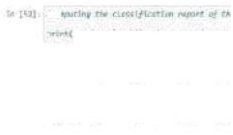
Project Development Phase

Model Performance Test

Date	13 November 2022
Team ID	PNT2022TMID15184
Project Name	Project — Web Phishing Detection
Maximum Marks	10 Marks

Model Performance Testing:

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

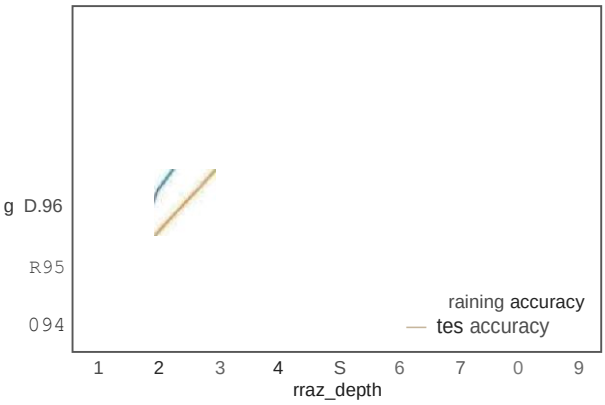
S.No.	Parameter	**Values**	**Screenshot**
1.	Metrics	**Classification Model:** Gradient Boosting Classification Accuracy Score- 97.4%	
2.	Tune the Model	Hyperparameter Tuning - 97% Validation Method — KFOLD & Cross Validation Method	

1. METRICS:

CLASSIFICATION REPORT:

```
In [52]: #computing the classification report of the model

         p+  nt(metrics.classification_report(y_test, y_test_gbc))

                      precision     nacal1   +1-score    support

                 —1     0..°s       e. 6      e..°7       .76
                  1     g . 97      0 . 99     0 . 96     1235

        accurac;'                              0.97      J2 1 1
       macro avg       .0  8       8.°7       0.97      221 1
    weighted avg       9. 97       0.°7       0 . 97    221 1
```

PERFORMANCE :



	ML *Moâet*	Accuracy	fJ_score	Recall	Precision
0	Cradient Boosting C tassifier	0.974	0.977	0.994	0.986
1	CatBoost Classified	0.972	0.975	0.994	0.989
2	Random Forest	0.96•	0.972	0.992	0.991
3	Support Vector Machine	0.964	0.968	0.980	0.96S
4	Decision Tree	0.938	0.962	0.991	0.993
5	K-Nearest Neighbors	0.956	0.9fi1	0.991	0.989
6	Log stic Regression	0.934	0.941	0.943	0.927
7	Naive Bayes ClaSsifier	0.605	0.454	0.292	0.997
8	XGBoonCla§fiKer	0.548	0.348	0.993	0.984
9	Mufti-layer Perceptron	0.543	0.543	0.989	0.983

2. TUNE THE MODEL - HYPERPARAMETER TUNING

```
In [58]: #HYPERPARAMETER TUNING
         grid.fit(X_train, y_train)
```

```
GridSearchCV
```

```
GridSearchCV(cv=5,
             estimator=GradientBoostingClassifier(learning_rate=0.7,
                                                  max_depth=4),
```

```
'çeitia' r.í: rraÿ| 1?, Ø, å, 4h iØ, (Ø, Ø, , )Ø lB, 1], lã, 13Ø
MlØlĄlĘ]tMĄNĘ)
''tilt''.;''{''tØœ'ti%l'''iíi''
GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

```
GradientBoostingClassifier
GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

```
In [59]: print("The best parameters are %s with a score of %0.2f"
         \ (grid.òest pants , grid.lest sroee_} )

The best parameters are {'max_features': 5, 'n_estimators': 200} with a score of 0.97
```

VALIDATION METHODS: KFOLD & Cross Folding

Wllexon signed-rank test

```
In [7°]: #KFOLD and Cross Validation Model

         from sclpy.stats  font  »11coxon
         fno+  sk1earn.datasets  b«po• t  1oad_1r1s
         +mm sk1earn.ense«tb1e Vapor-t GradlentBoest1ngC1ass1f1er
         from xgboost io}zort XGBC1ass:tf1er
         +n    sklearn.«odc1_se1ectSon  Prepaid  cross_va1_scorc,  Krold

         4 Load th.e doTcsel
         X =  1oad_1r1s ( ) . data
         y =  1oad_1rds ( ) . target

         # Prepare models and select your CV method
         modell = GradientBoostingClassifier(n_esticators=100)
         mWel2 = XG8Clssifier(n_estinators=1   )
         kf  = KFol d(n_sp1l ts =Z0,   randoni_state-None)
        4  Ex.W acl resulLe for  each .'nodes  or'. the sa.    foL d
         result s_nIodell - cross_va1_score(r«ode11, X, y, cv=kf)
         resu1ts_«tocIe12 = cross_va1_score(«ode12, X, y, cv=kf)
         stat.  p  =  w11coxon(resu1ts_m>de11,  results_eode12,  zero_eethod= 'zsp1it'),
         stat
```

5z2CV combined F test

```
In [89]: frcm  o1xtend.eva1uate  1mpoi-I  ccmb1ned_ftest_5x2cv
         fr"oo  sk1corn.tree  Import  Dec1slonTreeC1ossTfler,  ExtretreeCiassi-tSer
         from sk1earn.ensetrb1e  import GradlentBoost1ngC1asstf1er
         frmi  u1xtend.data  1npoi-t  1r1s_data

         *  P. epa. e  fiiata  or.d  cl.fis
         X,  y  =  1r1s_data()
         cIII    = Grad1ent8oostlngC1ass Liber ( )
         c1f2 = DeclsionTreeC1ass11°1er ()

         # Calculate p-value
         f, p = combined_ftest_5x2cv(estimator1=clf1,
                                     estimator2=clf2,
                                     X=X, y=y,
                                     random_seed=1)

         print('I - va 1ue-  ', I )
         pr int (  p-va Ke  : ',  p)

         f-va1ue: 1.727272727272733
         p-ve1ue:  6.2B40'135734291782
```