.78 169
	macro avg 0.78 0.78 0.77 169
	weighted avg 0.79 0.78 0.78 169
		78%	`confusion_matrix(y_test,ypred)`
			
			`array([[63, 12],`
			`[26, 68]])`