

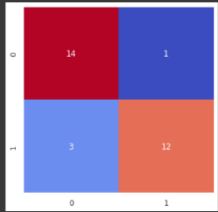
Project Development Phase

Model Performance Test

Team ID	PNT2022TMID40374
Project Name	Project - Detecting Parkinson's Disease using Machine Learning.
Maximum Marks	10 Marks

ML Model Performance Testing:

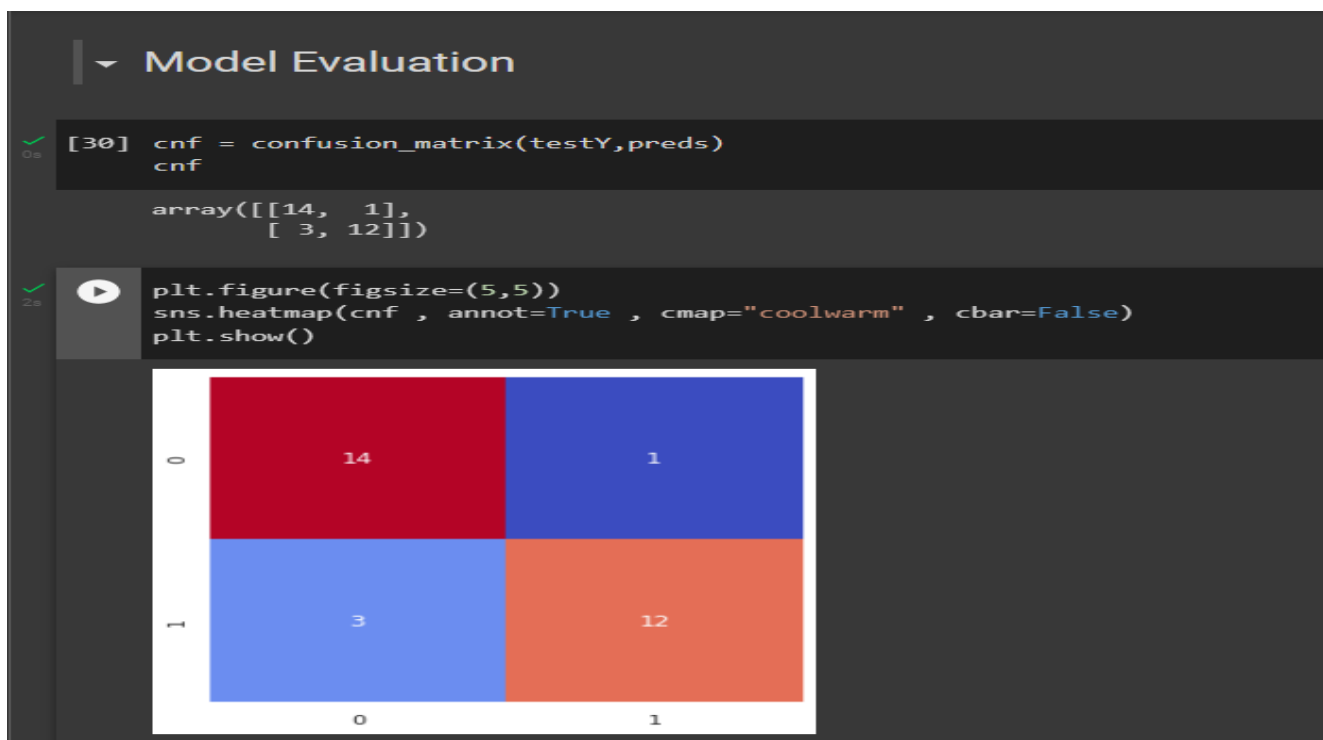
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot																														
1.	Metrics	Classification Model: Confusion Matrix, Accuracy Score & Classification Report	<div><div>Model Evaluation</div><pre>[30] cmf = confusion_matrix(testY,preds) cmf array([[14, 1], [3, 12]])</pre><pre>plt.figure(figsize=(5,5)) sns.heatmap(cmf, annot=True, cmap="coolwarm", cbar=False) plt.show()</pre></div> <div><pre>[32] acc = metrics.accuracy_score(testY,preds) acc 0.8666666666666667</pre><pre>[33] cr=classification_report(testY,preds) print(cr)</pre><table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.82</td><td>0.93</td><td>0.87</td><td>15</td></tr><tr><td>1</td><td>0.92</td><td>0.80</td><td>0.86</td><td>15</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.87</td><td>30</td></tr><tr><td>macro avg</td><td>0.87</td><td>0.87</td><td>0.87</td><td>30</td></tr><tr><td>weighted avg</td><td>0.87</td><td>0.87</td><td>0.87</td><td>30</td></tr></table><pre>[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])</pre></div>		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
	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																													

2.	Tune the Model	Hyperparameter Tuning - GridSearchCV	<div data-bbox="1161 275 1596 632"> <p>▼ Tuning the model</p> <pre> [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]}]) </pre> </div> <div data-bbox="1161 678 1596 1005"> <pre> [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 </pre> </div>
----	----------------	---	---

BRIEF DETAILED SCREENSHOTS:

METRICS:



TUNE THE MODEL:

▼ Tuning the model

✓
1m

[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]})
```

✓
0s

[52] grid.best_params_

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

✓
0s

[53] grid.best_estimator_

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

✓
0s

[54] grid.best_score_

```
0.7923809523809523
```