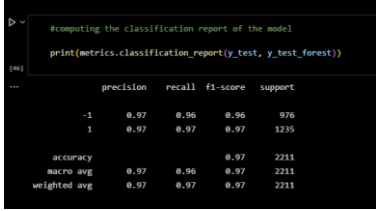
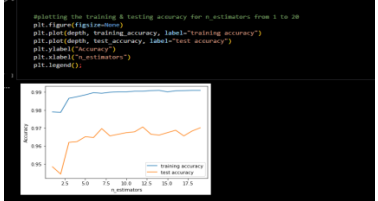


Project Development Phase Performance Testing

Team ID	PNT2022TMID52463
Project Name	Project – web phishing detection
Maximum Marks	10 Marks

Model Performance Testing:

Web phishing detection project team's performance testing using Random forest classification.

S.No.	Parameter	Values	Screenshot
1.	Metrics	Classification Model: Random forest classification Accuracy Score=96.6%	 <pre> #computing the classification report of the model print(metrics.classification_report(y_test, y_test_forest)) precision recall f1-score support -1 0.97 0.96 0.96 976 1 0.97 0.97 0.97 1235 accuracy 0.97 2211 macro avg 0.97 0.96 0.97 2211 weighted avg 0.97 0.97 0.97 2211 </pre>
2.	Tune the Model	Hyperparameter Tuning – 96% Validation Method – forest&cross validation	 <pre> #plotting the training & testing accuracy for n_estimators from 1 to 20 plt.figure(figsize=(10,6)) plt.plot(depth, training_accuracy, label="training accuracy") plt.plot(depth, test_accuracy, label="test accuracy") plt.plot(depth, accuracy, label="accuracy") plt.xlabel("n_estimators") plt.legend() </pre>

1.METRICS CLASSIFICATION REPORT:

```
# Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest = RandomForestClassifier(n_estimators=10)

# fit the model
forest.fit(x_train,y_train)
```

[13]

```
... RandomForestClassifier(n_estimators=10)
```

```
y_pred1=forest.predict(x_test)
from sklearn.metrics import accuracy_score
log_reg=accuracy_score(y_test,y_pred1)
log_reg
```

[14]

```
... 0.966078697421981
```

PERFORMANCE:

```
#plotting the training & testing accuracy for n_estimators from 1 to 20
plt.figure(figsize=None)
plt.plot(depth, training_accuracy, label="training accuracy")
plt.plot(depth, test_accuracy, label="test accuracy")
plt.ylabel("Accuracy")
plt.xlabel("n_estimators")
plt.legend();
```

