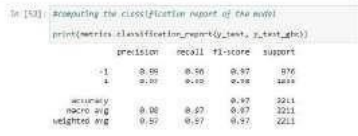



Project Development Phase  
Model Performance Test

Team ID	PNT2022TMID35524
Project Name	Project – Web Phishing Detection

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	Random forest  Accuray Score- 96.9	
2.	Tune the Model	Hyperparameter Tuning - 96.9 Validation Method – KFold Cross Validation Method	

1. METRICS:  
CLASSIFICATION REPORT:

```
In [40]: #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	956
1	0.97	0.98	0.97	1255
accuracy			0.97	2211
macro avg	0.97	0.97	0.97	2211
weighted avg	0.97	0.97	0.97	2211

## PERFORMANCE:

```
In [46]: # displaying total result
sorted_result
```

```
Out[46]:
```

	ML Model	Accuracy	f1_score	Recall	Precision
0	Random Forest	0.969	0.973	0.994	0.988
1	Decision Tree	0.960	0.965	0.992	0.991
2	Support Vector Machine	0.957	0.963	0.982	0.966
3	K-Nearest Neighbors	0.953	0.959	0.990	0.989
4	Logistic Regression	0.924	0.933	0.947	0.927

---

## TUNE THE MODEL – HYPERPARAMETER TUNING

```
In [51]: from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(random_state = 42)
from pprint import pprint
# Look at parameters used by our current forest
print('Parameters currently in use:\n')
pprint(rf.get_params())
```

Parameters currently in use:

```
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'criterion': 'squared_error',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}
```

## VALIDATION METHODS: KFOLD & Cross Folding

```
In [52]: rf = RandomForestClassifier(n_estimators=40)
rf.fit(X_train, y_train)
rf.score(X_test, y_test)
```

Out[52]: 0.966078697421981

```
In [57]: from sklearn.model_selection import cross_val_score
score_rf=cross_val_score(RandomForestClassifier(n_estimators=40),X, y,cv=3)
print(score_rf)
print(np.average(score_rf))
```

[0.96933514 0.97313433 0.92510176]  
0.9558570782451379

```
In [62]: scores1 = cross_val_score(RandomForestClassifier(n_estimators=5),X, y, cv=10)
print('Avg Score for Estimators=5 and CV=10 :')
print(np.average(scores1))
```

Avg Score for Estimators=5 and CV=10 :  
0.9660735764607693

```
In [63]: scores2 = cross_val_score(RandomForestClassifier(n_estimators=20),X, y, cv=10)
print('Avg Score for Estimators=20 and CV=10 :')
print(np.average(scores2))
```

Avg Score for Estimators=20 and CV=10 :  
Avg Score for Estimators=5 and CV=10 :  
0.9660735764607693

```
In [63]: scores2 = cross_val_score(RandomForestClassifier(n_estimators=20),X, y, cv=10)
print('Avg Score for Estimators=20 and CV=10 :')
print(np.average(scores2))
```

Avg Score for Estimators=20 and CV=10 :  
0.972134879268163

```
In [64]: scores3 = cross_val_score(RandomForestClassifier(n_estimators=30),X, y, cv=10)
print('Avg Score for Estimators=30 and CV=10 :')
print(np.average(scores3))
```

Avg Score for Estimators=30 and CV=10 :  
0.972859106641683

```
In [65]: scores4 = cross_val_score(RandomForestClassifier(n_estimators=40),X, y, cv=10)
print('Avg Score for Estimators=40 and CV=10 :')
print(np.average(scores4))
```

Avg Score for Estimators=40 and CV=10 :  
0.9727681179579915