

Project Development PhaseModel Performance Test

Date	13 November 2022
Team ID	PNT2022TMID10436
Project Name	Project – Web Phishing Detection
Maximum Marks	10 Marks

Model Performance Testing:

```
In [52]: #Computing the classification report of the model
print(metrics.classification_report(y_test, y_test_gbc))

          precision    recall  f1-score   support

     -1         0.98         0.96         0.97         879
      1         0.97         0.99         0.98         1335

 accuracy         0.98
 macro avg         0.98         0.97         0.97         2214
 weighted avg         0.97         0.97         0.97         2214
```

```
Wilcoxon signed-rank test

In [74]: #Wilcoxon and Cross Validation Model

from scipy.stats import wilcoxon
from sklearn.datasets import load_iris
from sklearn.model_selection import GradientBoostingClassifier
from sklearn.metrics import cross_val_score, kfold
from sklearn.model_selection import cross_val_score, kfold

# Load the dataset
x = load_iris().data
y = load_iris().target

# Prepare models and select your CV method
model1 = GradientBoostingClassifier(n_estimators=100)
model2 = NBClassifier(n_estimators=100)
kf = KFold(n_splits=5, random_state=None)

# Extract results for each model on the new folds
results_model1 = cross_val_score(model1, x, y, cv=kf)
results_model2 = cross_val_score(model2, x, y, cv=kf)
step_2 = {'model': results_model1, 'results_model1', 'results_model2', 'cv_method': 'cv(kf)'}
print(step_2)
```

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	

• METRICS: CLASSIFICATION REPORT:

```
In [52]: #computing the classification report of the model
print(metrics.classification_report(y_test, y_test_gbc))
```

	precision	recall	f1-score	support
-1	0.99	0.96	0.97	976
1	0.97	0.99	0.98	1235
accuracy			0.97	2211
macro avg	0.98	0.97	0.97	2211
weighted avg	0.97	0.97	0.97	2211

PERFORMANCE :

- TUNE THE MODEL – HYPERPARAMETER TUNING

VALIDATION METHODS: KFOLD & Cross Folding

Wilcoxon signed-rank test

```
In [78]: #KFOLD and Cross Validation Model

from scipy.stats import wilcoxon
from sklearn.datasets import load_iris
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import cross_val_score, KFold

# Load the dataset
X = load_iris().data
y = load_iris().target

# Prepare models and select your CV method
model1 = GradientBoostingClassifier(n_estimators=100)
model2 = XGBClassifier(n_estimators=100)
kf = KFold(n_splits=20, random_state=None)
# Extract results for each model on the same folds
results_model1 = cross_val_score(model1, X, y, cv=kf)
results_model2 = cross_val_score(model2, X, y, cv=kf)
stat, p = wilcoxon(results_model1, results_model2, zero_method='zsplit');
stat
```

Out[78]: 95.0

5x2CV combined F test

```
In [89]: from mlxtend.evaluate import combined_ftest_5x2cv
from sklearn.tree import DecisionTreeClassifier, ExtraTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from mlxtend.data import iris_data

# Prepare data and clfs
X, y = iris_data()
clf1 = GradientBoostingClassifier()
clf2 = DecisionTreeClassifier()

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

print('f-value:', f)
print('p-value:', p)

f-value: 1.727272727272733
p-value: 0.2840135734291782
```