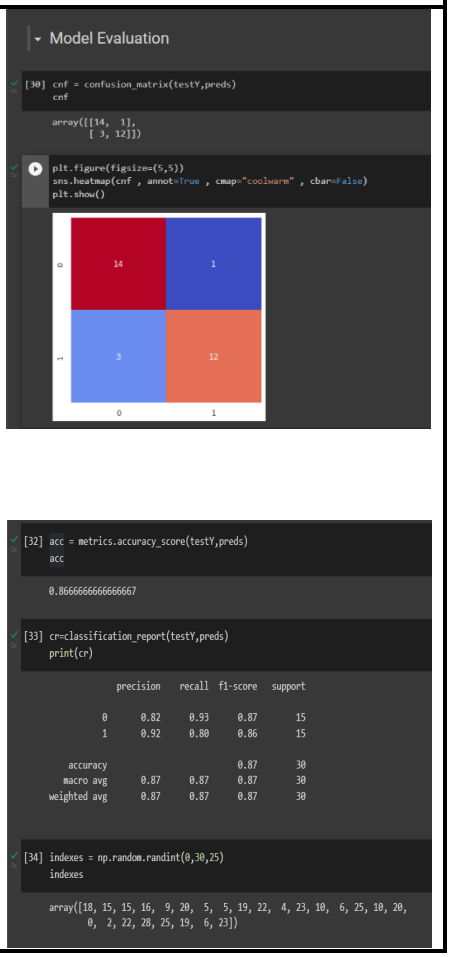
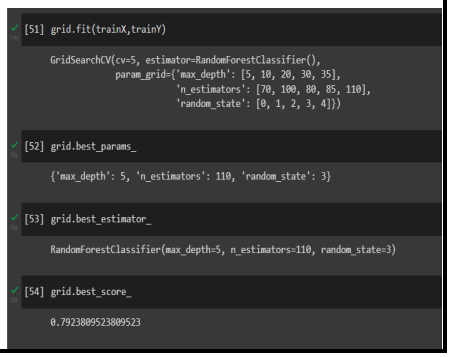
Project Development Phase
Model Performance Test

Team ID	PNT2022TMID24632
Project Name	Project - Detecting Parkinson's Disease using Machine Learning.
Maximum Marks	10 Marks

ML Model Performance Testing:

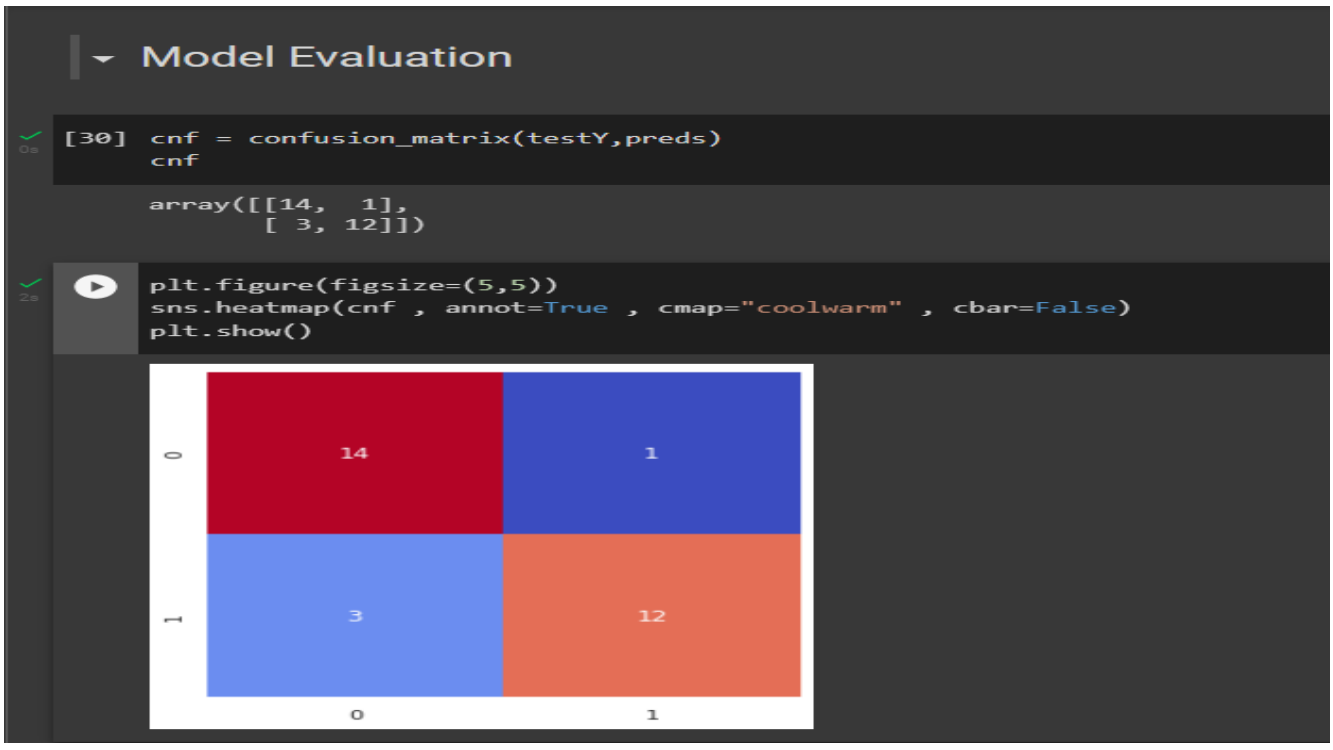
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix, Accuracy Score & Classification Report	

2.	Tune the Model	Hyperparameter Tuning -GridSearchCV	 

BRIEF DETAILED SCREENSHOTS:

METRICS:



```
[32] acc = metrics.accuracy_score(testY,preds)
     acc

     0.8666666666666667

[33] cr=classification_report(testY,preds)
     print(cr)

                   precision    recall  f1-score   support

                0       0.82      0.93      0.87        15
                1       0.92      0.80      0.86        15

         accuracy                           0.87        30
        macro avg       0.87      0.87      0.87        30
     weighted avg       0.87      0.87      0.87        30


[34] indexes = np.random.randint(0,30,25)
     indexes

     array([18, 15, 15, 16,  9, 20,  5,  5, 19, 22,  4, 23, 10,  6, 25, 10, 20,
             0,  2, 22, 28, 25, 19,  6, 23])
```

TUNE THE MODEL:

Tuning the model

```
[47] from sklearn.model_selection import GridSearchCV

[48] model=RandomForestClassifier()

[49] parameters={
         'max_depth': [5,10,20,30,35],
         'random_state':[0,1,2,3,4],
         'n_estimators':[70,100,80,85,110]
     }

[50] grid=GridSearchCV(model,parameters,cv=5)

[51] grid.fit(trainX,trainY)

     GridSearchCV(cv=5, estimator=RandomForestClassifier(),
                  param_grid={'max_depth': [5, 10, 20, 30, 35],
                              'n_estimators': [70, 100, 80, 85, 110],
                              'random_state': [0, 1, 2, 3, 4]})
```

```
[51] grid.fit(trainX,trainY)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [5, 10, 20, 30, 35],
                         'n_estimators': [70, 100, 80, 85, 110],
                         'random_state': [0, 1, 2, 3, 4]})
```

```
[52] grid.best_params_
```

```
{'max_depth': 5, 'n_estimators': 110, 'random_state': 3}
```

```
[53] grid.best_estimator_
```

```
RandomForestClassifier(max_depth=5, n_estimators=110, random_state=3)
```

```
[54] grid.best_score_
```

```
0.7923809523809523
```