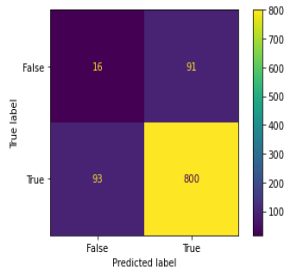


Project Development Phase Model Performance Test

| | |
|---------------|--|
| Date | 19 November 2022 |
| Team ID | PNT2022TMD04288 |
| Project Name | University Admit Eligibility Predictor |
| 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 | <p>Regression Model: MAE - 0.04555 MSE - 0.00426 RMSE – 0.06527 , R2 score – 0.71683</p> <p>Classification Model: Confusion Matrix - Accuracy Score -0.82 Classification Report -</p> | <p>Mean Squared Error (MSE)</p> <pre>from sklearn.metrics import mean_squared_error, r2_score mse = mean_squared_error(pred_test,y_test)</pre> <p>[25] mse</p> <p>0.004260810050671112</p> <p>Root Mean Squared Error (RMSE)</p> <pre>[26] rmse = np.sqrt(mse)</pre> <p>[27] rmse</p> <p>0.06527488070208257</p> <p>R2 Score</p> <pre>[28] r2_score(pred_test, y_test)</pre> <p>0.716831867909245</p> <p>Mean Absolute Error (MAE)</p> <pre>[29] from sklearn.metrics import mean_absolute_error mean_absolute_error(pred_test, y_test)</pre> <p>0.04555243196630539</p>  |

| | | | <div><div><div><div>In [44]:</div><div>Accuracy = metrics.accuracy_score(actual, predicted)</div><div>Accuracy</div></div></div><div><div>Out[44]:</div><div>0.816</div></div><div><div>Classification Report</div><div><div>In [56]:</div><div><pre>from sklearn.metrics import classification_report print(classification_report(actual, predicted))</pre></div></div><div><table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.15</td><td>0.15</td><td>0.15</td><td>107</td></tr><tr><td>1</td><td>0.90</td><td>0.90</td><td>0.90</td><td>893</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.82</td><td>1000</td></tr><tr><td>macro avg</td><td>0.52</td><td>0.52</td><td>0.52</td><td>1000</td></tr><tr><td>weighted avg</td><td>0.82</td><td>0.82</td><td>0.82</td><td>1000</td></tr></table></div></div></div> | | precision | recall | f1-score | support | 0 | 0.15 | 0.15 | 0.15 | 107 | 1 | 0.90 | 0.90 | 0.90 | 893 | accuracy | | | 0.82 | 1000 | macro avg | 0.52 | 0.52 | 0.52 | 1000 | weighted avg | 0.82 | 0.82 | 0.82 | 1000 |
|--------------|----------------|---|---|---------|-----------|--------|----------|---------|---|------|------|------|-----|---|------|------|------|-----|----------|--|--|------|------|-----------|------|------|------|------|--------------|------|------|------|------|
| | precision | recall | f1-score | support | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0.15 | 0.15 | 0.15 | 107 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0.90 | 0.90 | 0.90 | 893 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| accuracy | | | 0.82 | 1000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| macro avg | 0.52 | 0.52 | 0.52 | 1000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| weighted avg | 0.82 | 0.82 | 0.82 | 1000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. | Tune the Model | Hyperparameter Tuning - Validation Method | <div><div>2.Stratified K-Fold</div><div><div>In [63]:</div><div><pre>from sklearn import datasets from sklearn.tree import DecisionTreeClassifier from sklearn.model_selection import StratifiedKFold, cross_val_score X, y = datasets.load_iris(return_X_y=True) clf = DecisionTreeClassifier(random_state=42) sk_folds = StratifiedKFold(n_splits = 5) scores = cross_val_score(clf, X, y, cv = sk_folds) print("Cross Validation Scores: ", scores) print("Average CV Score: ", scores.mean()) print("Number of CV Scores used in Average: ", len(scores))</pre></div></div><div>Cross Validation Scores: [0.96666667 0.96666667 0.90933333 1.0 0.93333333 1.0] Average CV Score: 0.9533333333333334 Number of CV Scores used in Average: 5</div></div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |