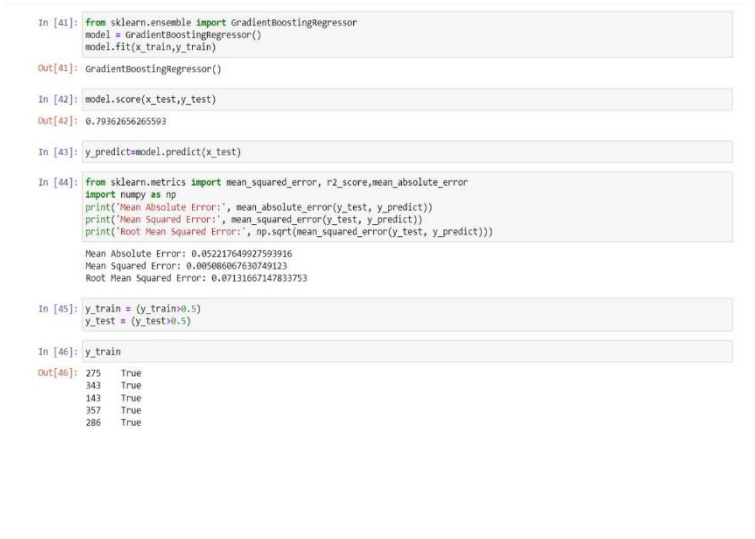
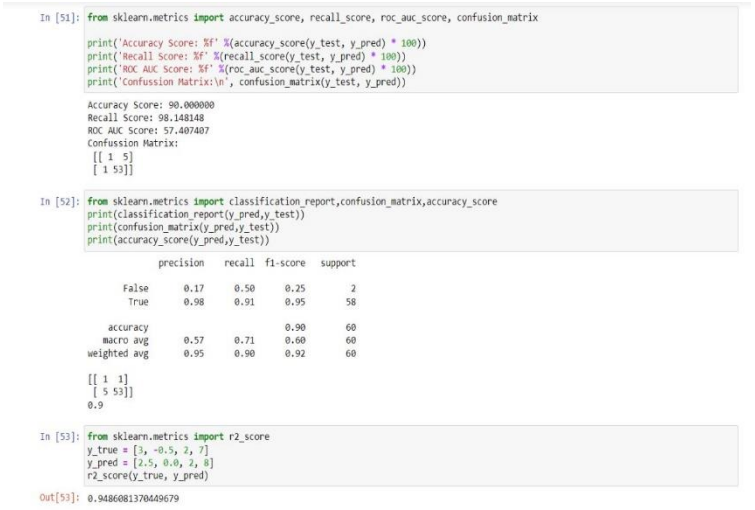
Project Development Phase
Model Performance Test
Applied Data Science

Date	10 November 2022
Team ID	PNT2022TMID00994
Project Name	Project – University Admit Eligibility Predictor
Maximum Marks	10 Marks

Model Performance Testing:

Performance metrics

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE = 0.0522 MSE = 0.0050 RMSE = 0.0713 R2 score =0.9486 **Classification Model:** Confusion Matrix = [[1 5] [1 53] Accuracy Score= 90.0000 Recall score= 98.1481 ROC AUC Score= 57.407	 

```
In [41]: from sklearn.ensemble import GradientBoostingRegressor
         model = GradientBoostingRegressor()
         model.fit(x_train,y_train)
Out[41]: GradientBoostingRegressor()

In [42]: model.score(x_test,y_test)
Out[42]: 0.79362656265593

In [43]: y_predict=model.predict(x_test)

In [44]: from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error
         import numpy as np
         print('Mean Absolute Error:', mean_absolute_error(y_test, y_predict))
         print('Mean Squared Error:', mean_squared_error(y_test, y_predict))
         print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_predict)))

         Mean Absolute Error: 0.05221764992759391 6
         Mean Squared Error: 0.005086067630749123
         Root Mean Squared Error: 0.07131667147833753

In [45]: y_train = (y_train>0.5)
         y_test = (y_test>0.5)

In [46]: y_train
Out[46]: 275    True
         343    True
         143    True
         357    True
         286    True
```

```
In [51]: from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix

         print('Accuracy Score: %f' %(accuracy_score(y_test, y_pred) * 100))
         print('Recall Score: %f' %(recall_score(y_test, y_pred) * 100))
         print('ROC AUC Score: %f' %(roc_auc_score(y_test, y_pred) * 100))
         print('Confussion Matrix:\n', confusion_matrix(y_test, y_pred))

         Accuracy Score: 90.000000
         Recall Score: 98.148148
         ROC AUC Score: 57.407407
         Confussion Matrix:
          [[ 1  5]
          [ 1 53]]

In [52]: from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
         print(classification_report(y_pred,y_test))
         print(confusion_matrix(y_pred,y_test))
         print(accuracy_score(y_pred,y_test))

                      precision    recall  f1-score   support

               False       0.17      0.50      0.25         2
                True       0.98      0.91      0.95        58

            accuracy                           0.90        60
           macro avg       0.57      0.71      0.60        60
        weighted avg       0.95      0.90      0.92        60

         [[ 1  1]
          [ 5 53]]
         0.9

In [53]: from sklearn.metrics import r2_score
         y_true = [3, -0.5, 2, 7]
         y_pred = [2.5, 0.0, 2, 8]
         r2_score(y_true, y_pred)
Out[53]: 0.9486081370449679
```

		Validation Method: Logistic Regression	306 True 52 True 239 True 168 True 136 True 92 False 97 True 275 True 42 True Name: Chance of Admit , dtype: bool In [48]: ```from sklearn.linear_model._logistic import LogisticRegression classifier=LogisticRegression(random_state=0) lr=classifier.fit(x_train,y_train.ravel())``` C:\Users\marve\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1): STOP: TOTAL NO. of ITERATIONS REACHED LIMIT. Increase the number of iterations (max_iter) or scale the data as shown in: https://scikit-learn.org/stable/modules/preprocessing.html Please also refer to the documentation for alternative solver options: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression n_iter_i = _check_optimize_result(In [49]: ```y_pred = lr.predict(x_test)``` In [50]: ```y_pred``` Out[50]: ```array([True, False, True, True, True, True, True, True, False, True])``` ds/UniversityAdmitEligibilityPredictor.ipynb# `rics import accuracy_score, recall_score, roc_auc_score, confusion_matrix`