INDIRA GANDHI INSTITUTE OF TECHNOLOGY
Dhenkanal, Sarang-759146

A PROJECT REPORT
ON

"Graduate Admission Prediction Using ML"

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

PIYUS PRABHANJANS	**387030**
PRITAM MISHRA	**387036**
R ARBIND PANDA	**387039**
JYOTI JENA	**387017**

Under the Guidance of
DR SRINIVAS SETHI
Asst. Proff,Department of CSE,IGIT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IGIT SARANG
DHENKANAL, SARANGA - 759146

2022-2023

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Dissertation entitled "GRADUATE ADMISSION PREDICTION USING ML" in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science Engineering is an authentic record of our own work carried out during a period from March,2022 to May,2022 under the guidance of Dr Srinivas Sethi, Asst Professor, CSEA Department,IGIT Sarang for Skill Project Work during the period of 6th Semester.

Pritam Mishra

Piyus Prabhanjans

R Arbind Panda

Jyoti Jena

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

DATE: 13.08.2022

Signature Of Supervisor

Dr. Srinivas Sethi

Asst. Professor, CSEA Department

IGIT , Sarang

ACKNOWLEDGEMENT

First of all, we express our gratitude to the almighty who blessed us with the zeal and enthusiasm to complete this research work successfully. We are extremely thankful to our supervisor Dr Srinivas Sethi, Asst Professor, Computer Science and Engineering Department, IGIT Sarang for his motivation and tireless efforts to help us to get deep knowledge of the research area and supporting us throughout the life cycle of our B.Tech. dissertation work.

We are also thankful to Dr. Sasmita Mishra, HOD of Computer Science and Engineering Department for her fruitful guidance through the early years of chaos and confusions. We wish to thank the faculty members and supporting staff of Computer Science and Engineering department for their full support and heartiest co-operation.

Pritam Mishra.

Piyus Prabhanjans.

Jyoti Jena.

R Arvind Panda.

CONTENTS

ABSTRACT

Data is the most important asset which is further processed to produce useful information. Data science and Data analytics techniques are widely used to generate useful patterns helpful for better understanding and this project is for prediction of graduate admissions from an Indian perspective by creating ML models with minimum MSE, RMSE and maximum R- square score. However, the manual process of record checking is time consuming and error prone due to the complexity of data we use combination of linear and non-linear machine learning algorithms like Linear Regression, Multiple Linear Regression and Random Forest Regression and the models built in this work are predicting the likelihood of a student taking up the admission into any university based on the student data collected any administrative officials can use this kind of an application to explore and analyze the patterns that are affecting the student admission and come up with new strategies to improve admission. **We have** created ML models to predict the 'Chance of Admit' with minimum MSE and RMSE and maximum R-Square score

By Building the following models:

1. Multiple linear Regressor (MLR)
2. Random Forest Regressor (RFR)
3. MLR with PCA
4. RFR with PCA

Finally plotted the actual and predicted values for all three models

LIST OF GRAPHS

CHAPTERS

INTRODUCTION

In this project, we will be using the admission _ predict dataset in csv format to predict the chances of students getting admission by a university based on several academic performance measurement. To yield the most accurate result, we will be going through several steps such as data preprocessing, t-test, cross validation, model selection, etc. to train a machine learning model, make prediction and measure its performance.

Machine learning models such as Multiple Linear Regressor, Random Forest Regressor and Multiple Linear Regressor with Principal Component Analysis with minimum MSE and RMSE and maximum R-Square score are created to predict the chance of admit

The dataset contains several parameters:

1. GRE Scores (out of 340)
2. TOEFL Scores (out of 120)
3. University Rating (out of 5)
4. Statement of Purpose and Letter of Recommendation Strength (out of 5)
5. Undergraduate GPA (out of 10)
6. Research Experience (either 0 or 1)
7. Chance of Admit (ranging from 0 to 1).

The last column is the response variable, which is a continuous value between 0 and 1, indicating the chances of getting admission.

Multiple Linear Regressor (MLR)

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable'

Moreover, Multiple Linear Regression is an extension of Simple Linear regression as it takes more than one predictor variable to predict the response variable

The libraries used in this model are:-

NumPy
Pandas
Scikit-Learn
Seaborn
Matplotlib

LinearRegression

mean_squared_error

train_test_split

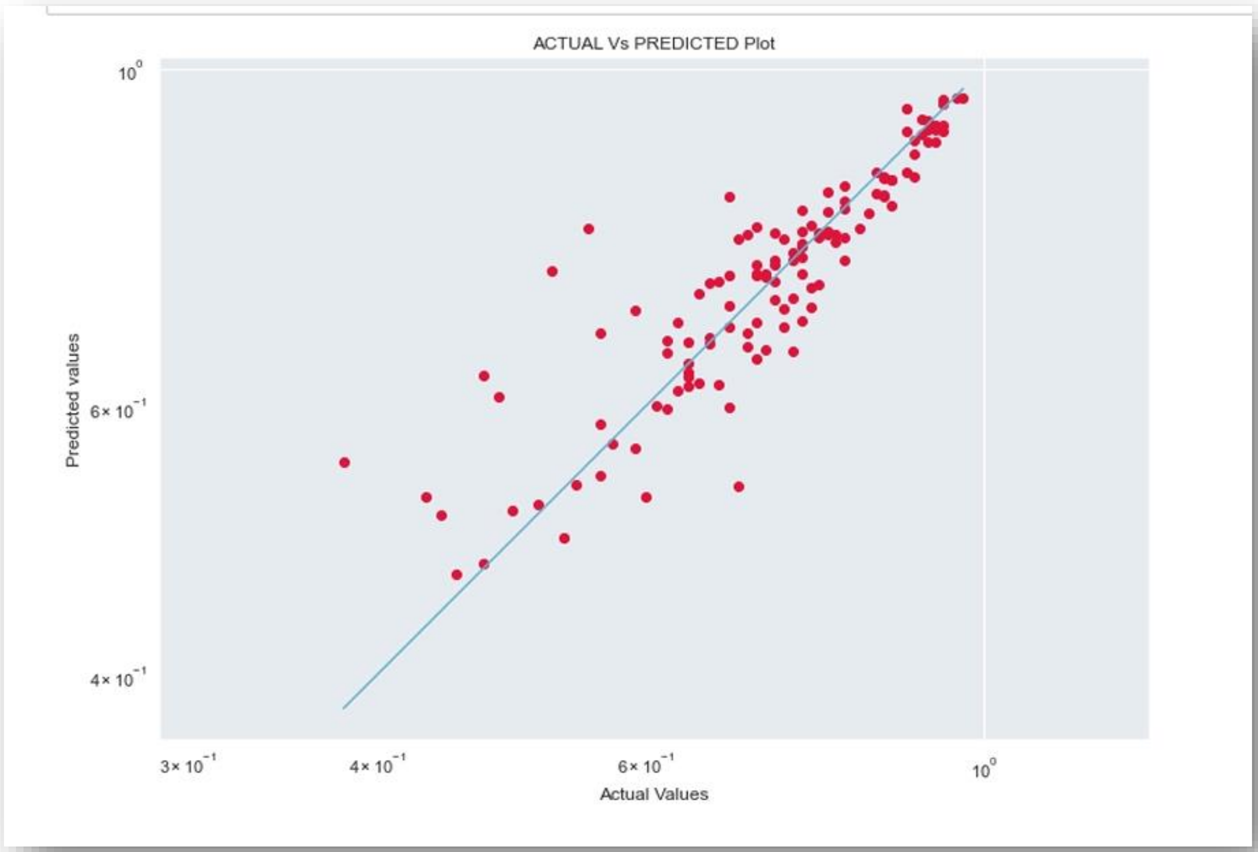
r2_score

It is a very powerful technique and can be used to understand the factors that influence profitability.

It looks at a relationship between the mean of the dependent and independent variables.

It is prone to underfitting and sensitive to out layers.
Simple implementation and highly interpretable.

This is the plot of actual vs predicted values for the model (MLR)



ALGORITHMS USED:

OUTLIER REMOVAL:

The first main algorithm used here to find the various outliers present in the dataset using the construction of the SEABORN DISPLOT. The displot here basically shows the frequency of the observation of occurrences. As seen , when CHANCE OF ADMIT is less than 0.4 so we remove the outliers by using:

```
dataset.drop(dataset.index[list((np.where(dataset['Chance of Admit ']<0.4)))],inplace=True)
```

CORRELATION MATRIX:

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

Correlation Matrix is here used to show the probabilistic dependencies of Chance of Admit on various other factors , and since all of them have quite a large share of dependency so we choose all of them.

TRAIN_TEST_SPLIT:

It is used to split arrays or matrices into random train and test subsets.

So basically what we are doing here is we are splitting the training and testing part in the ratio of 3:1, so that we can later on predict the values based upon the analysis.

X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state = 5)

MSE:

Mean square error (MSE) is the average of the square of the errors. The larger the number the larger the error. **Error** in this case means the difference between the observed values y1, y2, y3, … and the predicted ones pred(y1), pred(y2), pred(y3), … We square each difference (pred(yn) – yn)) ** 2 so that negative and positive values do not cancel each other out.

RMSE:

Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.

R2 SCORE:

The **r2 score** varies between 0 and 100%. It is *the proportion of the variance in the dependent variable that is predictable from the independent variable(s).*

HERE MSE, RMSE AND R2 SCORE are used to see the useability, ability and the accuracy of the model that we have designed.

```
mse = mean_squared_error(y_test,y_test_predict)

rmse = np.sqrt(mean_squared_error(y_test,y_test_predict))

r2 = r2_score(y_test,y_test_predict)
```

MODEL ANALYSIS

The MSE obtained in this model (MLR) is 0.0032448492350027444

The RMSE value obtained in this model (MLR) is 0.056963578144308 53

The R-Squared score value obtained in this model (MLR) is 0.8127800620182747

Multiple linear regression model is the most popular type of linear regression analysis. It is used to show the relationship between one dependent variable and two or more independent variables. In fact, everything you know about the simple linear regression modeling extends (with a slight modification) to the multiple linear regression models.

Random Forest Regressor (RFR)

Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression

Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between these trees while building the trees .

In this model (RFR) I first imported all the necessary libraries required for this model :

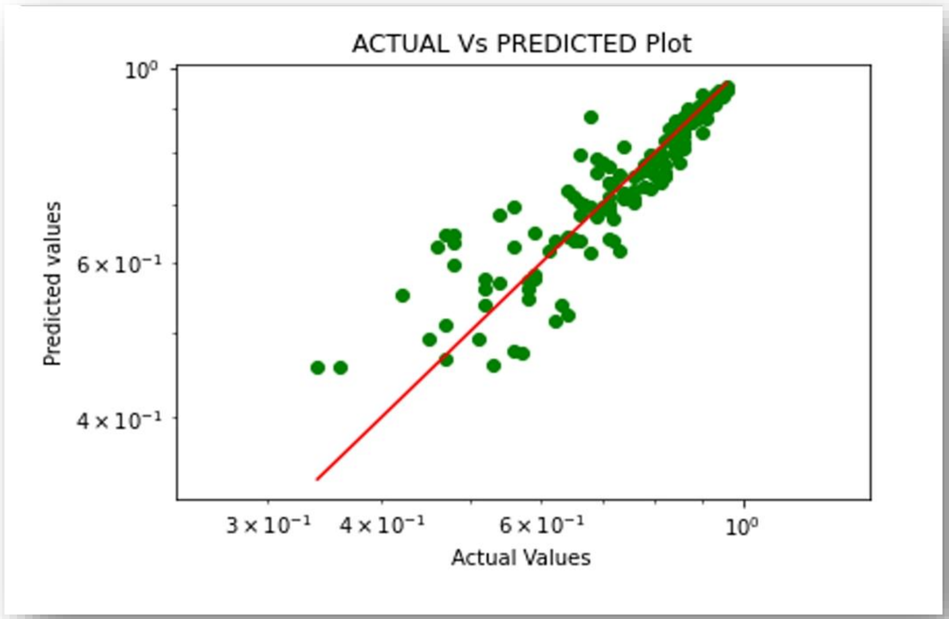
- Pandas
- Scikit-Learn
- Seaborn
- NumPy
- Matplotlib
- RandomForestRegressor
- mean_squared_error
- train_test_split
- r2_score

- It operates by constructing a multitude of decision trees at training time and outputting the class that is the **mode** of the **classes (classification)** or **mean prediction (regression)** of the individual trees
- A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which **aggregates many decision trees**.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as **bagging**. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

This is the plot of actual vs predicted values for the model(RFR)



ALGORITHMS USED:

TRAIN_TEST_SPLIT:

It is used to split arrays or matrices into random train and test subsets.

So basically what we are doing here is we are splitting the training and testing part in the ratio of 3:1, so that we can later on predict the values based upon the analysis.

```
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state = 25)
```

MSE:

Mean square error (MSE) is the average of the square of the errors. The larger the number the larger the error. **Error** in this case means the difference between the observed values y1, y2, y3,

… and the predicted ones pred(y1), pred(y2), pred(y3), … We square each difference (pred(yn) – yn)) ** 2 so that negative and positive values do not cancel each other out.

RMSE:

Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.

R2 SCORE:

The **r2 score** varies between 0 and 100%. It is *the proportion of the variance in the dependent variable that is predictable from the independent variable(s).*

HERE MSE, RMSE AND R2 SCORE are used to see the useability, ability and the accuracy of the model that we have designed.

```
mse = mean_squared_error(y_test,y_test_predict)

rmse = np.sqrt(mean_squared_error(y_test,y_test_predict))

r2 = r2_score(y_test,y_test_predict)
```

MODEL ANALYSIS:

The MSE obtained in this model(RFR) is 0.0038120131999999952

The RMSE value obtained in this model(RFR) is 0.06174150305912543

The R-Squared score value obtained in this model(RFR) is 0.8189295004736319

Random Forest grows multiple decision trees which are merged together for a more accurate prediction. The logic behind the Random Forest model is that multiple uncorrelated models (the individual decision trees) perform much better as a group than they do alone.

Principal Component Analysis with Multiple Linear Regression

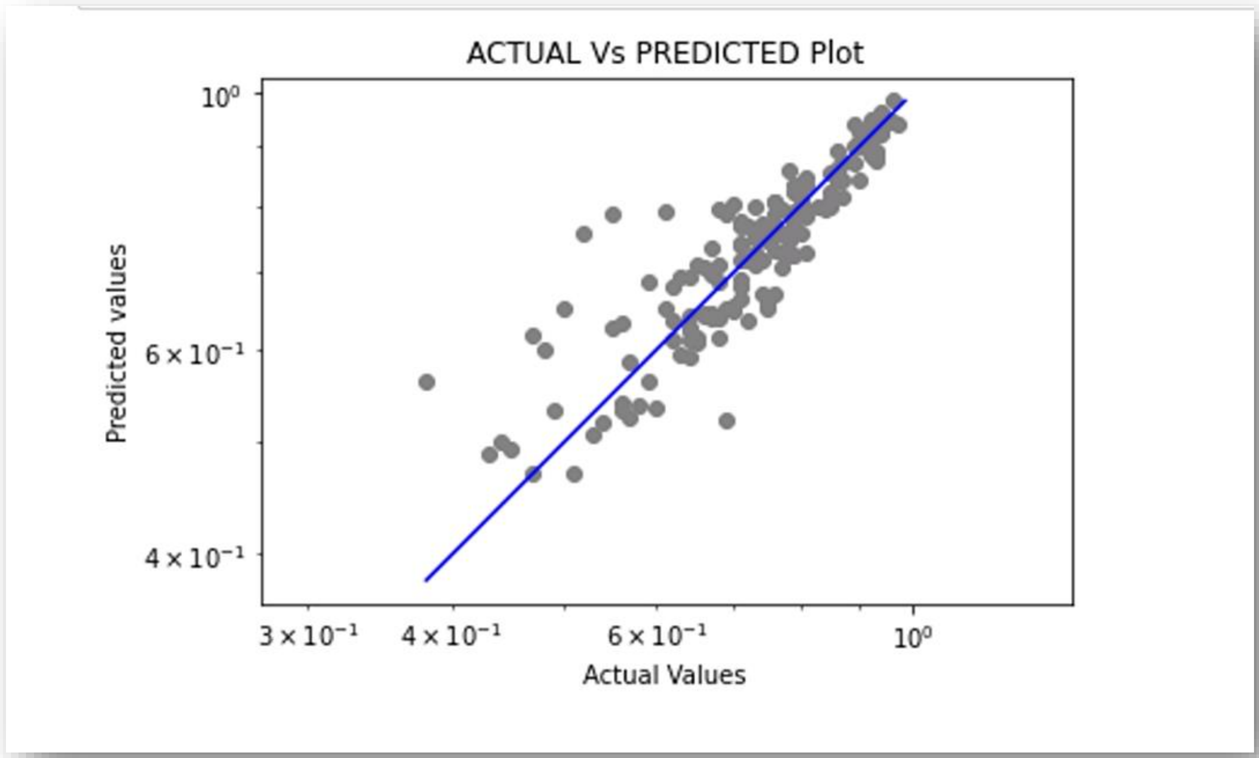
(PCA WITH MLR)

- Principal Component Analysis (PCA) is an unsupervised, non-parametric statistical technique primarily used for dimensionality reduction in machine learning
- The primary problem associated with high-dimensionality in the machine learning field is model overfitting, which reduces the ability to generalize beyond the examples in the training set

In this model (PCA with MLR), I imported all the necessary libraries required for this model

- NumPy
- Pandas
- **Scikit-Learn**
- **Seaborn**
- **Matplotlib**
- **LinearRegression**
- mean_squared_error
- train_test_split
- r2_score
- StandardScaler
- PCA

This is the plot of actual vs predicted values for the model (PCA with MLR)



Goal of PCA:-

- Identify patterns in data
- Detect the correlation between variables
- Reduce the dimensions of a d-dimensional dataset by projecting it onto a k-dimensional subspace(where k<d)

ALGORITHMS USED:

CONVERTING THE CHANCE OF ADMISSION TO VARIOUS CATEGORIES:

The first major algorithm we have used here is we converted the Chance of Admit column to the sequence of BAD, MEDIUM and GOOD, which would allow an easier approach for the PCA to prepare dataset for analysis.

```
dataset1 = dataset.copy()

dataset1['Chance of Admit ']=pd.cut(np.array(y),3, labels=["bad", "medium", "good"])

dataset1['Chance of Admit '][:5]
```

IDENTIFYING THE PRINCIPAL COMPONENTS:

Here basically we are taking 4 principal components, because earlier it was seen that at components =3 the dataset did not produce 85% of the total information(which would make the model more inaccurate), hence by adding one more principal component now we have 91.57% of the total information which is considerable.

```
        pca = PCA(n_components=4)

        principalComponents = pca.fit_transform(x)

        principalDf = pd.DataFrame(data = principalComponents

          , columns = ['principal component 1', 'principal component 2','principal component 3','principal component 4'])
```

Then we took the dataset produced from PCA along with the Chance of Admit column to prepare our final dataset on which we will perform our Multiple Linear Regression.

TRAIN_TEST_SPLIT:

It is used to split arrays or matrices into random train and test subsets.

So basically what we are doing here is we are splitting the training and testing part in the ratio of 7:3, so that we can later on predict the values based upon the analysis.

In case of PCA it is important to note the training part must be atleast 70% of the dataset other there might be inaccuracy in the desired output.

```
        X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state = 5)
```

MSE:

Mean square error (MSE) is the average of the square of the errors. The larger the number the larger the error. **Error** in this case means the difference between the observed values y1, y2, y3, … and the predicted ones pred(y1), pred(y2), pred(y3), … We square each difference (pred(yn) – yn)) ** 2 so that negative and positive values do not cancel each other out.

RMSE:

Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.

R2 SCORE:

The **r2 score** varies between 0 and 100%. It is *the proportion of the variance in the dependent variable that is predictable from the independent variable(s).*

HERE MSE, RMSE AND R2 SCORE are used to see the useability, ability and the accuracy of the model that we have designed.

```
mse = mean_squared_error(y_test,y_test_predict)

rmse = np.sqrt(mean_squared_error(y_test,y_test_predict))

r2 = r2_score(y_test,y_test_predict)
```

MODEL ANALYSIS:

The MSE obtained in this model (PCA with MLR) is 0.0035268267109371987

The RMSE value obtained in this model (PCA with MLR) is 0.05938709212393884

The R-Squared score value obtained in this model (PCA with MLR) is 0.7937833209433434

PCA is based on the Pearson correlation coefficient framework and inherits the following assumptions.

- ***Sample size:*** Minimum of 150 observations and ideally a 5:1 ratio of observation to features (Pallant, 2010)
- ***Correlations:*** The feature set is correlated, so the reduced feature set effectively represents the original data space.
- ***Linearity:*** All variables exhibit a constant multivariate normal relationship, and principal components are a linear combination of the original features.

Principal Component Analysis with Random Forest Regression

(PCA WITH RFR)

- Principal Component Analysis (PCA) is an unsupervised, non-parametric statistical technique primarily used for dimensionality reduction in machine learning
- The primary problem associated with high-dimensionality in the machine learning field is model overfitting, which reduces the ability to generalize beyond the examples in the training set

In this model (PCA with RFR), I imported all the necessary libraries required for this model

- NumPy
- Pandas
- **Scikit-Learn**
- **Seaborn**
- **Matplotlib**
- RandomForestRegressor
- mean_squared_error
- train_test_split
- r2_score
- StandardScaler
- PCA

ALGORITHMS USED:

CONVERTING THE CHANCE OF ADMISSION TO VARIOUS CATEGORIES:

The first major algorithm we have used here is we converted the Chance of Admit column to the sequence of BAD, MEDIUM and GOOD, which would allow an easier approach for the PCA to prepare dataset for analysis.

```
dataset1 = dataset.copy()

dataset1['Chance of Admit ']=pd.cut(np.array(y),3, labels=["bad", "medium", "good"])

dataset1['Chance of Admit '][:5]
```

IDENTIFYING THE PRINCIPAL COMPONENTS:

Here basically we are taking 4 principal components, because earlier it was seen that at components =3 the dataset did not produce 85% of the total information(which would make the model more inaccurate), hence by adding one more principal component now we have 91.57% of the total information which is considerable.

```
pca = PCA(n_components=4)

principalComponents = pca.fit_transform(x)

principalDf = pd.DataFrame(data = principalComponents

   , columns = ['principal component 1', 'principal component 2','principal component 3','principal component 4'])
```

Then we took the dataset produced from PCA along with the Chance of Admit column to prepare our final dataset on which we will perform our Multiple Linear Regression.

TRAIN_TEST_SPLIT:

It is used to split arrays or matrices into random train and test subsets.

So basically what we are doing here is we are splitting the training and testing part in the ratio of 7:3, so that we can later on predict the values based upon the analysis.

In case of PCA it is important to note the training part must be atleast 70% of the dataset other there might be inaccuracy in the desired output.

```
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state = 5)
```

Random Forest Regressor (RFR)

Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression. Random forest is a bagging technique and not a boosting technique..

MSE:

Mean square error (MSE) is the average of the square of the errors. The larger the number the larger the error. **Error** in this case means the difference between the observed values y1, y2, y3, … and the predicted ones pred(y1), pred(y2), pred(y3), … We square each difference (pred(yn) – yn)) ** 2 so that negative and positive values do not cancel each other out.

RMSE:

Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.

R2 SCORE:

The **r2 score** varies between 0 and 100%. It is *the proportion of the variance in the dependent variable that is predictable from the independent variable(s).*

HERE MSE, RMSE AND R2 SCORE are used to see the useability, ability and the accuracy of the model that we have designed.

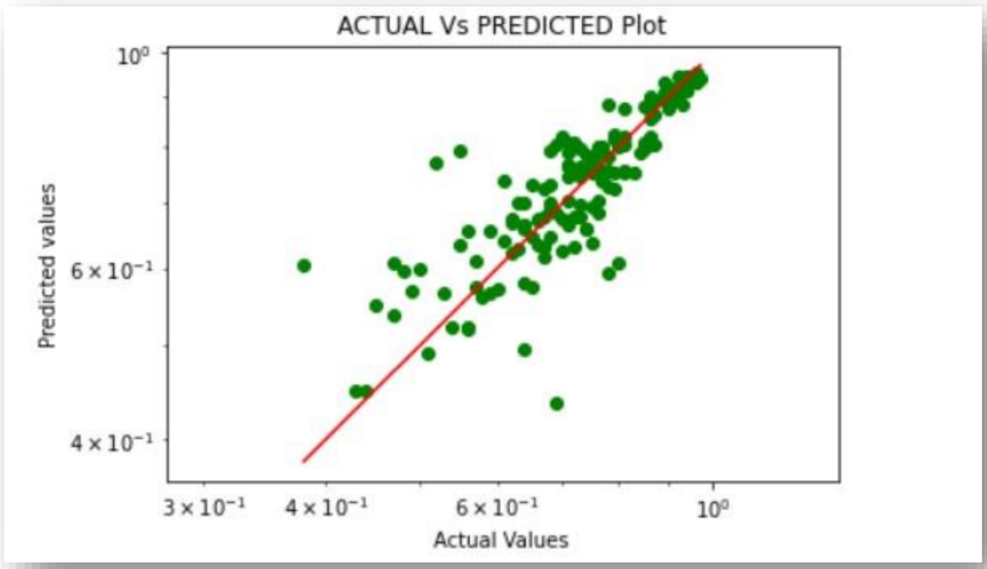
```
mse = mean_squared_error(y_test,y_test_predict)

rmse = np.sqrt(mean_squared_error(y_test,y_test_predict))

r2 = r2_score(y_test,y_test_predict)
```

MODEL ANALYSIS:

The MSE obtained in this model (PCA with RFR) is 0.004636949999999895

The RMSE value obtained in this model (PCA with RFR) is 0.06809515401260202

The R-Squared score value obtained in this model (PCA with RFR) is 0.7288734297643837



CONCLUSION

Admissions Prediction model is build by using 3 different types of algorithms
1.Multiple linear regressor (MLR)
2.Random Forest Regressor (RFR)
3.MLR with PCA on same data set by training we obtained results regarding Mean square error (MSE)**,**Root Mean Square Error (RMSE**),** R-Squared score and the model is build with minimum MSE and RMSE and maximum R2 score.
4.RFR with PCA on same data set by training we obtained results regarding Mean square error (MSE),Root Mean Square Error(RMSE) R-Squared score and the model is built with minimum MSE and RMSE and maximum R2 score.

ANALYSIS:

MODEL	**MSE**	**RMSE**	**R2 SCORE**
MLR	0.0032448492350027444	0.05696357814430853	0.8127800620182747
RFR	0.0038120131999999952	0.0038120131999999952	0.8189295004736319
MLR with PCA	0.0035268267109371987	0.05938709212393884	0.7937833209433434
RFR with PCA	0.0046369499999999895	0.06809515401260202	0.7288734297643837

Lowe**r** values of MSE and RMSE and higher value of R2 Score indicate how well the regression model fits the observed data.

BIBLIOGRAPHY

Kaggle (Dataset and Information)(https://www.kaggle.com/)

Towards Data Science (https://towardsdatascience.com/)

Peter Bruce, Andrew Bruce, Peter Gedeck **. "Practical Statistics For Data Science "**

Geron Aurelien **. "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems"**

KDNuggets (https://www.kdnuggets.com/)