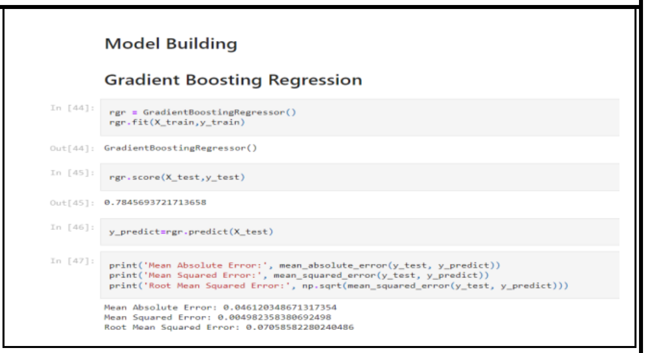
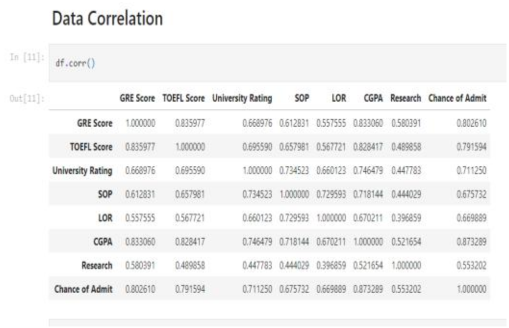
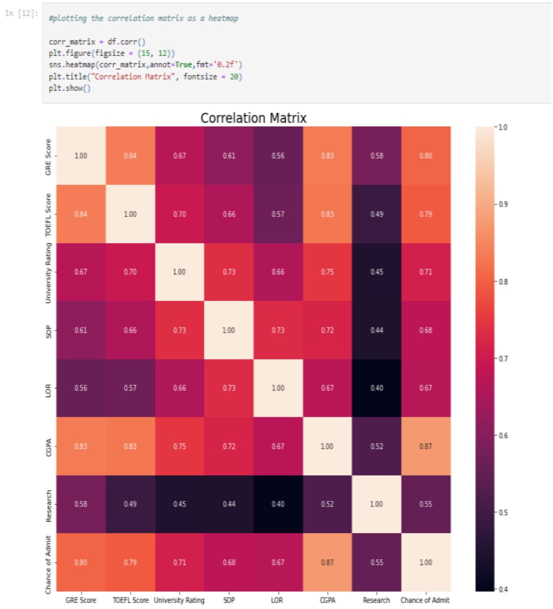
PERFORMANCE METRICS

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

PERFORMANCE METRICS:

S.No.	PARAMETER	VALUES	SCREENSHOT
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuracy Score- & Classification Report -	  

Model Building

Gradient Boosting Regression

```
In [44]: rgr = GradientBoostingRegressor()
         rgr.fit(X_train,y_train)

Out[44]: GradientBoostingRegressor()

In [45]: rgr.score(X_test,y_test)

Out[45]: 0.7845693721713658

In [46]: y_predict=rgr.predict(X_test)

In [47]: 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: 0.046120348671317354
         Mean Squared Error: 0.004982358380692498
         Root Mean Squared Error: 0.07058582280240486
```

Data Correlation

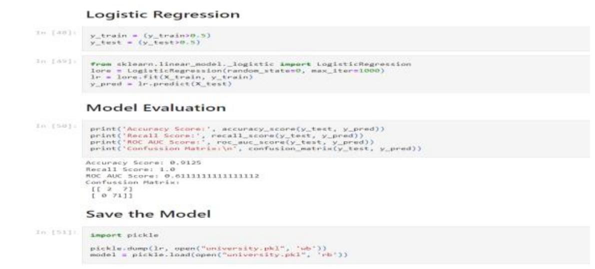
```
In [11]: df.corr()

Out[11]:
```

	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 [12]: #plotting the correlation matrix as a heatmap

         corr_matrix = df.corr()
         plt.figure(figsize = (15, 12))
         sns.heatmap(corr_matrix,annot=True,fmt='0.2f')
         plt.title("Correlation Matrix", fontsize = 20)
         plt.show()
```

2.	**Tune the Model**	Hyper parameter Tuning - Validation Method -	

Logistic Regression

```
In [48]: y_train = (y_train>0.5)
         y_test = (y_test>0.5)
```

```
In [49]: from sklearn.linear_model._logistic import LogisticRegression
         lore = LogisticRegression(random_state=0, max_iter=1000)
         lr = lore.fit(X_train, y_train)
         y_pred = lr.predict(X_test)
```

Model Evaluation

```
In [50]: 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: 0.9125
         Recall Score: 1.0
         ROC AUC Score: 0.6111111111111112
         Confussion Matrix:
          [[ 2  7]
          [ 0 71]]
```

Save the Model

```
In [51]: import pickle

         pickle.dump(lr, open("university.pkl", 'wb'))
         model = pickle.load(open("university.pkl", 'rb'))
```