Predicting Chance of Admit using Linear Regression, Decison Tree,Random Forest

ALLU SAI PRUDHVI ---- B160865CS
CH . JAGADEESH -----B160547CS
M. Sharath -----B160855CS
Pullcallu Rahul -----B160607CS

Abstract

The number of students wanting to pursue higher education abroad is increasing rapidly every year, especially in the US-based universities, and they find it difficult to find the best university. And so, this paper helps predict Indian students eligibility to be admitted in the best university based on the attributes such as GRE score,TOEFL score,CGPA,research papers etc. The possibility of their chance of admit is then calculated using their scores . This project is done using machine learning techniques.This project is done using machine learning techniques namely decision tree, linear regresion and randomm forest algorithms to predict the output that helps the students to admit to the best university and also helps the student find the possibility to admit to other universities based on their scores. Out of the three algorithms linear regression showed higher accuracy.

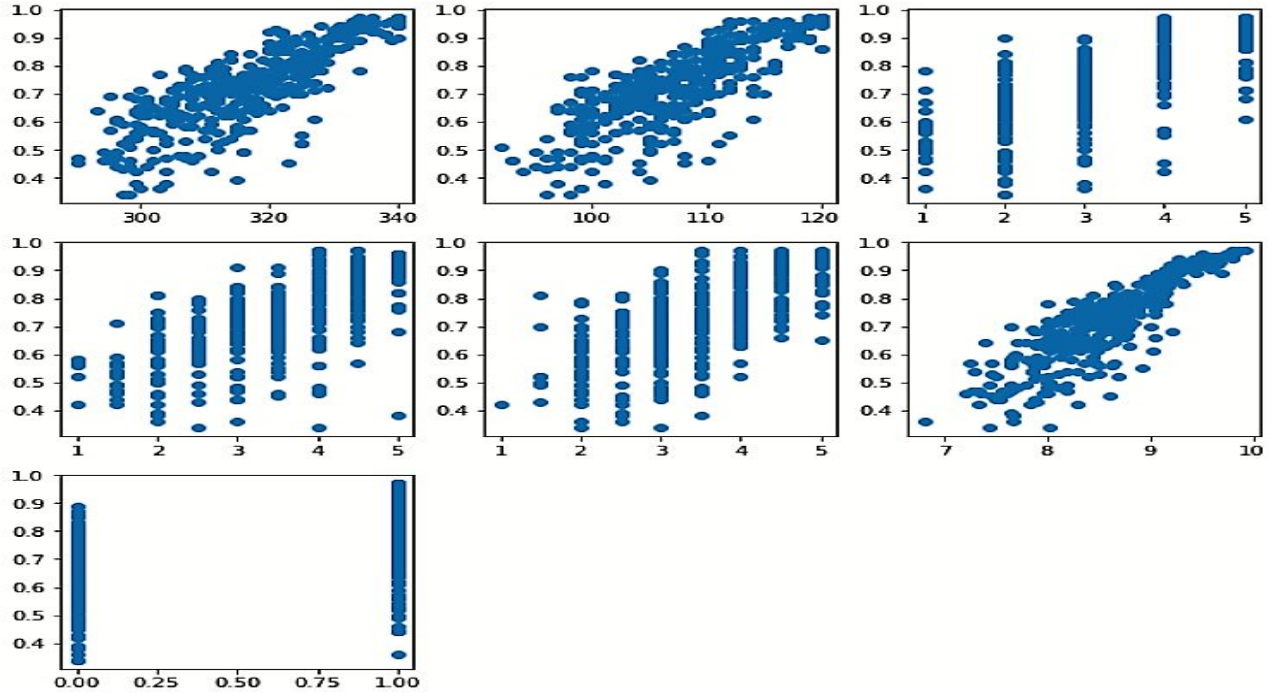
DATASET

We performed analysis on kaggle dataset. The dataset has 400 rows and 9 column. Chance of admit is he feature which we need to predictt.

Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4.0	4.5	8.87	1	0.76
3	316	104	3	3.0	3.5	8.00	1	0.72
4	322	110	3	3.5	2.5	8.67	1	0.80
5	314	103	2	2.0	3.0	8.21	0	0.65

Data Analysis

Plotting between dependant and independant variables(features) to obtain the relationship between the variables.



Preprocessing

Before training the model , preprocessing the dataset is must and should for every model to get a good accuracy.some of the techniques , which we checked are:

- Removing NULL Values.
- Checking for Outliers
- Normalizing the attributes.

1. Checking for NULL values

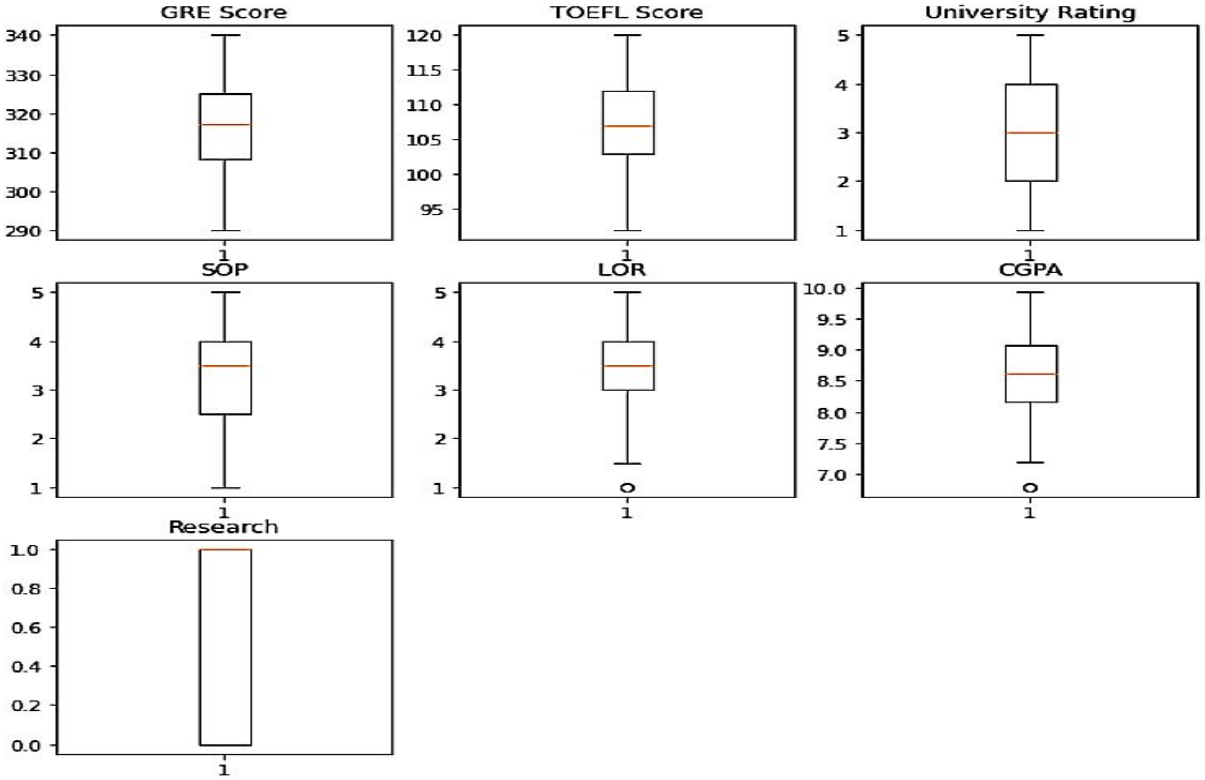
There are no values found in this dataset.

```
data.isnull().sum()
```

```
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

2. Outliers

It is observed that there are no outliers for all attributes.



3. Normalizing values

To give equal importance to each and every attribute , we normalize the values using StandardScaler from Preprocessing package.

```
from sklearn.preprocessing import StandardScaler

data=sc.fit_transform(data)
```

+ Code + Markdown

ALGORITHMS

In this paper we are going to use three algorithms to predict the output column.

1. LINEAR REGRESSION MODEL
2. DECISION TREE MODEL
3. RANDOM FOREST REGRESSOR MODEL

1. Linear Regression

Linear Regression is an algorithm of machine learning, based on supervised learning. It performs regression. Linear regression executes the task of predicting a dependent variable value (y) based on an independent variable (x) given. So this technique of regression finds a linear relation between x(input) and y(output).

```
from sklearn.linear_model import LinearRegression

M1=LinearRegression()
M1.fit(x_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
y_pred= M1.predict(x_test)
```

Decision Tree

Decision tree algorithm falls under the Supervised Learning category. They can be used to solve regression problems as well as classification problems. Decision tree uses tree representation to solve the problem where each leaf node corresponds to a class label and attributes on the tree's internal node are represented. Using the decision tree we may represent any boolean function on discrete attributes.

```
from sklearn import tree

m2=tree.DecisionTreeRegressor(criterion='mse',max_depth=4,random_state=1)
m2.fit(x_train,y_train)
```

```
DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=4,
                      max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort='deprecated',
                      random_state=1, splitter='best')
```

Random Forest Regressor

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

```
from sklearn.ensemble import RandomForestRegressor

m3=RandomForestRegressor(max_depth=8,n_estimators=300,random_state=1)
m3.fit(x_train,y_train)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=8, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=300, n_jobs=None, oob_score=False,
                      random_state=1, verbose=0, warm_start=False)
```

Results

```
print("LINEAR REGRESSION")
errors(y_test,y_pred)
```

```
LINEAR REGRESSION
MAE is 0.04495779899577675
MSE is 0.004442679729994745
RMSE is 0.06665342999422269
R2 score is 0.80790436770201
```

```
print("DECISION TREE")

errors(y_test,y_pred)
```

```
DECISION TREE
MAE is 0.05801653126226336
MSE is 0.006768528556293033
RMSE is 0.08227106755289512
R2 score is 0.7073377214275022
```

```
print("RANDOM FOREST REGRESSOR")
errors(y_test,y_pred)
```

```
RANDOM FOREST REGRESSOR
MAE is 0.05801653126226336
MSE is 0.006768528556293033
RMSE is 0.08227106755289512
R2 score is 0.7073377214275022
```

Conclusion

In this paper we focused more on predicting the MS Graduate admission dataset using Linear regression,Decision tree and Random forest.Applied many pre-processing techniques to improve the accuracy and to reduce the errors.It is observed that principal component analysis is not showing any significant change.Among all, linear regression showed better performance.While calculating feature_importances in Random Forest Regressor, CGPA attribute showed higher value compared to other attributes.