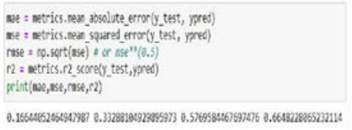
Project Development Phase
Model Performance Test

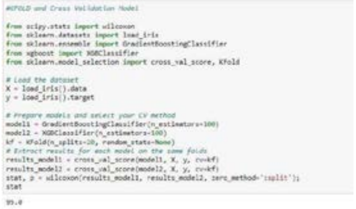
Date	10 November 2022
Team ID	PNT2022TMID41762
Project Name	Web Phishing Detection
Maximum Marks	10 Marks

Model Performance Testing:

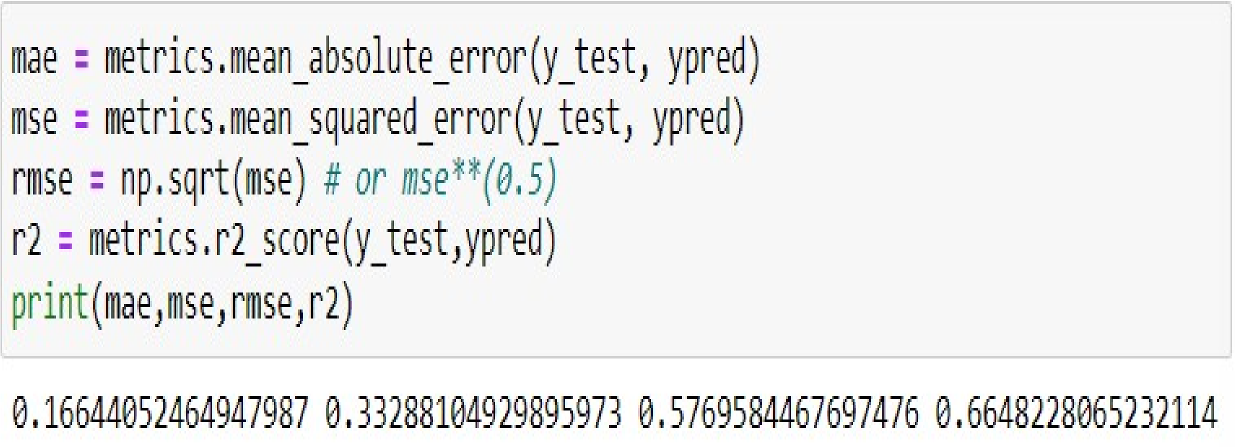
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot

1.	Metrics	**Regression Model:**
		MAE – 16.6%, MSE -33.2, RMSE – 57.6%, R2 score – 66.4%
			
			mae = metrics.mean_absolute_error(y_test, ypred)
			mse = metrics.mean_squared_error(y_test, ypred)
			rmse = np.sqrt(mse) # or mse**(0.5)
			r2 = metrics.r2_score(y_test,ypred)
			print(mae,mse,rmse,r2)
			0.16644052464947987 0.33288104929895973 0.5769584467697476 0.6648228065232114

2.	Tune the Model	Hyperparameter Tuning – 95%
		Validation Method – KFOLD & Cross Validation Method
			
			#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
			99.0

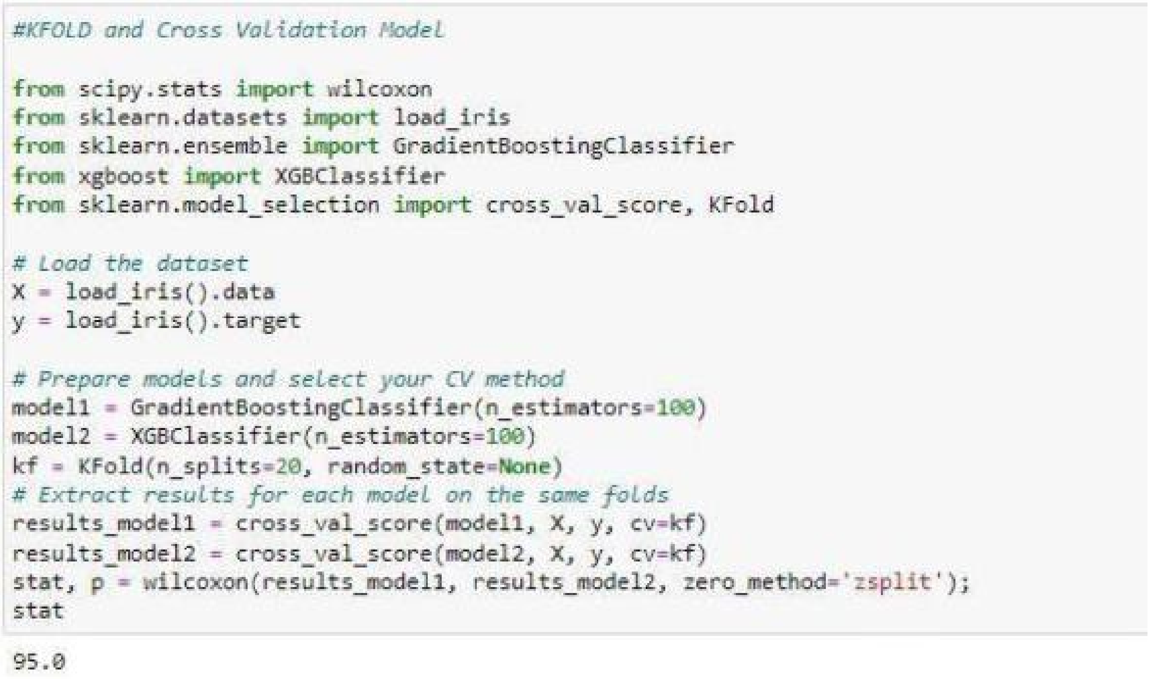
Metrics:



```
mae = metrics.mean_absolute_error(y_test, ypred)
mse = metrics.mean_squared_error(y_test, ypred)
rmse = np.sqrt(mse) # or mse**(0.5)
r2 = metrics.r2_score(y_test,ypred)
print(mae,mse,rmse,r2)
```

```
0.16644052464947987 0.33288104929895973 0.5769584467697476 0.6648228065232114
```

Tune the model:



```
#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
```

```
95.0
```