Project Development Phase
Model Performance Test

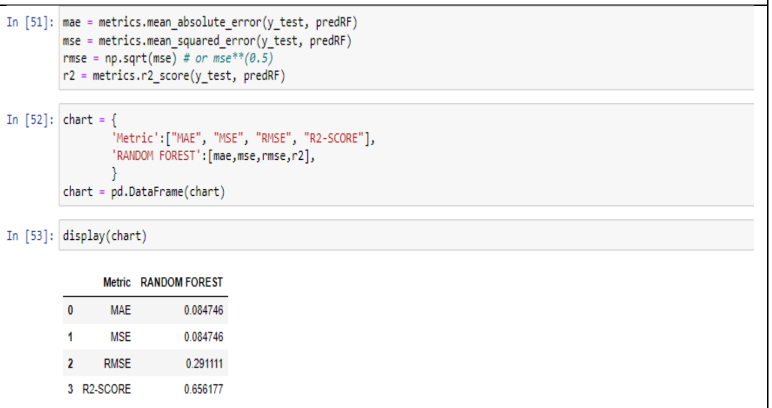
Date	10 November 2022
Team ID	PNT2022TMID40423
Project Name	Detection of Parkinson's disease using machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

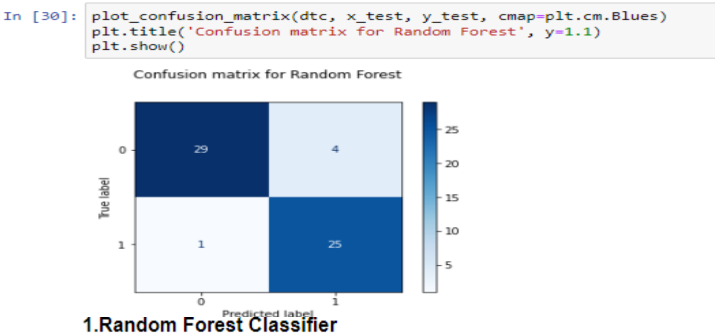
Project team shall fill the following information in model performance testing template.

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr>
<td>1.</td>
<td>Metrics</td>
<td><b>Regression Model:</b>
MAE - , MSE - , RMSE - , R2 score -

<b>Classification Model:</b>
Confusion Matrix - , Accuray Score- & Classification Report -</td>
<td>

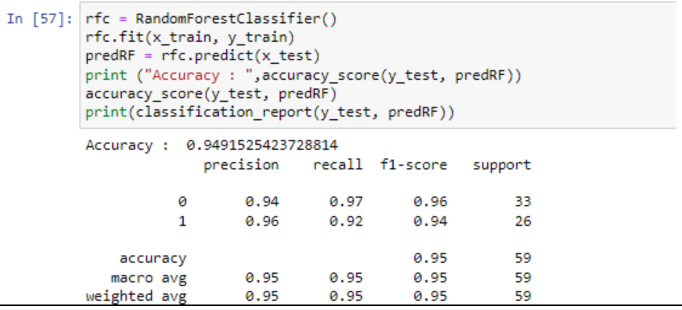


```
In [51]: mae = metrics.mean_absolute_error(y_test, predRF)
         mse = metrics.mean_squared_error(y_test, predRF)
         rmse = np.sqrt(mse) # or mse**(0.5)
         r2 = metrics.r2_score(y_test, predRF)

In [52]: chart = {
                'Metric':["MAE", "MSE", "RMSE", "R2-SCORE"],
                'RANDOM FOREST':[mae,mse,rmse,r2],
                }
         chart = pd.DataFrame(chart)

In [53]: display(chart)

     Metric  RANDOM FOREST
0       MAE       0.084746
1       MSE       0.084746
2      RMSE       0.291111
3  R2-SCORE       0.656177
```



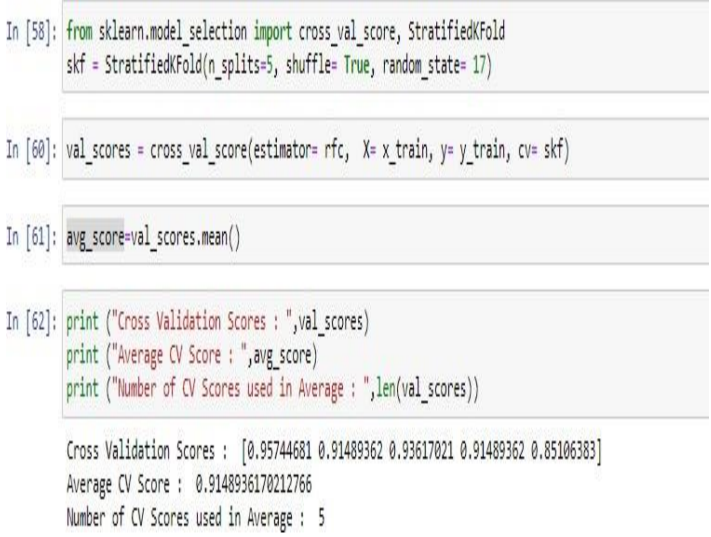
```
In [30]: plot_confusion_matrix(dtc, x_test, y_test, cmap=plt.cm.Blues)
         plt.title('Confusion matrix for Random Forest', y=1.1)
         plt.show()
```

1.Random Forest Classifier



```
In [57]: rfc = RandomForestClassifier()
         rfc.fit(x_train, y_train)
         predRF = rfc.predict(x_test)
         print ("Accuracy : ",accuracy_score(y_test, predRF))
         accuracy_score(y_test, predRF)
         print(classification_report(y_test, predRF))

         Accuracy :  0.9491525423728814
                       precision    recall  f1-score   support

                    0       0.94      0.97      0.96        33
                    1       0.96      0.92      0.94        26

             accuracy                           0.95        59
            macro avg       0.95      0.95      0.95        59
         weighted avg       0.95      0.95      0.95        59
```

</td>
</tr>
</table>

2.	Tune the Model	Hyperparameter Tuning - Validation Method -	In [58]: `from sklearn.model_selection import cross_val_score, StratifiedKFold`
			`skf = StratifiedKFold(n_splits=5, shuffle= True, random_state= 17)`
			In [60]: `val_scores = cross_val_score(estimator= rfc,  X= x_train, y= y_train, cv= skf)`
			In [61]: `avg_score=val_scores.mean()`
			In [62]: `print ("Cross Validation Scores : ",val_scores)`
			`print ("Average CV Score : ",avg_score)`
			`print ("Number of CV Scores used in Average : ",len(val_scores))`
			`Cross Validation Scores :  [0.95744681 0.91489362 0.93617021 0.91489362 0.85106383]`
			`Average CV Score :  0.9148936170212766`
			`Number of CV Scores used in Average :  5`