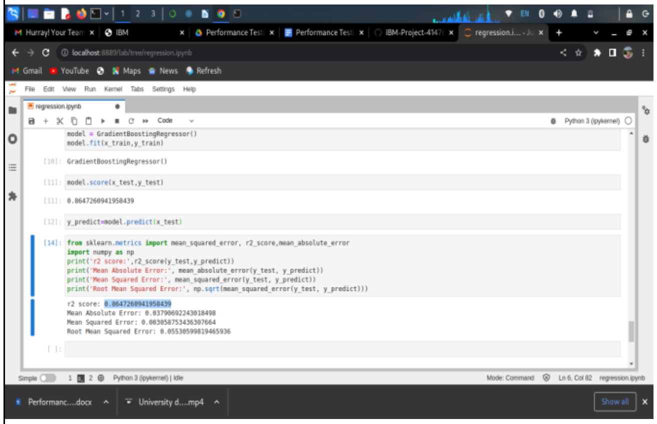
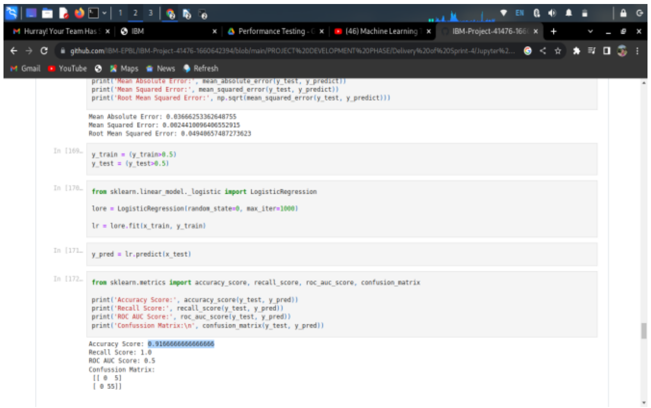
Project Development Phase
Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID40252
Project Name	Project – 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:** MAE –0.03790692243018498, MSE –0.003058753436307664, RMSE – 0.05530599819465936, R2 score – 0.8647260941958439 **Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	
2.	Tune the Model	Hyperparameter Tuning 0.9166666666666666 Validation Method –LinearRegression model	

```
model = GradientBoostingRegressor()
model.fit(x_train,y_train)

[10]: GradientBoostingRegressor()

[11]: model.score(x_test,y_test)

[11]: 0.8647260941958439

[12]: y_predict=model.predict(x_test)

[14]: from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error
      import numpy as np
      print('r2 score:',r2_score(y_test,y_predict))
      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)))

      r2 score: 0.8647260941958439
      Mean Absolute Error: 0.03790692243018498
      Mean Squared Error: 0.003058753436307664
      Root Mean Squared Error: 0.05530599819465936

[ ]:
```

```
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: 0.036662533362648755
Mean Squared Error: 0.0024410096406552915
Root Mean Squared Error: 0.04940657487273623

In [169…  y_train = (y_train>0.5)
          y_test = (y_test>0.5)

In [170…  from sklearn.linear_model._logistic import LogisticRegression

          lore = LogisticRegression(random_state=0, max_iter=1000)

          lr = lore.fit(x_train, y_train)

In [171…  y_pred = lr.predict(x_test)

In [172…  from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix

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

          Accuracy Score: 0.9166666666666666
          Recall Score: 1.0
          ROC AUC Score: 0.5
          Confussion Matrix:
           [[ 0  5]
           [ 0 55]]
```