



# Capstone Project

## Bike Sharing Demand Prediction

### Team Members

Saurabh Hase

Aakash Sharma

# Is Bike Available for Rent?

To make availability of the Bikes as per the need for renting purpose, By considering the various factors like Temperature, Date, Visibility, Snowfall, Rainfall and etc.

Our goal here is to build a predictive model, which could help Rental bike services in predicting the Rental bike count for particular condition

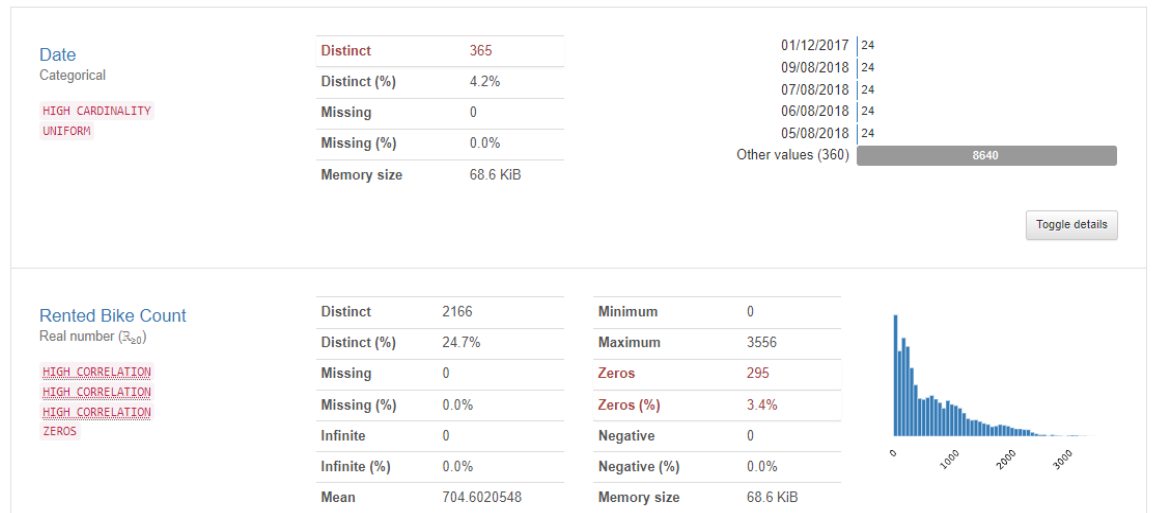


# Data Description

- **Date:** year-month-day
- **Rented Bike count** - Count of bikes rented at each hour
- **Hour** - Hour of the day
- **Temperature**-Temperature in Celsius
- **Humidity** - %
- **Windspeed** - m/s
- **Visibility** - 10m
- **Dew point temperature** - Celsius
- **Solar radiation** - MJ/m<sup>2</sup>
- **Functional Day** - NoFunc(Non Functional Hours), Fun(Functional hours)
- **Rainfall** - mm
- **Snowfall** - cm
- **Seasons** - Winter, Spring, Summer, Autumn
- **Holiday** - Holiday/No holiday

# Exploratory Data Analysis

## Pandas Profiling



The **pandas\_profiling** is an open-source library in Python that includes a method named as `ProfileReport()` which generates a basic report on the input `DataFrame`.

# Sweetviz

Sweetviz is an open-source Python library that generates beautiful, high-density visualizations to kickstart EDA



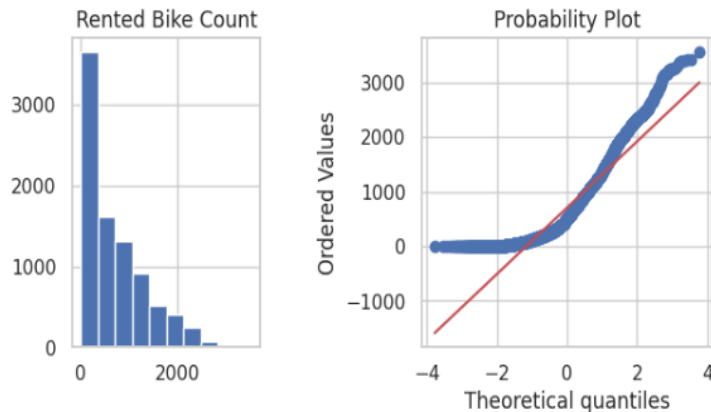
# Normalisation

**Normalization** is used to scale the data of an attribute so that it falls in a smaller range, such as -1.0 to 1.0 or 0.0 to 1.0.

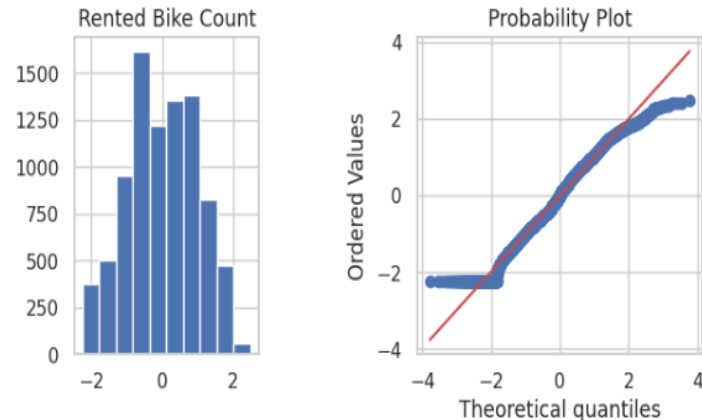
## Yeo-Johanson

This transformation is an extended version of box cox transformation which also deals with negative values as well as zero values in the dataset

**Before transformation**



**After Transformation**

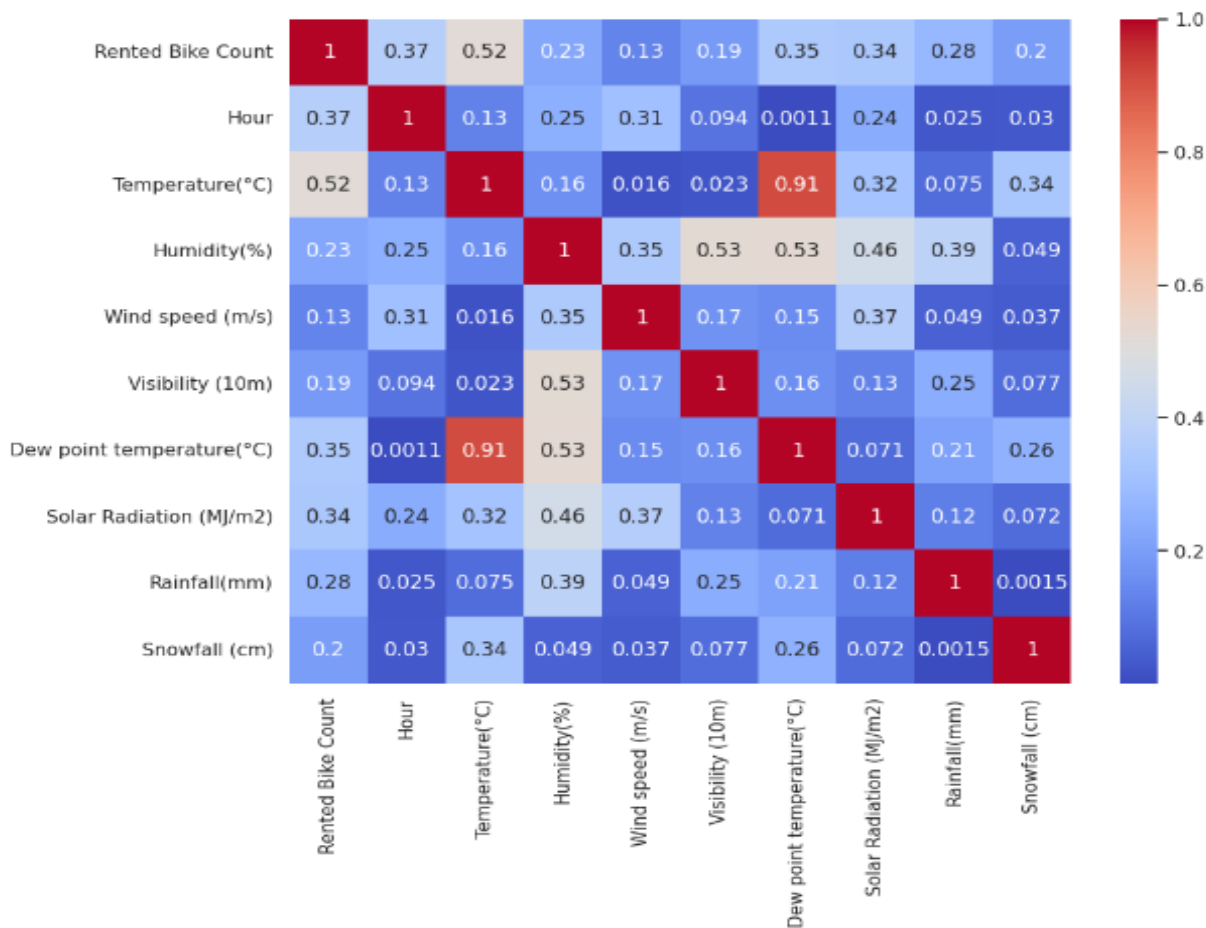


# Correlation

## VIF

	variables	VIF
0	Hour	1.180457
1	Temperature(°C)	1.602807
2	Humidity(%)	2.523959
3	Wind speed (m/s)	1.297669
4	Visibility (10m)	1.502836
5	Solar Radiation (MJ/m2)	1.827327
6	Rainfall(mm)	1.198674
7	Snowfall (cm)	1.163389

## Heatmap



# One hot encoding

One Hot Encoding is method to produce binary integers of 0 and 1 to encode categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

## Machine learning models

- Linear Regression
- Lasso Regression
- Ridge Regression
- Elastic Net Regression
- Polynomial Regression
- Decision Tree
- Random Forest
- Gradient Boosting
- eXtreme Gradient Boosting



# Evaluation matrices

## Mean Squared Error(MSE)

Mean Squared Error ( **MSE** ) is defined as Mean or Average of the square of the difference between actual and estimated values.

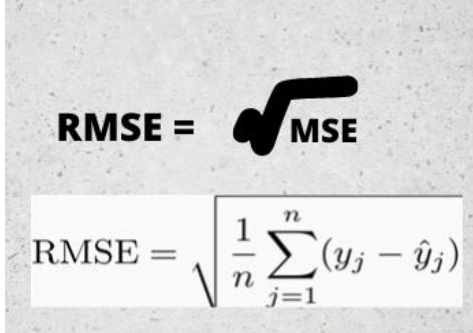
$$MSE = \frac{1}{n} \sum \underbrace{\left( y - \hat{y} \right)^2}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}}$$

This means that MSE is calculated by the **square of the difference** between the predicted and actual target variables, divided by the number of data points.

It is always **non-negative** values and close to zero are better.

## Root Mean Squared Error(RMSE)

Root Mean Square Error ( **RMSE** ) is also used as a measure for model evaluation. It is the square root of Mean Squared Error (MSE). This is the same as **Mean Squared Error** (MSE) but the root of the value is considered while determining the accuracy of the model.



The diagram illustrates the relationship between RMSE and MSE. At the top, it states **RMSE = √ MSE**, where the square root symbol is a large, bold, black icon. Below this, a white rectangular box contains the mathematical formula for RMSE: 
$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

## R Squared (R<sup>2</sup>)

R-Squared is a statistical measure of fit that indicates how much **variation** of a dependent variable is explained by the independent variable(s) in a regression model.

$$\mathbf{R^2\ Squared = 1 - \frac{SSr}{SSm}}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

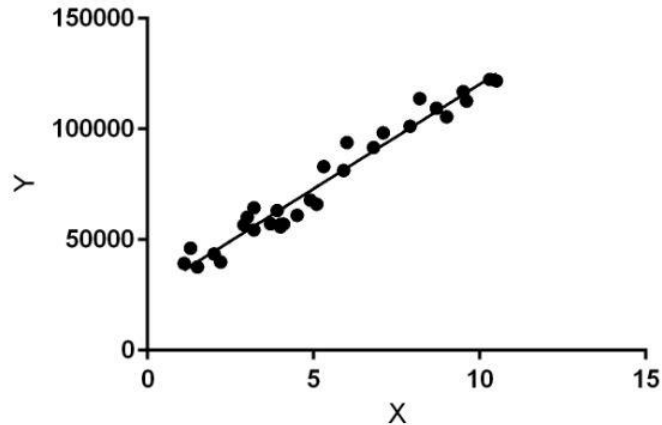
## Adjusted R-squared

**R-squared** describes the amount of variance of the dependent variable represented by every single independent variable, while **Adjusted R-squared** measures variation explained by only the independent variables that actually affect the dependent variable.

$$R^2_{adjusted} = \left[ \frac{(1-R^2)(n-1)}{n-k-1} \right]$$

# Linear Regression

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting.



$$y = b \cdot x + c.$$

=====Evaluation Matrix=====

MSE : 1.076126943011156  
RMSE : 1.0373653854891998  
R2 : 0.2705943519735111  
Adjusted R2 : 0.26045785194303306

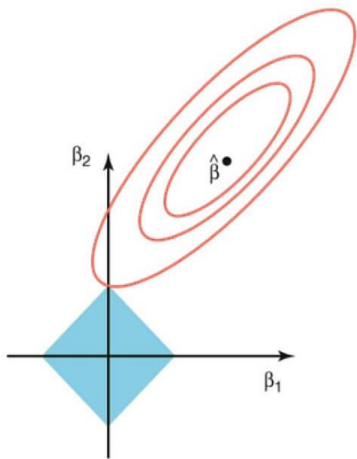
=====Evaluation Matrix=====

Loss function:  $(\text{Predicted output} - \text{actual output})^2$ .

# Lasso Regression

Lasso regression is a **regularization technique**. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters)

## LASSO REGRESSION



$$N^{-1} \sum_{i=1}^N f(x_{\{i\}}, y_{\{i\}}, \alpha, \beta)$$

=====Evaluation Matrix=====

MSE : 1.0757081296209068

RMSE : 1.0371635018746594

R2 : 0.2708782263382419

Adjusted R2 : 0.26074567129024995

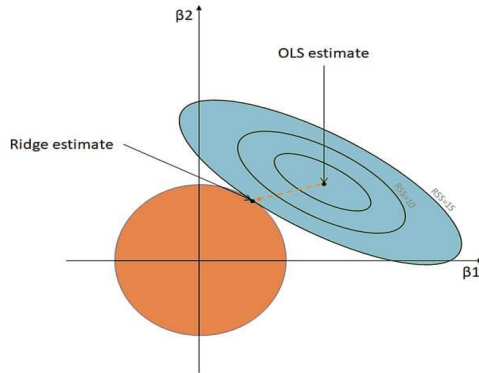
=====Evaluation Matrix=====

# Ridge Regression

**Ridge Regression** is another type of regression in machine learning and is usually used when there is a **high correlation** between the parameters. . This is because as the correlation increases the least square estimates give unbiased values

$$\beta = (X^T X + \lambda I)^{-1} X^T y$$

**RIDGE REGRESSION**



→

=====Evaluation Matrix=====

MSE : 1.0750024132133889

RMSE : 1.036823231420568

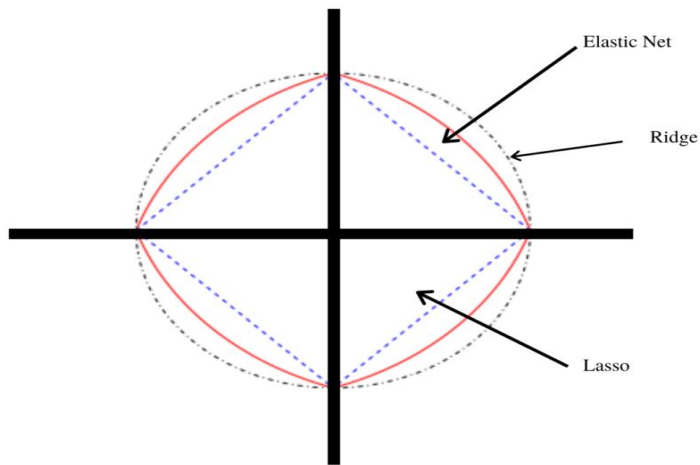
R2 : 0.2713565653826191

Adjusted R2 : 0.2612306577793666

=====Evaluation Matrix=====

# Elastic net regression

Elastic net linear regression uses the **penalties from both** the lasso and ridge techniques to regularize regression models. The technique combines both the **lasso and ridge** regression methods by learning from their shortcomings to improve the regularization of statistical models.



=====Evaluation Matrix=====

MSE : 1.0755311517134152

RMSE : 1.0370781801356228

R2 : 0.2709981830832511

Adjusted R2 : 0.2608672950658788

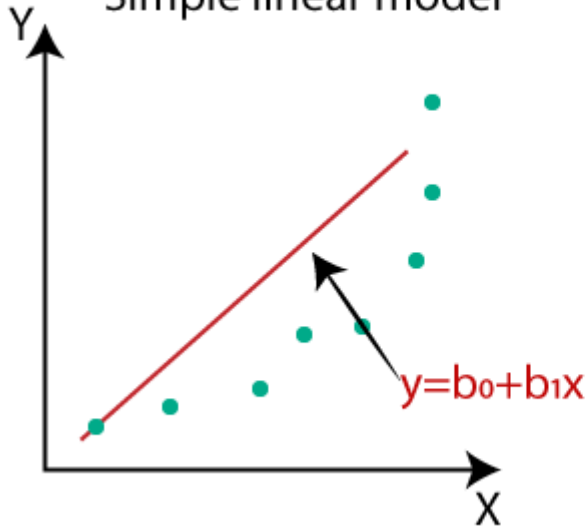
=====Evaluation Matrix=====

# Polynomial regression

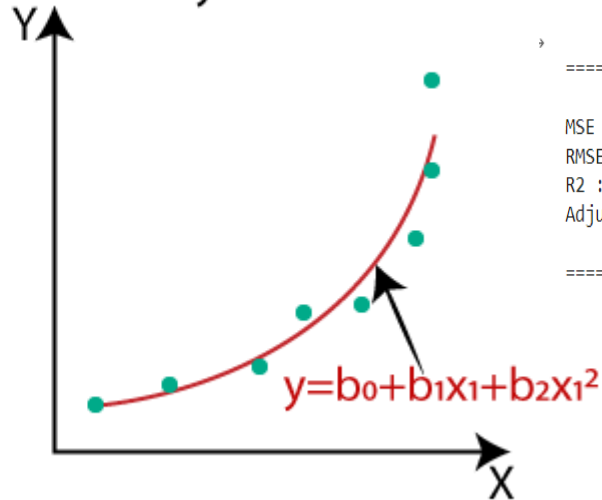
Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial

$$y = b_0 + b_1x_1 + b_2x_1^2 + b_3x_1^3 + \dots + b_nx_1^n$$

Simple linear model



Polynomial model



=====Evaluation Matrix=====

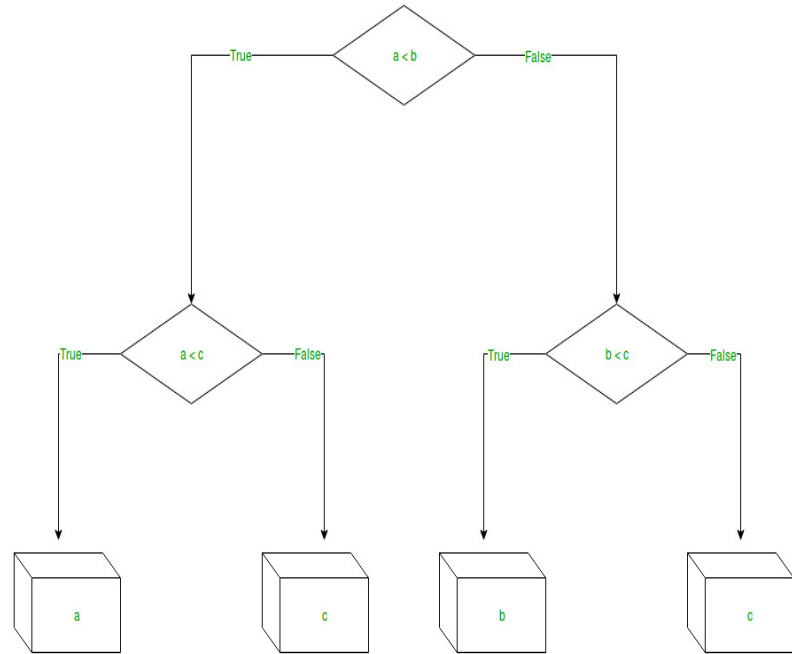
MSE : 0.5172362381404242  
RMSE : 0.7191913779658542  
R2 : 0.6494140064852105  
Adjusted R2 : 0.6445419370906795

=====Evaluation Matrix=====

# Decision Tree Regression

Decision tree builds regression or classification models in the form of a **tree structure**. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**.

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values



=====Evaluation Matrix=====

MSE : 0.3835991113517791  
 RMSE : 0.6193537852889729  
 R2 : 0.7399940962215746  
 Adjusted R2 : 0.736380812092633

=====Evaluation Matrix=====



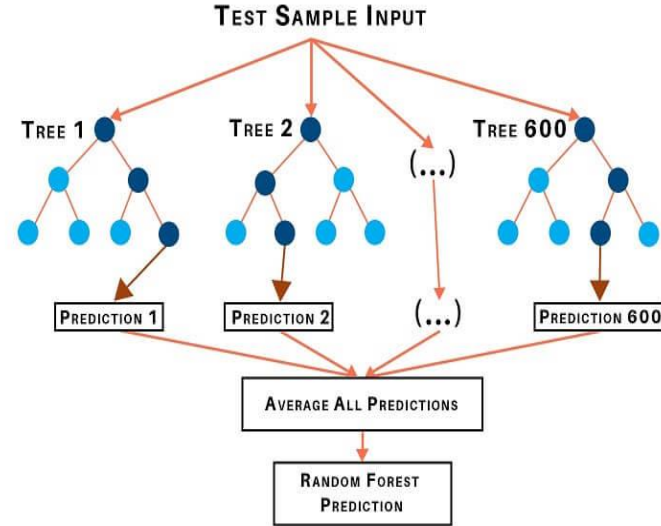
# Random Forest Regressor

Random forest, as its name suggests, comprises an enormous amount **of individual decision trees** that work as a group or as they say, an ensemble.

Every individual decision tree in the random forest lets out a class prediction and the class with the **most votes** is considered as the model's prediction.

Random forest uses this by permitting every individual tree to randomly sample from the dataset with replacement, bringing about various trees. This is known as **Bagging**.

## RANDOM FOREST REGRESSION



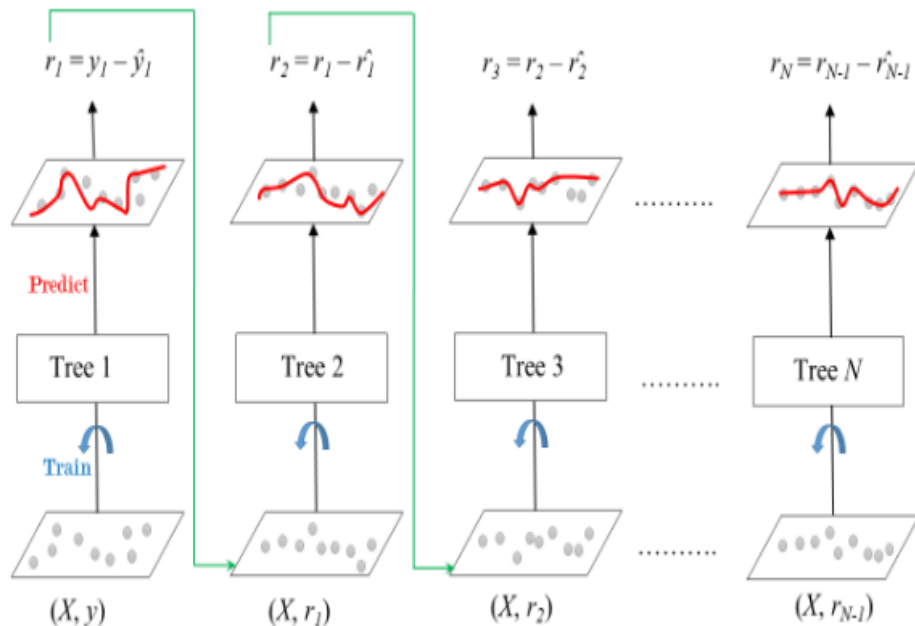
=====Evaluation Matrix=====

MSE : 0.3299459372764718  
RMSE : 0.5744092071654769  
R2 : 0.7763605569437385  
Adjusted R2 : 0.7732526550135994

=====Evaluation Matrix=====

# Gradient Boost

Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its **predecessor's error**. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the **residual errors** of predecessor as labels.



=====Evaluation Matrix=====

MSE : 0.23948280694289772

RMSE : 0.48936980591664797

R2 : 0.8376770388253573

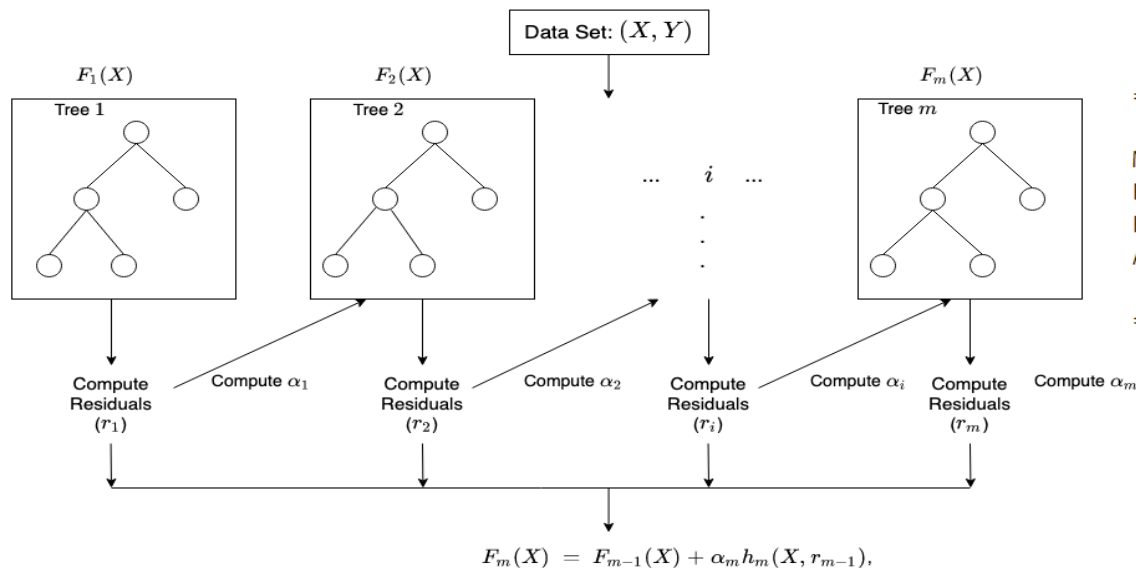
Adjusted R2 : 0.8354212478188769

=====Evaluation Matrix=====

# Xtreme Gradient Boost

XGBoost is a powerful approach for building supervised regression models. The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners. The objective function contains **loss function and a regularization term**. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values.

XGBoost optimised the gradient boosting algorithm through **parallel processing, Tree-pruning, handling missing values and regularisation** to avoid overfitting/bias



=====Evaluation Matrix=====

MSE : 0.21646689301638986

RMSE : 0.4652600273141782

R2 : 0.8532773708507885

Adjusted R2 : 0.8512383765835152

=====Evaluation Matrix=====

# Results

	Models	Mean_square_error	Root_Mean_square_error	R2	Adjusted_R2
	Linear	1.08	1.04	0.27	0.26
1	Lasso	1.08	1.04	0.27	0.26
2	Ridge	1.08	1.04	0.27	0.26
3	Elasticnet	1.08	1.04	0.27	0.26
4	Polynomial_degree-2	0.52	0.72	0.65	0.64
5	Polynomial_degree-3	6.65E+33	8.15E+16	-4.51E+33	-4.57E+33
6	Decision_Tree	0.38	0.62	0.74	0.74
7	Random_Forest	0.33	0.57	0.78	0.77
8	Gradient_Boosting	0.24	0.49	0.84	0.84
9	Xtreme_GB	0.22	0.47	0.85	0.85

# Conclusion

With this we have completed our experiments of using different models and evaluating those models with help of different evaluation matrices.

## **Steps we have followed:**

- Exploratory Data Analysis
- Handling Null values
- Dealing with outliers
- Normalisation using Yeo-Johanson Method
- Checking Correlation of variables
- One hot encoding
- Applying Various models like Linear Regression, Lasso Regression, Ridge Regression, Elastic Net Regression, Polynomial Regression, Decision Tree, Random Forest, Gradient Boosting, eXtreme Gradient Boosting
- Evaluating Models using Mean squared error, Rootmean Squared error, R2 Score, Adjusted R2 squared

**Finally after doing comparison we have found that XG Boost model is giving us best R2 score of 85% and Adjusted R2 Score of 85%**

**THANK YOU**