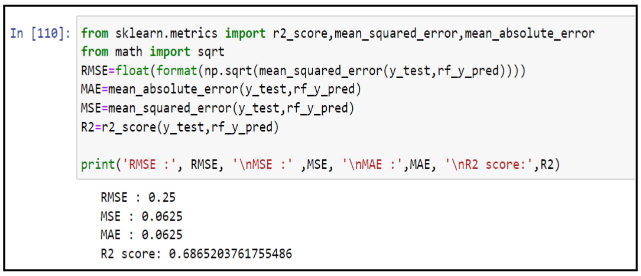
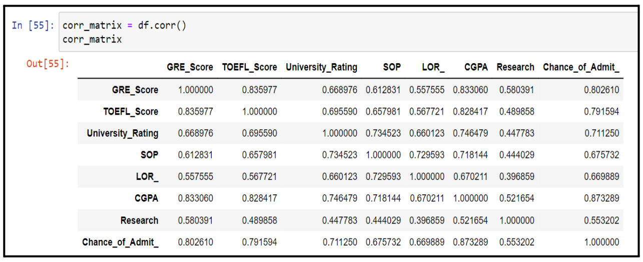
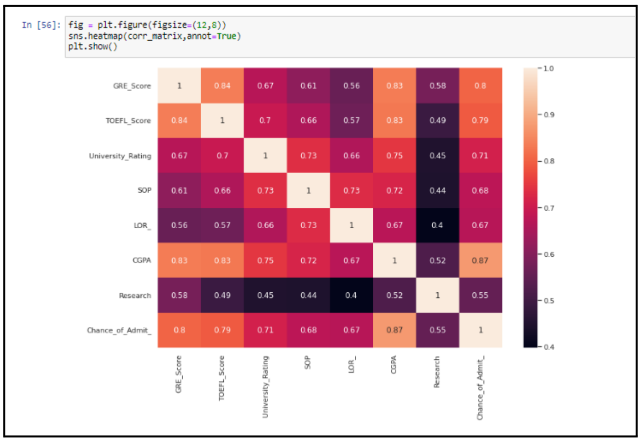
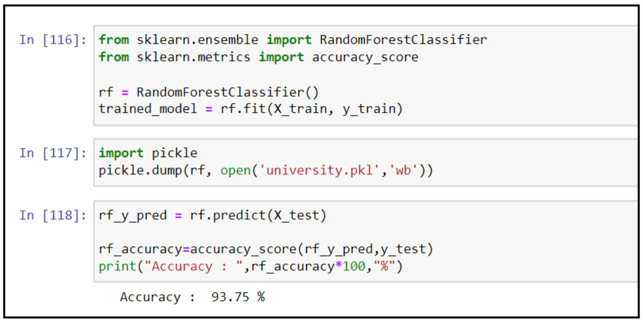
Project Development Phase
Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID21562
Project Name	University Admit Eligibility Predictor
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	   

```
In [110]: from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
          from math import sqrt
          RMSE=float(format(np.sqrt(mean_squared_error(y_test,rf_y_pred))))
          MAE=mean_absolute_error(y_test,rf_y_pred)
          MSE=mean_squared_error(y_test,rf_y_pred)
          R2=r2_score(y_test,rf_y_pred)

          print('RMSE :', RMSE, '\nMSE :' ,MSE, '\nMAE :',MAE, '\nR2 score:',R2)

             RMSE : 0.25
             MSE : 0.0625
             MAE : 0.0625
             R2 score: 0.6865203761755486
```

```
In [55]: corr_matrix = df.corr()
         corr_matrix
```

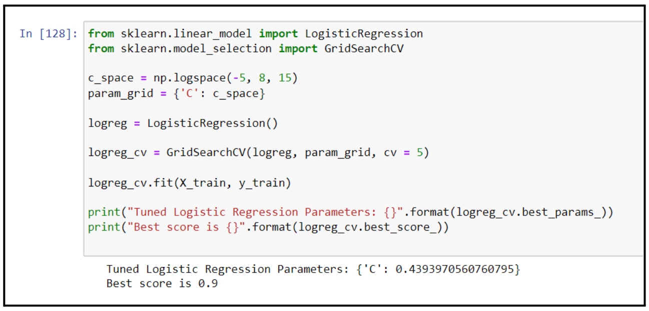
Out[55]:

	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR_	CGPA	Research	Chance_of_Admit_
GRE_Score	1.000000	0.835977	0.668976	0.612831	0.557555	0.833060	0.580391	0.802610
TOEFL_Score	0.835977	1.000000	0.695590	0.657981	0.567721	0.828417	0.489858	0.791594
University_Rating	0.668976	0.695590	1.000000	0.734523	0.660123	0.746479	0.447783	0.711250
SOP	0.612831	0.657981	0.734523	1.000000	0.729593	0.718144	0.444029	0.675732
LOR_	0.557555	0.567721	0.660123	0.729593	1.000000	0.670211	0.396859	0.669889
CGPA	0.833060	0.828417	0.746479	0.718144	0.670211	1.000000	0.521654	0.873289
Research	0.580391	0.489858	0.447783	0.444029	0.396859	0.521654	1.000000	0.553202
Chance_of_Admit_	0.802610	0.791594	0.711250	0.675732	0.669889	0.873289	0.553202	1.000000

```
In [56]: fig = plt.figure(figsize=(12,8))
         sns.heatmap(corr_matrix,annot=True)
         plt.show()
```

GRE_Score 1 0.84 0.67 0.61 0.56 0.83 0.58 0.8
TOEFL_Score 0.84 1 0.7 0.66 0.57 0.83 0.49 0.79
University_Rating 0.67 0.7 1 0.73 0.66 0.75 0.45 0.71
SOP 0.61 0.66 0.73 1 0.73 0.72 0.44 0.68
LOR_ 0.56 0.57 0.66 0.73 1 0.67 0.4 0.67
CGPA 0.83 0.83 0.75 0.72 0.67 1 0.52 0.87
Research 0.58 0.49 0.45 0.44 0.4 0.52 1 0.55
Chance_of_Admit_ 0.8 0.79 0.71 0.68 0.67 0.87 0.55 1
GRE_Score
TOEFL_Score
University_Rating
SOP
LOR_
CGPA
Research
Chance_of_Admit_
1.0
0.9
0.8
0.7
0.6
0.5
0.4

```
In [116]: from sklearn.ensemble import RandomForestClassifier
          from sklearn.metrics import accuracy_score

          rf = RandomForestClassifier()
          trained_model = rf.fit(X_train, y_train)

In [117]: import pickle
          pickle.dump(rf, open('university.pkl','wb'))

In [118]: rf_y_pred = rf.predict(X_test)

          rf_accuracy=accuracy_score(rf_y_pred,y_test)
          print("Accuracy : ",rf_accuracy*100,"%")

             Accuracy :  93.75 %
```

2.	Tune the Model	Hyperparameter Tuning - Validation Method -	

```
In [128]: from sklearn.linear_model import LogisticRegression
          from sklearn.model_selection import GridSearchCV

          c_space = np.logspace(-5, 8, 15)
          param_grid = {'C': c_space}

          logreg = LogisticRegression()

          logreg_cv = GridSearchCV(logreg, param_grid, cv = 5)

          logreg_cv.fit(X_train, y_train)

          print("Tuned Logistic Regression Parameters: {}".format(logreg_cv.best_params_))
          print("Best score is {}".format(logreg_cv.best_score_))

            Tuned Logistic Regression Parameters: {'C': 0.4393970560760795}
            Best score is 0.9
```