PROJECT REPORT
ON
ADMISSION PREDICTION

Submitted By :- (Group-3,Project-1)

Name	Email	College
Tirth Patel	tirthpatel23t@gmail.com	Ahmedabad Institute Of Technology
Aniket Sharad Khatode	aniketak03@gmail.com	Amrutvahini College Of Engineering
Visista	Visista17@gmail.com	Annamacharya Institute Of Technology And Sciences

Under The Guidance Of :-

Gurvansh Singh

M.Tech

Knowledge Solutions India

Abstract

Artificial Intelligence has become one of the most emerging field recently. Machine Learning is one of the subset of Artificial Intelligence. There are majorly three types of Machine Learning : Supervised Machine Learning, Unsupervised Machine Learning, Reinforcement Machine Learning. This project is based on Supervised Machine Learning. In Supervised Machine Learning, inputs and outputs both are known and on the basis of them predictions are done. Examples of Supervised Machine Learning are training data or voice or image to predict output. It is also used in forecasting or recommendation system.

Content

1. Abstract…………………………………………………………..…2
2. Content…………………………………………………………...3
3. Chapters :-
    - i.Introduction………………………………………………4
    - ii.Libraries…………………………………………………5
    - iii.Mathematics……………………………………………...6
    - iv.Dataset……………………………………………………7
    - v.EDA & Data-Preprocessing……………………………...8
    - vi.Models……………………………………………………...10
    - vii.Accuracy Checking Methods…………………………….14
4. Graphs…………………………………………………………...……15
5. Conclusion…...…………………………………………………….....18
6. References……………………………………………………………19

Chapters

i. Introduction :-

This project is based on Supervised Machine Learning. So,we require both inputs and outputs. Inputs are features (independent variables). They are generally indicated by ‘x’. Outputs are dependent variables. They are generally indicated by ‘y’. In this dataset, features are GRE scrores, TOEFL scores, University Rating, Statement of Purpose (SOP), Recommendation Strength (LOR), CGPA, Research. Independent variable is Chance of Admit. In this project, ML models to predict the ‘Chance of Admit’ with minimum MSE and RMSE and maximum R-Square score are built. Following models are built :

1) Multiple Linear Regression
2) Random Forest Regression
3) Multiple Linear Regression with Principal Component Analysis
4) Random Forest Regression with Principal Component Analysis

ii. Libraries :-

The most important step is importing required libraries so that every necessary modules can be used in code conveniently. ‘import’ keyword is used to import libraries. Libraries included in this project are :

1) Numpy
2) Pandas
3) Matplotlib
4) Scikit-learn
5) Scipy
6) Mlxtend
7) Statsmodels
8) Xlrd

iii. Mathematics :-

Concepts of mean, median, mode are used in data-preprocessing. In principal component analysis, concepts of linear algebra like variance, standard deviation, eigen vector etc. are used.

Mean : Mean is the average of the values.

Median : Median is the middle value of set which is already sorted. If there two median, then mean of two values is median.

Mode : Mode is the most repeated value

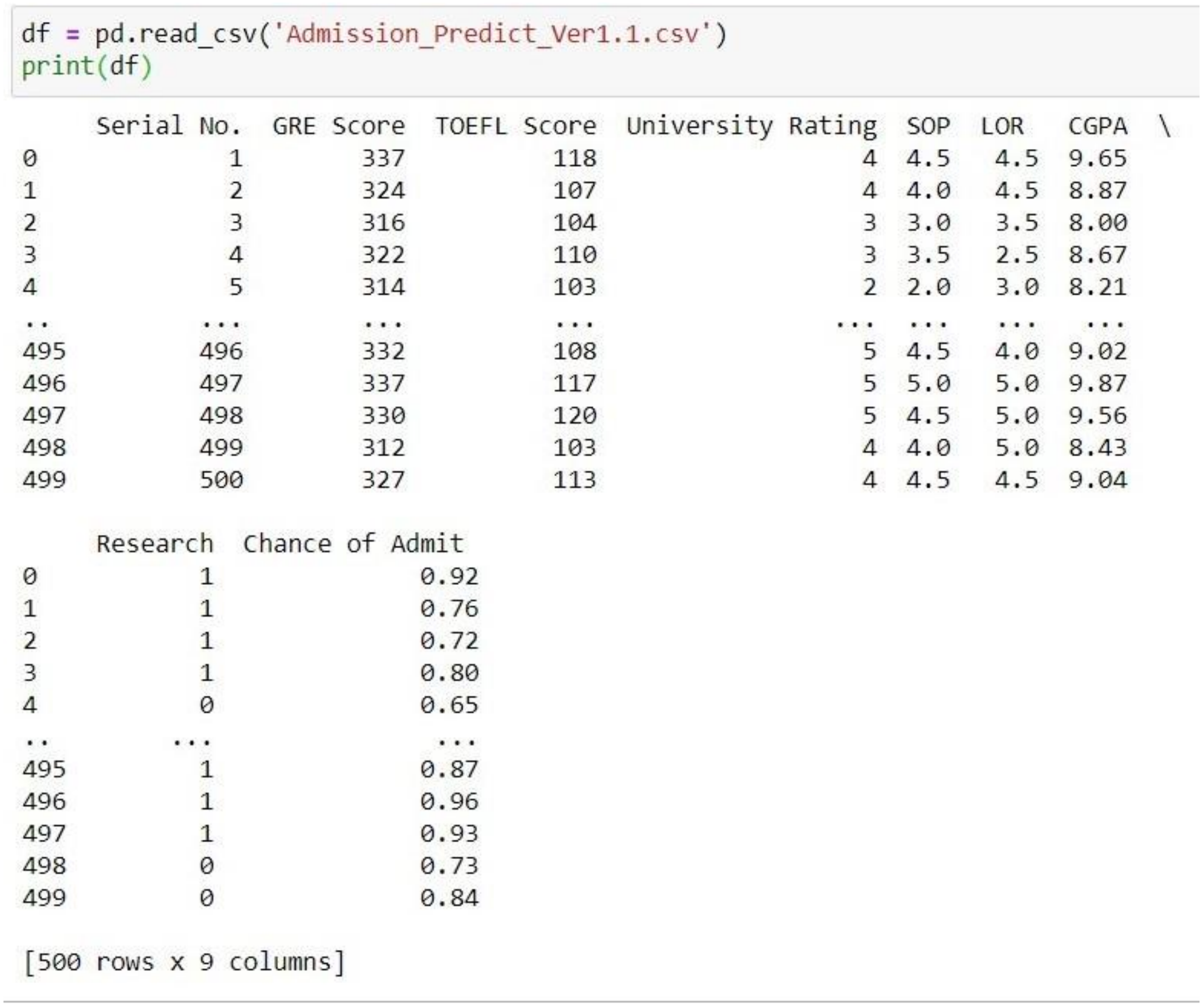
Standard Deviation : Standard Deviation shows how spread out the values in Dataset.

Variance : Variance is square of Standard Deviation.

Eigenvector : An Eigenvector is vector whose direction remains unchanged when a linear transformation is applied to it.

iv. Dataset :-

After importing libraries, second step is to read dataset. Pandas library object is used to read dataset. In this dataset, there are GRE scrores, TOEFL scores, University Rating, Statement of Purpose (SOP), Recommendation Strength (LOR), CGPA, Research, Chance of Admit columns. Dimension is 500 rows and 9 columns.



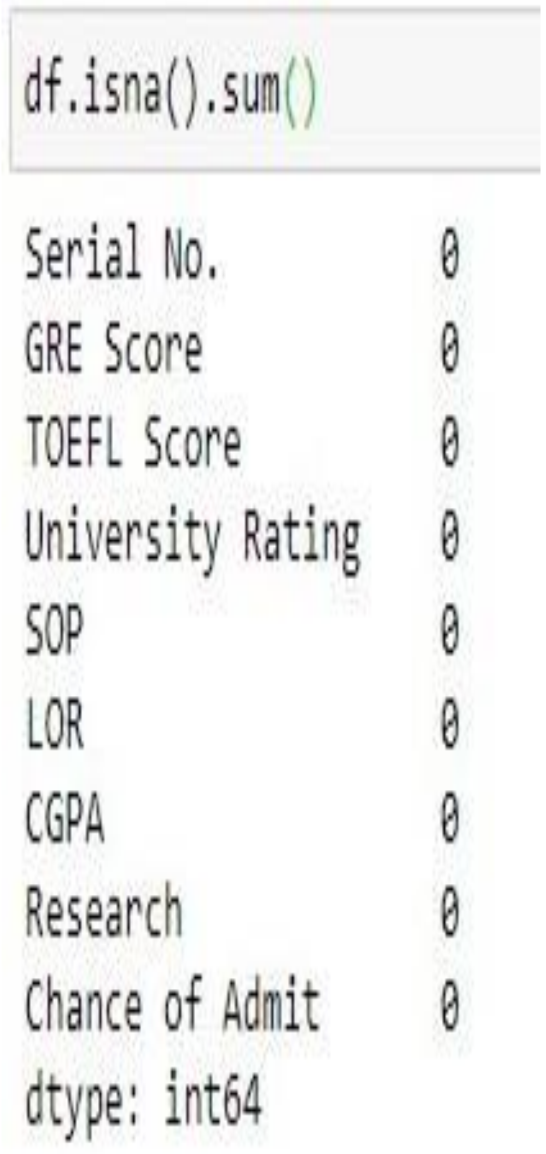
```
df = pd.read_csv('Admission_Predict_Ver1.1.csv')
print(df)
```

```
     Serial No.  GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  \
0             1        337          118                  4  4.5   4.5  9.65
1             2        324          107                  4  4.0   4.5  8.87
2             3        316          104                  3  3.0   3.5  8.00
3             4        322          110                  3  3.5   2.5  8.67
4             5        314          103                  2  2.0   3.0  8.21
..          ...        ...          ...                ...  ...   ...   ...
495         496        332          108                  5  4.5   4.0  9.02
496         497        337          117                  5  5.0   5.0  9.87
497         498        330          120                  5  4.5   5.0  9.56
498         499        312          103                  4  4.0   5.0  8.43
499         500        327          113                  4  4.5   4.5  9.04

     Research  Chance of Admit
0           1             0.92
1           1             0.76
2           1             0.72
3           1             0.80
4           0             0.65
..        ...              ...
495         1             0.87
496         1             0.96
497         1             0.93
498         0             0.73
499         0             0.84

[500 rows x 9 columns]
```

Admission_Predict_ver1.1.csv

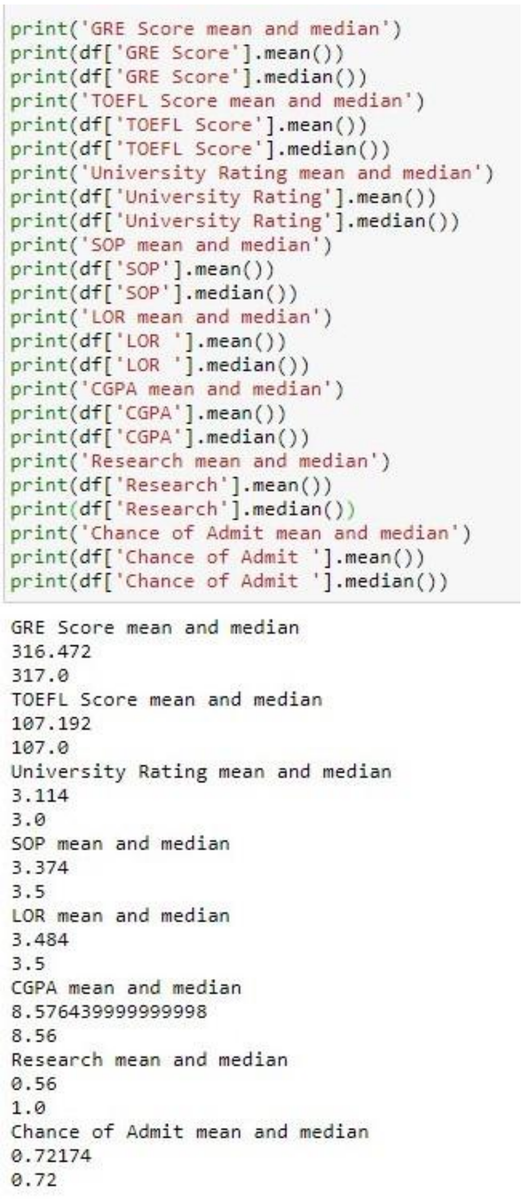
v. EDA & Data-Preprocessing :-

Exploratory Data Analysis & Data-Preprocessing are necessary steps before creating any model. After EDA on this dataset, it is found that there are no null values, outliers, missing value.



```
df.isna().sum()
```

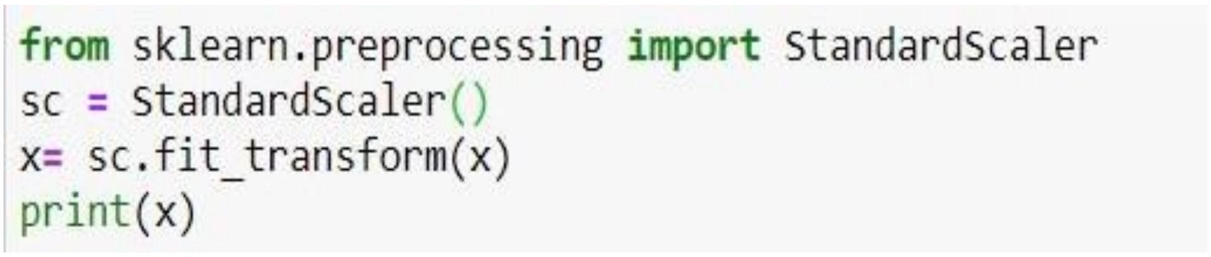
```
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```



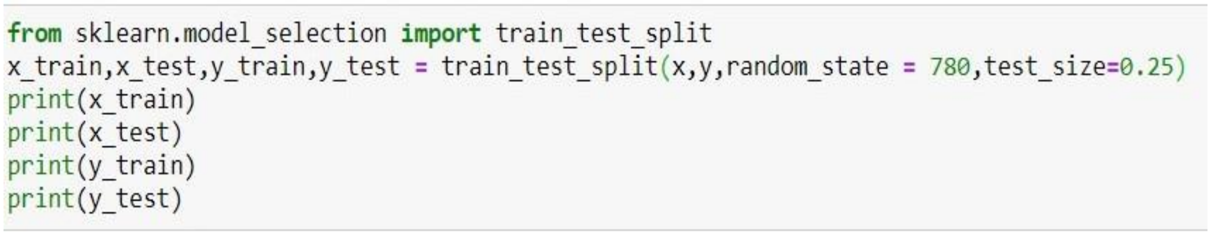
```
print('GRE Score mean and median')
print(df['GRE Score'].mean())
print(df['GRE Score'].median())
print('TOEFL Score mean and median')
print(df['TOEFL Score'].mean())
print(df['TOEFL Score'].median())
print('University Rating mean and median')
print(df['University Rating'].mean())
print(df['University Rating'].median())
print('SOP mean and median')
print(df['SOP'].mean())
print(df['SOP'].median())
print('LOR mean and median')
print(df['LOR '].mean())
print(df['LOR '].median())
print('CGPA mean and median')
print(df['CGPA'].mean())
print(df['CGPA'].median())
print('Research mean and median')
print(df['Research'].mean())
print(df['Research'].median())
print('Chance of Admit mean and median')
print(df['Chance of Admit '].mean())
print(df['Chance of Admit '].median())
```

```
GRE Score mean and median
316.472
317.0
TOEFL Score mean and median
107.192
107.0
University Rating mean and median
3.114
3.0
SOP mean and median
3.374
3.5
LOR mean and median
3.484
3.5
CGPA mean and median
8.576439999999998
8.56
Research mean and median
0.56
1.0
Chance of Admit mean and median
0.72174
0.72
```

In Data-Preprocessing, x (features) and y (output) are extracted from dataset. There is no categorical data so encoding is not required. As there is large difference in values, scaling of features is done. After that using train_test_split(), x and y are splitted into train set (x_train, y_train) and test set (x_test, y_test) for training and testing respectively.



```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x= sc.fit_transform(x)
print(x)
```



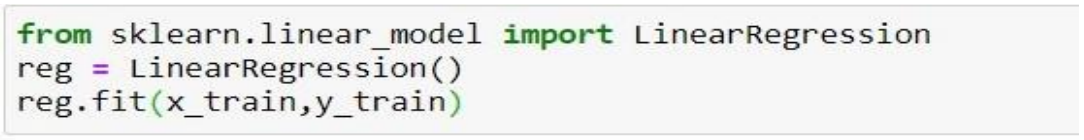
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state = 780,test_size=0.25)
print(x_train)
print(x_test)
print(y_train)
print(y_test)
```

test_size and train_size can be given exclusively or bydefault it is 0.25 (25% of dataset). random_state distributes dataset randomly into train and test according to the seed value.

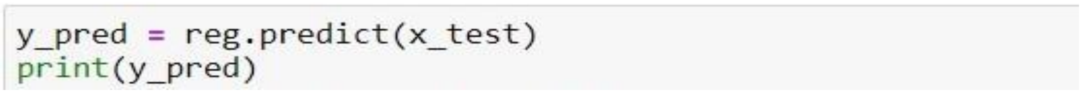
vi. Models :-

1) Multiple Linear Regression

The regression is used to find the relationship between variables. In Machine Learning and in Statistical modelling, that relationship is used to predict the outcome of future events. Linear Regression uses the relationship between the data-points to draw a straight line through all them. Multiple Regression is like Linear Regression, but with more than one independent value or variables. Output will always be one. ‘x_train’ and ‘y_train’ are trained by fitting them in model using the object of LinearRegression class. Predict() takes ‘x_train’ as argument to predict the output (y_pred). This predicted output (y_pred) can be compared with actual output (y_test) to check accuracy. Plotting more than 3 dimension is not possible so Automatic Backward Elimination is used. It removes the least significant features which affects less in predicting output. After that again training and modelling can be done for predicting output. scatter() and plot() are used for plotting graph.



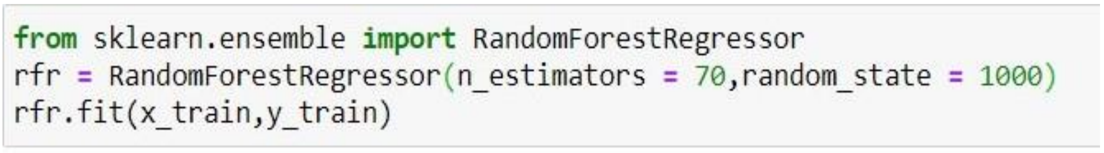
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(x_train,y_train)
```



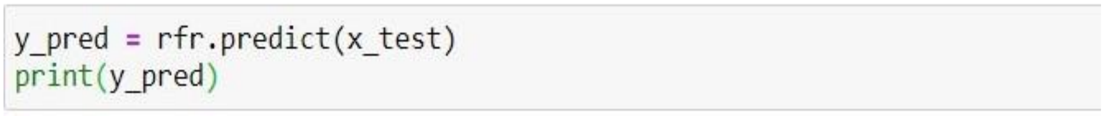
```
y_pred = reg.predict(x_test)
print(y_pred)
```

2) Random Forest Regression

Random Forest is a popular Machine Learning Algorithm. It can be used for both Classification and Regression problems. Here it is used as Random Forest Regression as it is a regression problem. It is based on the concept of ensemble learning. Multiple algorithms are combined to solve complex problem & to improve the performance of the model. It takes prediction from each tree and based on the average, predicts the final output. ‘x_train’ and ‘y_train’ are trained by fitting them in model using the object of RandomForestRegressor class. Predict() takes ‘x_train’ as argument to predict the output (y_pred). This predicted output (y_pred) can be compared with actual output (y_test) to check accuracy. The advantage of RF is that it takes less training time compared to other algorithm. It predicts output with high accuracy even for large dataset. scatter() and plot() are used for plotting graph.



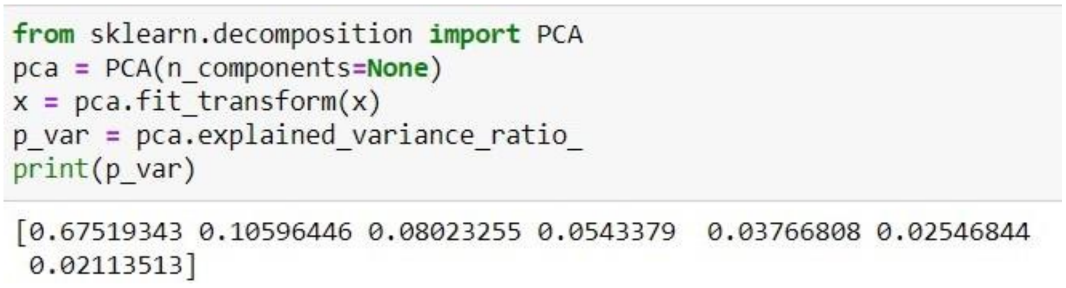
```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor(n_estimators = 70,random_state = 1000)
rfr.fit(x_train,y_train)
```



```
y_pred = rfr.predict(x_test)
print(y_pred)
```

3) Multiple Linear Regression with Principal Component Analysis

Plotting more than 3 dimension is not possible and moreover working with multiple features can be inefficient. Principal Component Analysis is most popular dimensionality reduction algorithm. It detect correlation and covariance between features (as it is unsupervised algorithm it considers only input). The goal is to reduce the dimensions of a d-dimensional dataset by projecting it onto a (k)-dimensional subspace (where k<d) in order to increase the computational efficiency while retaining most of the information. Using the object of PCA class, fit_transform() takes x as argument and fit x in model and transform x into required features. After PCA, only significant features are left and these features are splitted into train set and test set. Now modeliing procedure is similar to Multiple Linear Regression as described above.

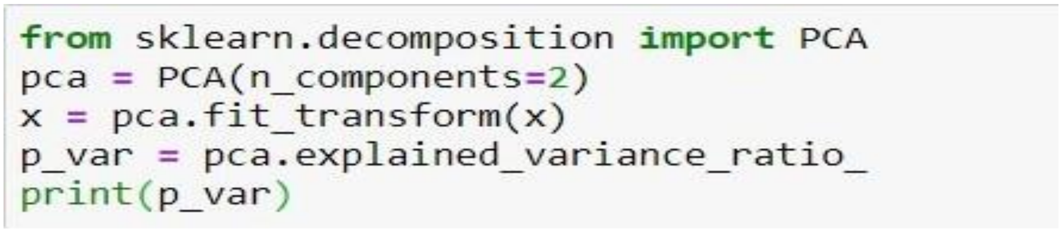


```
from sklearn.decomposition import PCA
pca = PCA(n_components=None)
x = pca.fit_transform(x)
p_var = pca.explained_variance_ratio_
print(p_var)
```

```
[0.67519343 0.10596446 0.08023255 0.0543379  0.03766808 0.02546844
 0.02113513]
```

Initially, n_components=none is taken because eigen values are not known and also eigen vectors to create w matrix, are not known.

‘p_var’ is calculated which describes percentage variance of each column. On the basis of it eigen vectors can be considered. Here they are first two so, n_components=2 is taken.



```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
x = pca.fit_transform(x)
p_var = pca.explained_variance_ratio_
print(p_var)
```

4) Random Forest Regression with Principal Component Analysis

Principal Compoenet Analysis is done on dataset by performing EDA and Pre-processing. As described above, dimensionality reduction is done similarly. After splitting x (new features) and y into train set and test set, both are trained and fitted in random forest regressor model and ouput is predicted in a similar way as mentioned in 3) RFR.

vii. Accuracy Checking Methods :-

MAE : Mean Absolute Error represents the difference between original values & predicted values extracted by averaged difference.

MSE : Mean Squared Error represents the difference between original values & predicted values extracted by squared the average distance.

RMSE : Root Mean Squared Error is the error rate by the square root of MSE.

R-Squared : It represents the coefficient of how well the values fit compared to the original values. Its value lies between 0 & 1. 1 indicates model fits perfectly to dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2}$$

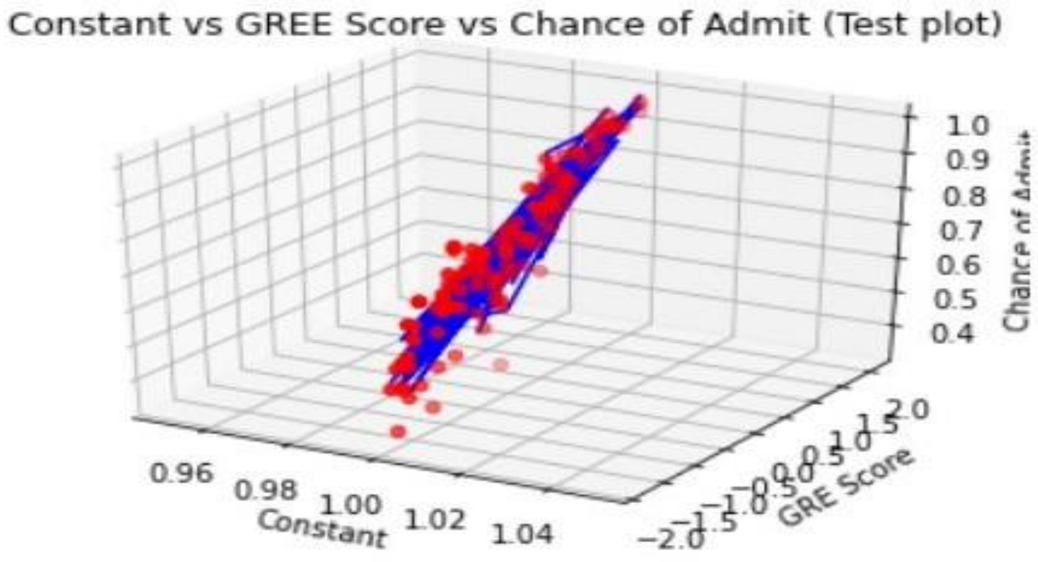
$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

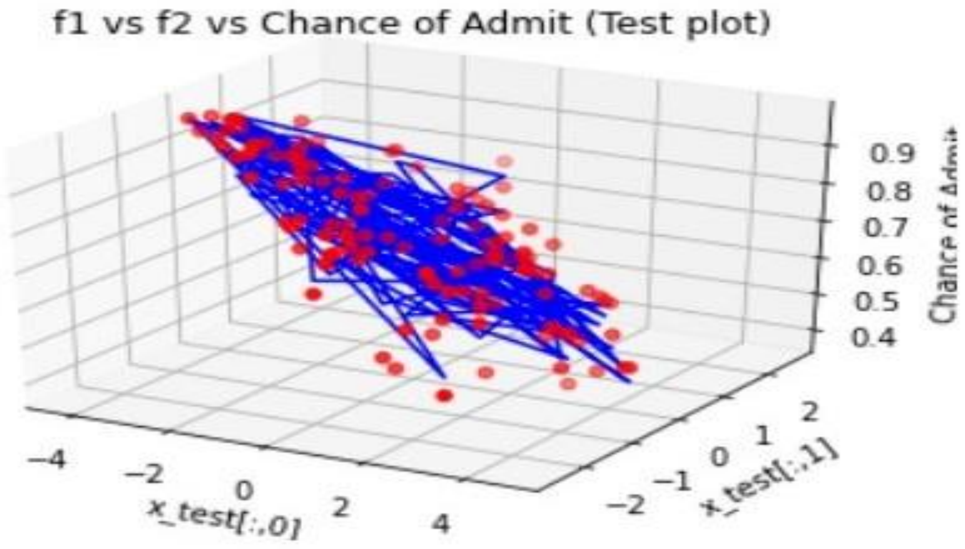
$\hat{y}$ $-$ *predicted value of y*

$\bar{y}$ $-$ *mean value of y*

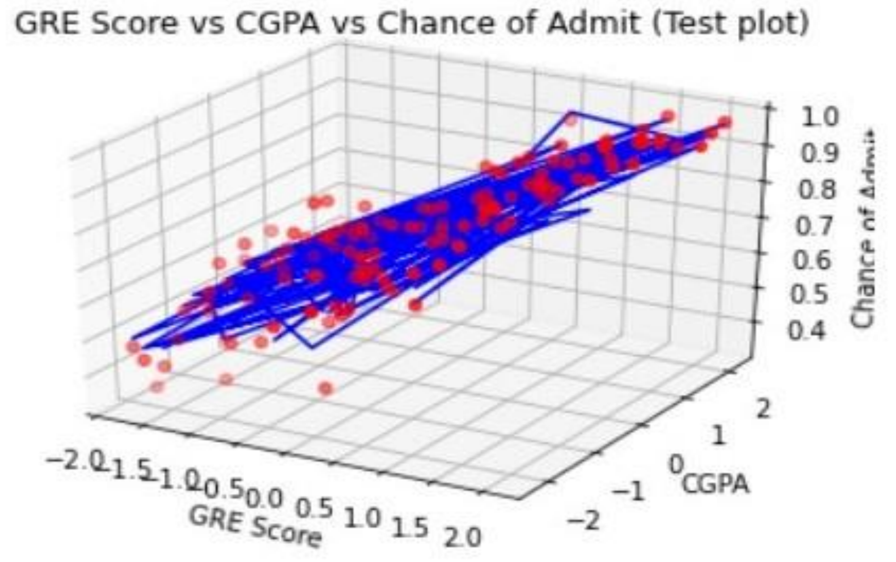
Graphs



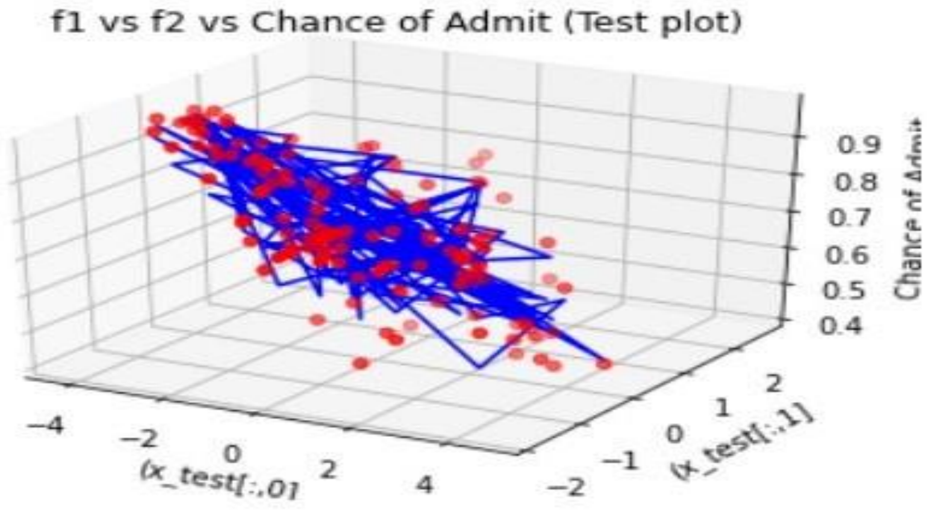
MLR



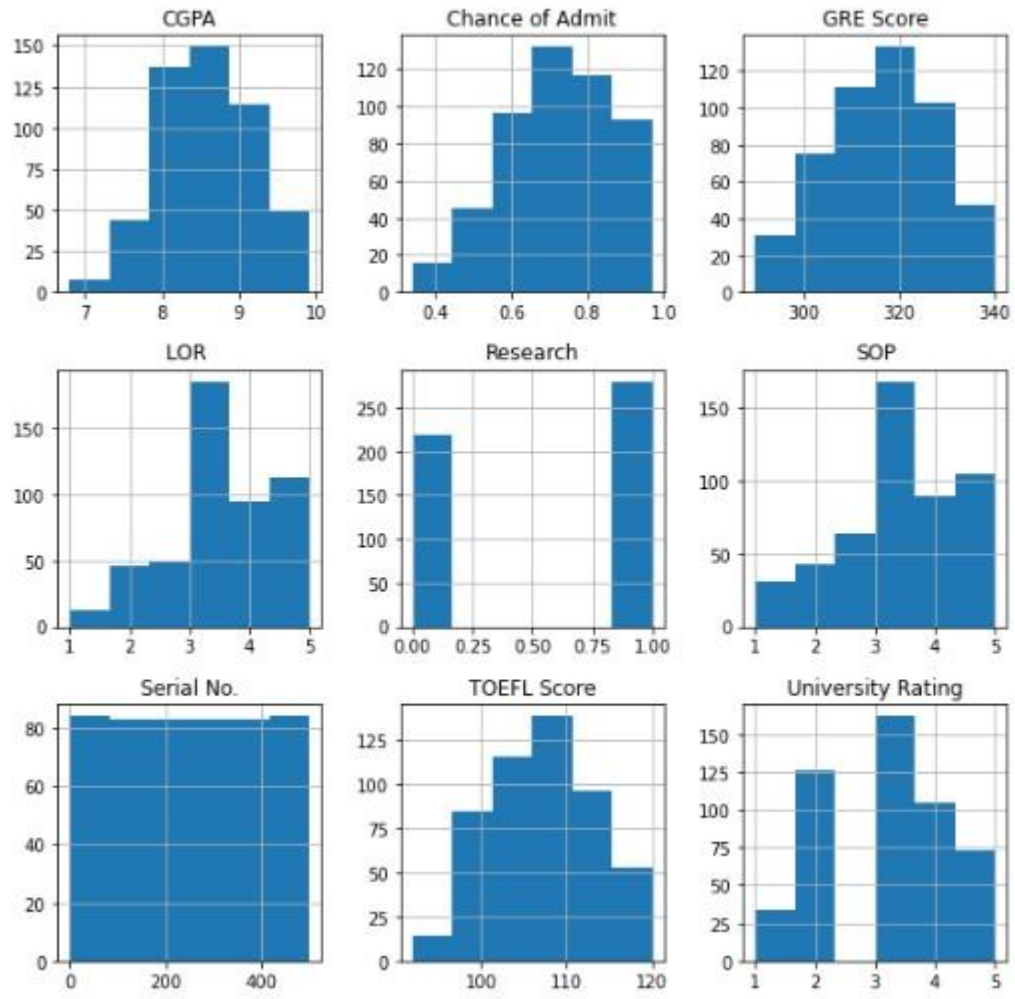
MLR with PCA



RFR



RFR with PCA


CGPA
Chance of Admit
GRE Score
LOR
Research
SOP
Serial No.
TOEFL Score
University Rating

Conclusion

From the below given table, we can conclude that Mean Squared Error of Multiple Linear Regression is least i.e., **0.0025** which can be considered as good among all other models. Root Mean Squared Error of Multiple Linear Regression is least i.e., **0.0500** which can be considered good among all other models. R-Squared Score of Multiple Linear Regression is Highest i.e., **88%** which can be said that model best fits to dataset. Random Forest Regression with Principal Component Analysis has highest MSE and RMSE which is not good for model. Moreover it has least R2_score i.e., 0.810 among all models, which is shows model does not fit well in dataset.

ACM\Model	MLR	MLR_PCA	RFR	RFR_PCA
MSE	0.0025	0.0032	0.0028	0.0040
RMSE	0.0500	0.0569	0.0533	0.0637
R2_Score	0.883	0.837	0.867	0.810

References

https://www.mentorrbuddy.com/home

https://plotly.com/python/v3/ipython-notebooks/principal-component-analysis/

https://www.datatechnotes.com/2019/02/regression-model-accuracy-mae-mse-rmse.html#:~:text=The%20MSE%2C%20MAE%2C%20RMSE%2C,difference%20over%20the%20data%20set.

https://www.javatpoint.com/