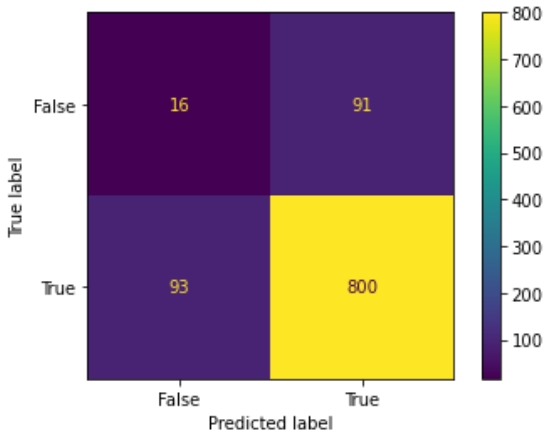
**Project Development Phase**
**Model Performance Test**

| | |
|---|---|
| Date | 19 November 2022 |
| Team ID | PNT2022TMID04255 |
| 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 | **Regression Model:**<br>MAE - , 0.04555<br>MSE - , 0.00426<br>RMSE – 0.06527 ,<br>R2 score – 0.71683<br><br>**Classification Model:**<br>Confusion Matrix -<br>Accuracy Score-<br>Classification Report - | **Mean Squared Error (MSE)**<br>In [25]: `from sklearn.metrics import mean_squared_error, r2_score`<br>`mse = mean_squared_error(pred_test,y_test)`<br>In [26]: `mse`<br>Out[26]: 0.004260810050671112<br>**Root Mean Squared Error (RMSE)**<br>In [27]: `rmse = np.sqrt(mse)`<br>In [28]: `rmse`<br>Out[28]: 0.06527488070208257<br>**R2 Score**<br>In [29]: `r2_score(pred_test, y_test)`<br>Out[29]: 0.7168318679092451<br>**Mean Absolute Error (MAE)**<br>In [32]: `from sklearn.metrics import mean_absolute_error`<br>`mean_absolute_error(pred_test, y_test)`<br>Out[32]: 0.0455524319663054<br>  |

| | | | |
|---|---|---|---|
| | | | In [44]:<br>```<br>Accuracy = metrics.accuracy_score(actual, predicted)<br>Accuracy<br>```<br>Out[44]: 0.816<br><br>**Classification Report**<br><br>In [56]:<br>```<br>from sklearn.metrics import classification_report<br>print(classification_report(actual, predicted))<br>```<br><br>| | precision | recall | f1-score | support |<br>|---|---|---|---|---|<br>| 0 | 0.15 | 0.15 | 0.15 | 107 |<br>| 1 | 0.90 | 0.90 | 0.90 | 893 |<br>| accuracy | | | 0.82 | 1000 |<br>| macro avg | 0.52 | 0.52 | 0.52 | 1000 |<br>| weighted avg | 0.82 | 0.82 | 0.82 | 1000 |
| 2. | Tune the Model | Hyperparameter Tuning - Validation Method – | **2.Stratified K-Fold**<br><br>In [63]:<br>```<br>from sklearn import datasets<br>from sklearn.tree import DecisionTreeClassifier<br>from sklearn.model_selection import StratifiedKFold, cross_val_score<br><br>X, y = datasets.load_iris(return_X_y=True)<br><br>clf = DecisionTreeClassifier(random_state=42)<br><br>sk_folds = StratifiedKFold(n_splits = 5)<br><br>scores = cross_val_score(clf, X, y, cv = sk_folds)<br><br>print("Cross Validation Scores: ", scores)<br>print("Average CV Score: ", scores.mean())<br>print("Number of CV Scores used in Average: ", len(scores))<br>```<br><br>Cross Validation Scores: [0.96666667 0.96666667 0.9 0.93333333 1. ]<br>Average CV Score: 0.9533333333333334<br>Number of CV Scores used in Average: 5 |