**Seoul Bike Sharing Demand Prediction**

**Team Members**

Meenu Benny, Thasniya Mansoora, T.Ompriya Subudhi

*Data science Enthusiast*

Alma Better

# Abstract:

# Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

In market share we can see that Bike Sharing system has a global market share which was valued around $3.39 billion in 2019 and is projected to grow to $6.98 billion by 2027 with a compound annual growth rate of around 14% indicatively from 2020 to 2027.

The data generated by these systems makes them attractive for researchers because the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Bike sharing systems therefore function as a sensor network, which can be used for studying mobility in a city

**Keywords***: Bike-Sharing, Data Mining, Data Analysis, Linear Regression, Machine Learning.*

# Problem Statement

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall, Seasons), the number of bikes rented per hour, and date information.

Based on the given data we have to build a machine learning model which will be helping us to predict the number of bikes that must be made available by predicting the demand for bikes rented per day.

* **Date**: year-month-day
* **Rented Bike count** - Count of bikes rented at each hour
* **Hour** - Hour of the day
* **Temperature**-Temperature in Celsius
* **Humidity** - %
* **Wind Speed** - m/s
* **Visibility** - 10m
* **Dew point temperature** - Celsius
* **Solar radiation** - MJ/m2
* **Rainfall** - mm
* **Snowfall** - cm
* **Seasons** - Winter, Spring, Summer, Autumn
* **Holiday** - Holiday/No holiday
* **Functional Day** – No Func(Non Functional Hours), Fun(Functional hours)

# Introduction:

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world globally in South Korea, São Paulo, China and Australia.

Bike sharing system generally rents bikes on an hour, day and month basis and is generally based on static pricing inclusive of hour, days or month. Because of its affordability and easy renting system anyone can commute on arrival.

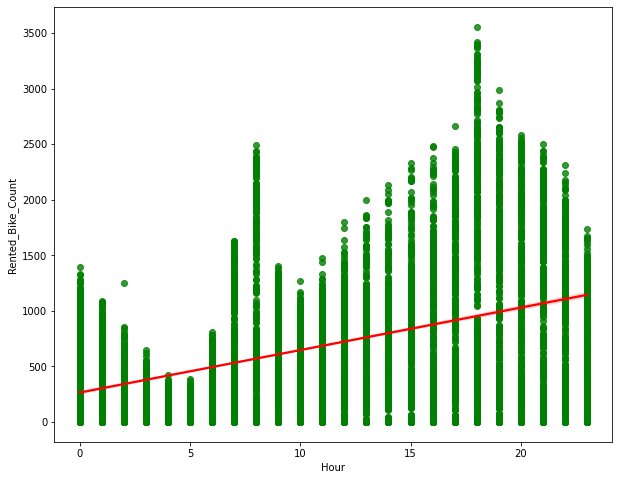
We are tasked to build a predictive model so as to find the approximate number of bikes rented based on the given dataset.

# Exploratory Data Analysis

**Exploratory data analysis** (EDA) is an approach of analyzing data sets to summarize their main characteristics, often use statistical graphics and other data visualization methods like matplotlib and seaborn. It will help us to distribute and relate between dependent and independent variables also it supports the selection of appropriate statistical tools and techniques. We have gone through analysis of every independent as well as the dependent variable to check which independent factor affects the dependent factor.

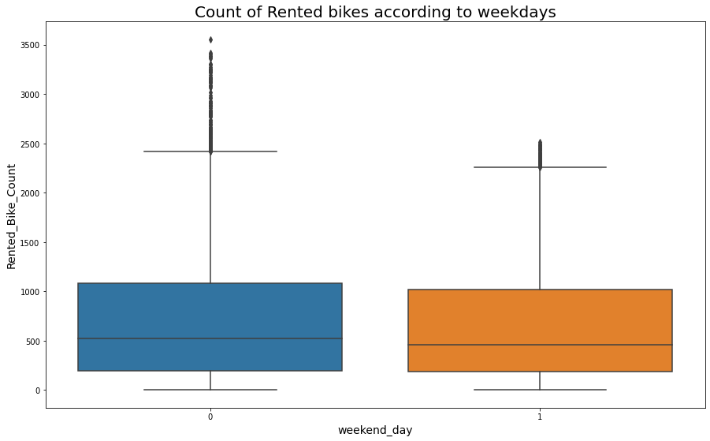
## Hour based Analysis

The rented bike count with hour based analysis showed that the bike demand is at its peak at 08:00 AM and in the evening between 06:00 PM to 09:00 PM. Here we can conclude that most of the bike users belong to the working category as the time indicating bike count at the peak is mostly the working hours start and end time.



## Weekend day and Week off day based Analysis

The Weekend day and Week off day based Analysis shows almost equal weightage on rented bike count.



## Count of rented bike according to weekend days

## 

## The below plot shows that for weekends the

## rented bike counts remain in saddle condition

## whereas for weekdays rented bike count is peak

## at 8.00 A.M and in the evening at 6.00 P.M which

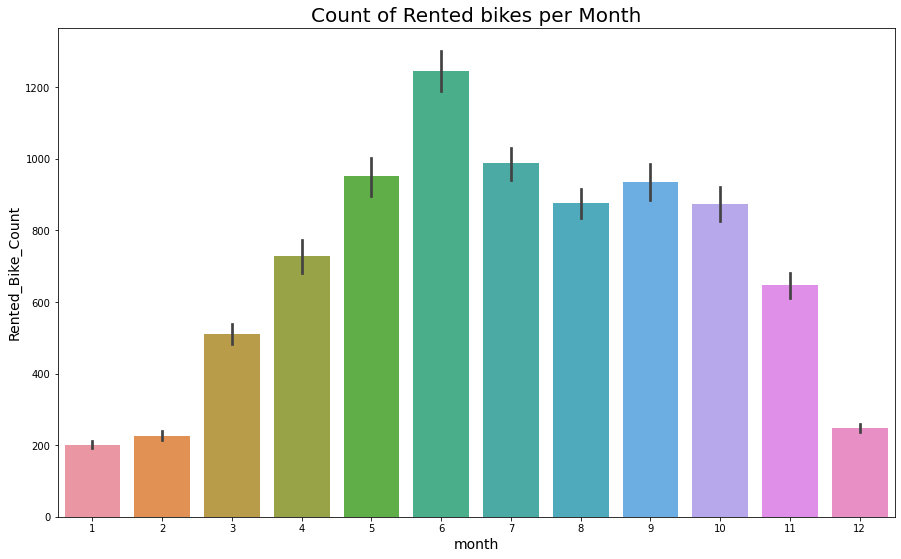
## may be the result of working and class time which

## rent off bikes during the day.

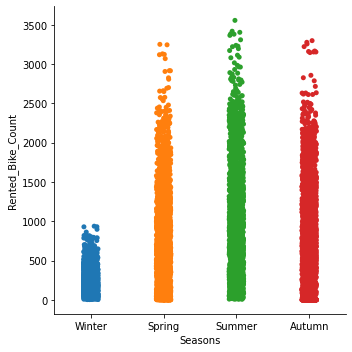
## 

## Month based Analysis

The below chart contains the average bike count over each month of a calendar year. We can see here that the graph shows more entries in months number 5 to 10 i.e., May to October which mostly correlates to season data. We can see that the most rentals are in June and May while the least are in January and February.



## 3.5 Season-wise Analysis

During the season-wise analysis, it was found that the month plays a significant role in rented bike demands. The demands are most likely to be high during summer followed by autumn and spring while winter shows the least demand.

## Function day v/s Rented bike count

## The below box plot figure shows the dependency of rented bike count on the functioning days. Here we can clearly see that the rented bike count directly proportional to only functioning days.

## 

## Holiday v/s Rented bike count

## The below box plot figure shows the relation

## of rented bike count on holidays. Since its values

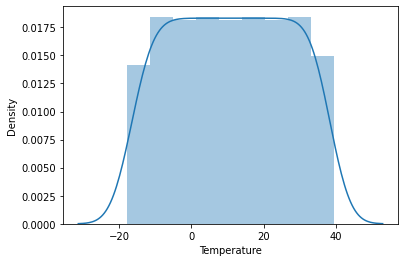
## are unidirectional it may not be an important

## feature to predict bike sharing demand.

## Analyzing Numerical Variables

The numerical variables of the data set include Temperature(°C), Humidity (%), Wind Speed (m/s), Visibility (10m), Dew Point Temperature(°C), Solar Radiation (MJ/m2), Rainfall (mm) Snowfall (cm). All the independent variables listed here represent the weather of the city which has a crucial role in rented bike demand deviation.

## 3.8.1Temperature

 In the density plot for **Temperature** we can see that the median is greater than the mean we can say to some extent that this is negatively skewed.

## 3.8.2 Snowfall

## The average snowfall in Seoul is 2 cm The regression plot shows a similar decrease in the Rented Bike Count with an increase in snowfall. Less is the snowfall, more is the rented bike count.

## 

## 3.8.3 Rainfall

## The average rainfall in Seoul is 2 mm The regression plot shows a similar decrease in the Rented Bike Count with an increase in rainfall. Less is the rainfall, more is the rented bike count.

## 

## 

## 3.8.4 Humidity

In the regression plot for **Humidity** we can see that the mean is greater than the median we can say to some extent that this is positively skewed.

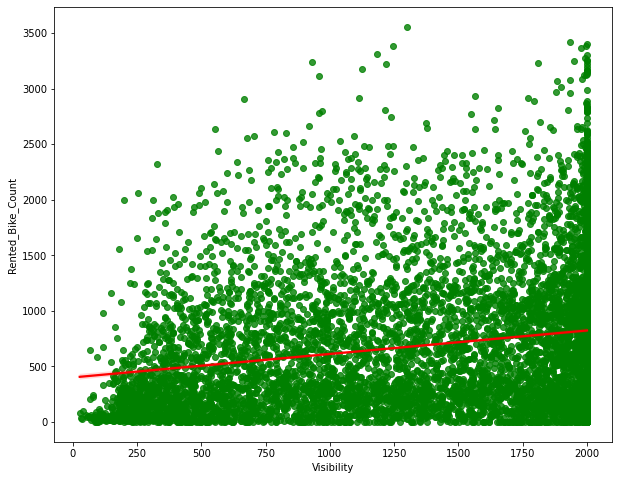
## 

## 

## 

## 3.8.5 Visibility

In the regression plot for **Visibility** we can see that median is greater than mean we can say to some extent that this is negatively skewed.



## 3.8.6 Wind Speed (m/s)

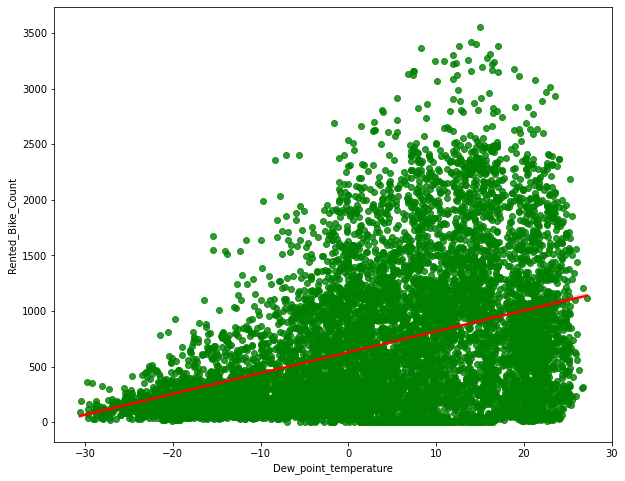
In regression plot for **Windspeed** we can see that mean is greater than the median we can say to some extent that this is positively skewed.

## 

## Dew Point Temperature (°C)

In the density plot for **Dewpoint**

**Temperature** we can see that median is greater than mean we can say to some extent that this is negatively skewed.



## Solar Radiation

In regression plot for **Solar Radiation** we can see that mean is greater than median we can say that this is positively skewed.

## 

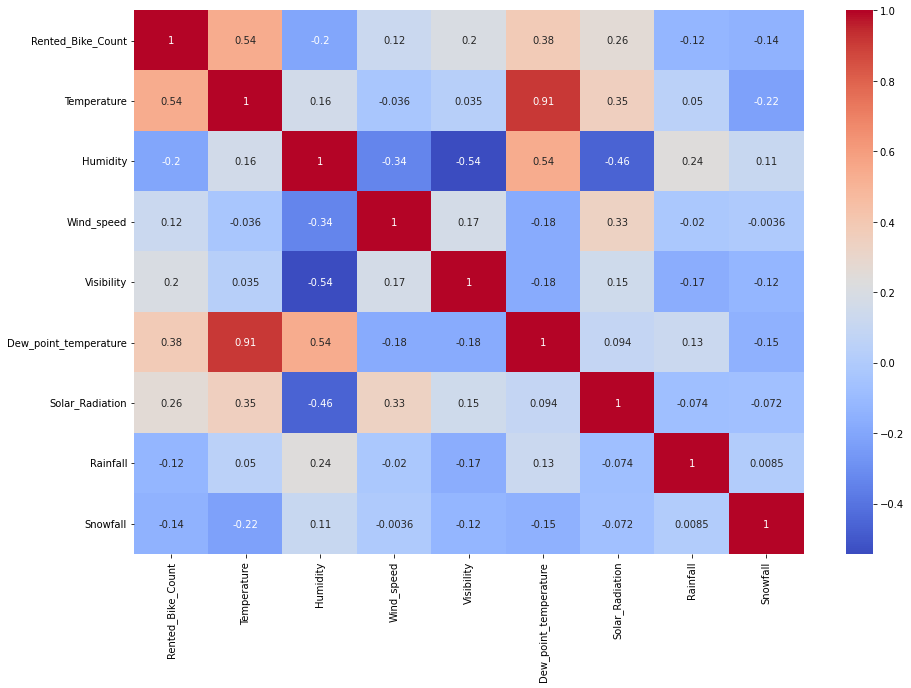
## 

## 

1. **Correlation Analysis**

The correlation analysis has been done to get a better understanding of dependent and independent variables’ multicollinearity. Multicollinearity may not affect the accuracy of the model as much but we might lose reliability in determining the effects of individual independent features on the dependent feature in your model and that can be a problem when we want to interpret your model.

## Heatmap

Let’s check the heatmap plotted concerning independent variables.

We can infer the following from the above heatmap

Temperature and Dew Point Temperature (feels like temperature) are highly correlated, as one would expect.

**5 Feature Description**

* **Date**: Date feature which is **str** type is needed to convert it into Datetime format DD/MM/YYYY.
* **Rented Bike Count**: Number of bikes rented which is our Dependent variable according to our problem statement which is **int** type.
* **Hour**: Hour feature which is in 24-hour format which tells us number bike rented per hour is **int** type.
* **Temperature(°C)**: Temperature feature which is in Celsius scale(°C) is **Float** type.
* **Humidity (%)**: Feature humidity in air (%) which is **int** type.
* **Wind speed (m/s)**: Wind Speed feature which is in (m/s) is **float** type.
* **Visibility (10m)**: Visibility feature which is in 10m, is **int** type.
* **Dew point temperature(°C)**: Dew point Temperature in (°C) which tells us temperature at the start of the day

is **Float** type.

* **Solar Radiation (MJ/m2)**: Solar radiation or UV radiation is **Float** type.
* **Rainfall(mm)**: Rainfall feature in mm which indicates 1 mm of rainfall which is equal to 1 litre of water per metre square is **Float** type.
* **Snowfall (cm)**: Snowfall in cm is Float type. Seasons: Season, in this feature four seasons are present in data is **str** type.
* **Holiday**: whether no holiday or holiday can be retrieved from this feature is **str** type.
* **Functioning Day**: Whether the day is Functioning Day or not can be retrieved from this feature is **str** type.

**6 Feature Engineering**

The provided data in its raw form wasn’t directly used as an input to the model. Several feature engineering was carried out where few features were modified, few were dropped, and few were added. Below is a summary of the feature engineering carried out with the provided data set

* The *Date Time* column which contained the date-time stamp in ‘YYYY-MM-DD HH:MM: SS’ format was split into individual [‘month’, ‘date’, ‘day’, ‘hour’] categorical columns
* Drop *season* column: This is because the season column falls under four categorical data, autumn, summer, spring, and winter and we have added each category individually after encoding.
* Drop *date* column: Intuitively, there should be no dependency on the date. Hence drop this column
* We divided the whole data set into numerical feature and categorical features.
* Columns which have datatype int, float are included in numerical feature and columns which have datatype objects are included in categorical feature.
* Then we plot all regression plot with respect to each dependent and independent features.
* *One Hot Encoding* of categorical feature:

1. *Hours*: Split hour column to hour\_0, hour\_1, ..., hour\_23. Drop the hour column since they are a function of the rest of the retained hour columns.
2. *Month*: Split month column to month\_1, month\_2, ..., month\_12. Drop month columns since they are a function of the rest of the retained month columns
3. *Seasons*: Split the season’s column into autumn, summer, spring, and winter. Drop

the seasons column since it is the function of the rest of the season’s columns

* + *Ordinal Encoding:* The Holiday and Functioning day columns have been encoded using ordinal encoding to provide equal weightage to the deciding entries.

## 6.1 Normalization

The univariate analysis of rented bike data shows a positive skewness which would have been a problem while predicting the values on the test data set. So to ensure the minimization of errors we have taken the square root of the rented bike count data which tends the data for equal weightage.

The need for normalization is basically for making sure that a table contains only data directly related to the primary key, that each data field contains only one item of data, and that redundant (duplicated and unnecessary) data is eliminated.

The difference between the rented bike count data plot before and after normalization is shown below in fig 6.1.1 and fig 6.1.2 respectively:

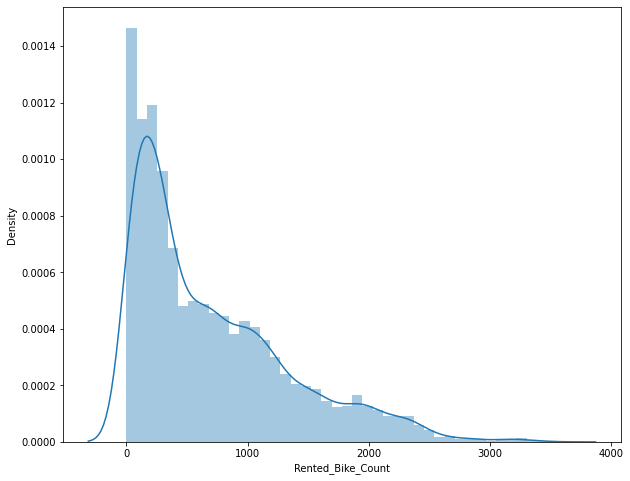
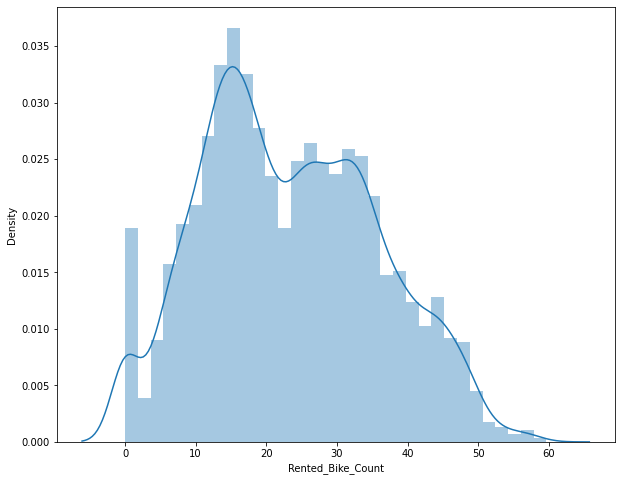
****

FIG [6.1.1]

**** FIG [6.1.2]

## 7 Building Machine Learning Algorithm

The provided data is first cleaned followed by exploratory data analysis and transformed using Feature Engineering. We then split the data into the Train set (for hyperparameter tuning) and Test set (for model evaluation).

Training is the most important step in machine learning. In training, you pass the prepared data to your machine learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

After training our model, we have to check to see how it’s performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that we split our data into earlier. If testing was done on the same data which in used for training, will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give disproportionately high accuracy.

Using MSE(Mean Squared Error) as our evaluation metrics, we compare various models and select the regression algorithm based on the lowest MSE on the Test data.

The final method used for submission is then obtained by training the selected Regression Algorithm on the entire Input Data set.

## train test split opencv python Train/Test Split

## 

we have two datasets.

One has independent features, called (x).

One has dependent variables, called (y).

First we should avoid the Overfitting or Underfitting. Then do the training and testing phase.

Here in our project,the train/test split was done as 75/25 % data with a random state of 42. The final data was of shape (8760, 16) which was split to (700,50) as Train data and (1752, 50) as Test data.

To normalize the data after the split, using the Min-Max Scalar module will give equal weightage to all the parameters to retain data from one-way deviation.

## 7.2 Linear Regression

## Linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression.

## This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

## In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

## On the basis of actual and predicted values using linear regression the below graph is plotted.

## 

Train errors,

MSE\_linear: 52.52693137394992

MAE\_linear: 5.582390382307961

RMSE\_linear: 7.247546576183551

R2\_linear : 0.6637858120748639

Adjusted R2\_linear : 0.6613102359097456

Test errors,

MSEtest\_linear: 54.96043918116387

MAEtest\_linear: 5.598781644509546

RMSEtest\_linear: 7.413530817442109

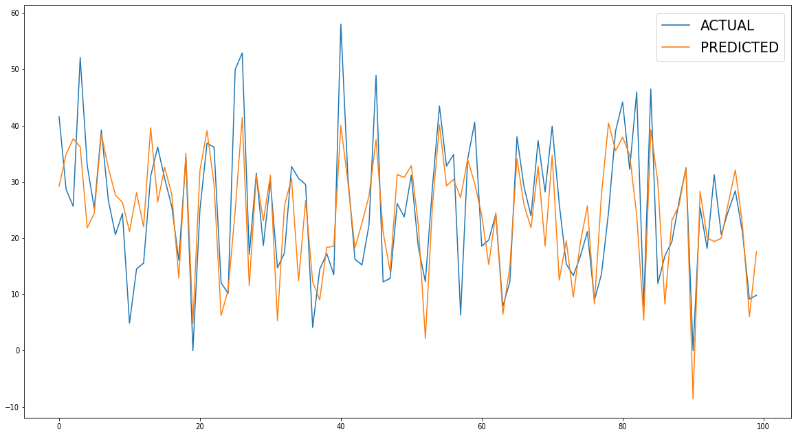
R2test\_linear: 0.6368545829042638

Adjusted\_R2test\_linear : 0.6341807096076546

## 7.3 LASSO Regression

The acronym “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator.

**Lasso regression** is a type of [linear regression](https://www.statisticshowto.com/probability-and-statistics/regression-analysis/find-a-linear-regression-equation/)that uses [shrinkage](https://www.statisticshowto.com/shrinkage-estimator/). Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multi collinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

On the basis of actual and predicted values using lasso regression the below graph is plotted.

On train dataset,

MSE\_lasso: 52.52708910162982

MAE\_lasso: 5.5827620347156905

RMSE\_lasso: 7.24755745762873

R2\_lasso : 0.6637848024921142

Adjusted\_R2\_lasso : 0.6613092188933447

On test dataset,

MSEtest\_lasso: 54.95571756718334

MAEtest\_lasso: 5.598772395072732

RMSEtest\_lasso: 7.4132123649052

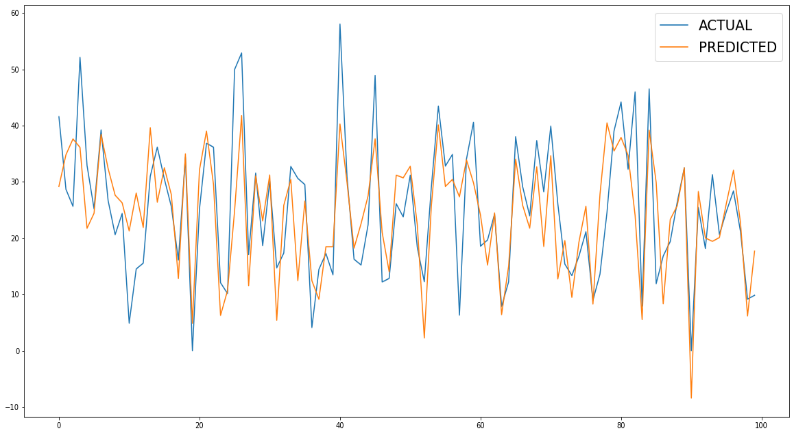
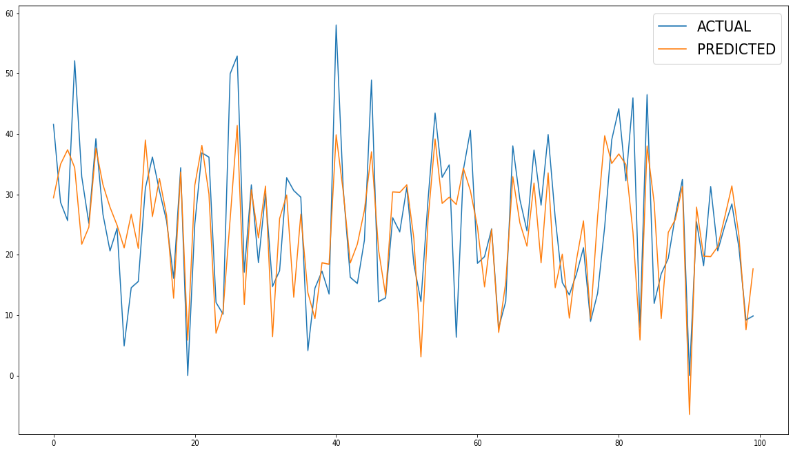
R2test\_lasso: 0.6368857804802632

Adjusted R2test\_lasso : 0.6342121368942919

## 7.4 Ridge Regression

Ridge [regression](https://www.mygreatlearning.com/blog/what-is-regression/) is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

The assumptions of ridge regression are the same as that of linear regression: linearity, constant variance, and independence. However, as ridge regression does not provide confidence limits, the distribution of errors to be normal need not be assumed.

 The same data set is then trained and tested using RIDGE regression and based on the values of the evaluation matrix the errors are calculated and the graph is plotted as shown.

On train set,

MSE\_ridge: 52.544149680845855

MAE\_ridge: 5.587902829067304

RMSE\_ridge: 7.24873435027425

R2\_ridge : 0.663675601199813

Adjusted R2\_ridge : 0.6611992135418272

On test set,

MSEtest\_ridge: 54.95706198219

MAEtest\_ridge: 5.60144447902748

RMSEtest\_ridge: 7.413303041302844

R2test\_ridge: 0.6368768973971655

Adjusted\_R2test\_ridge : 0.6342031884042316

## 

## 7.5 ElasticNet Regularization

**Elastic net** is a popular type of regularized linear regression that combines two popular penalties, specifically the L1 and L2 penalty functions.

## 

## Elastic Net is a middle ground between Ridge Regression and Lasso Regression. The regularization term is a simple mix of both Ridge and Lasso’s regularization terms and you can control the mix ratio ****r****. When r = 0, Elastic Net is equivalent to Ridge Regression, and when r = 1, it is equivalent to Lasso Regression

The same data set is then trained and tested using ElasticNet regression and based on the values of the evaluation matrix the errors are calculated and the graph is plotted as shown.

## 

## On train dataset,

MSE\_elasticnet: 53.26256523989348

MAE\_elasticnet: 5.641934303650608

RMSE\_elasticnet: 7.2981206649310

R2\_elasticnet : 0.6590771695484683

Adjusted R2\_elasticnet : 0.6565669232128841

On test dataset,

MSEtest\_elasticnet: 55.10118138663063

MAEtest\_elasticnet: 5.623537161348871

RMSEtest\_elasticnet: 7.423017000292444

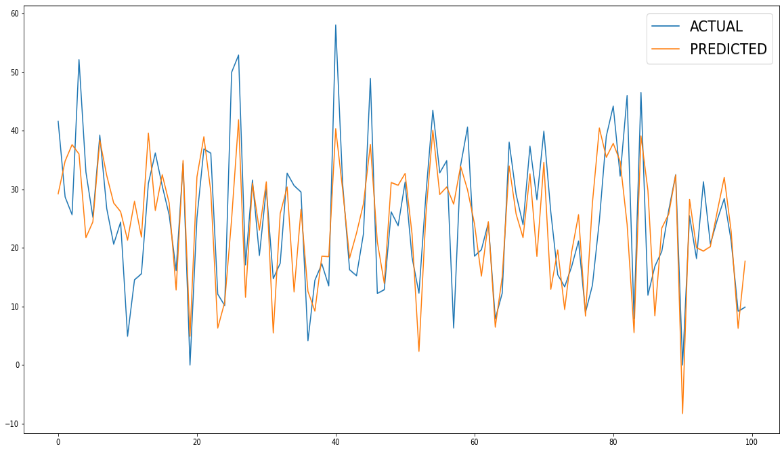
R2test\_elasticnet : 0.6359246433392117

Adjusted\_R2test\_elasticnet : 0.6332439228115667

## 7.5.1 Cross Validation on ElasticNet Regularization

Cross validation is a technique for assessing how the statistical analysis generalizes to an independent data set.It is a technique for evaluating machine learning model by training several models on the subsets of the available input data and evaluating them on the complementary subset of data.Using cross validation we there are high chances that we can detect overfitting with ease.

The same data set is then trained and tested using ElasticNet Cross Validation based on the values of the evaluation matrix the errors are calculated and the graph is plotted as shown.



In our model,

The best fit alpha value is : {'alpha': 0.01, 'l1\_ratio': 0.6}

Using {'alpha': 0.01, 'l1\_ratio': 0.6} the negative mean squared error is: -52.861049248319986

On train dataset,

MSE\_cvelasticnet: 52.56164148630267

MAE\_cvelasticnet: 5.590787693371766

RMSE\_cvelasticnet: 7.249940791917039

R2\_cvelasticnet : 0.6635636397161858

Adjusted\_R2\_cvelasticnet : 0.6610864276754398

On test dataset,

MSEtest\_cvelasticnet: 54.94722184538258

MAEtest\_cvelasticnet: 5.602709069007094

RMSEtest\_cvelasticnet: 7.412639330588166

R2test\_cvelasticnet : 0.6635636397161858

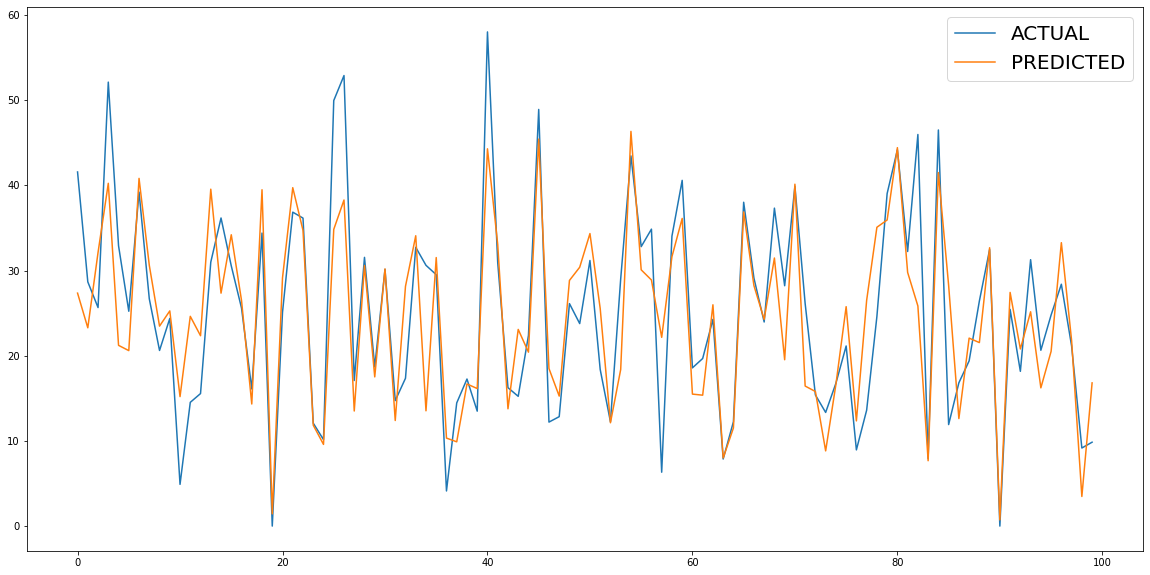
Adjusted\_R2test\_cvelasticnet : 0.6342686848216161

## 7.6 Polynomial Regression

In polynomial regression, the relationship between the independent variable x and the dependent variable y is described as an nth degree polynomial in x. Polynomial regression describes the fitting of a nonlinear relationship between the value of x and the conditional mean of y. It usually corresponded to the least-squares method.

This is a type of Linear Regression in which the dependent and independent variables have a curvilinear relationship and the polynomial equation is fitted to the data.Machine learning is also referred to as a subset of Multiple Linear Regression. Because we convert the Multiple Linear Regression equation into a Polynomial Regression equation by including more polynomial elements.

The same data set is then trained and tested using Polynomial Regression based on the values of the evaluation matrix the errors are calculated and the graph is plotted as shown.



Train errors,

MSE\_poly: 32.11998279550421

MAE\_poly: 4.194232933168463

RMSE\_poly: 5.667449408288018

R2\_poly : 0.7945391156154448

Adjusted\_R2\_poly : 0.7819209118067771

Test errors,

MSEtest\_poly: 43.065487324727954

MAEtest\_poly: 4.385647767139046

RMSEtest\_poly: 6.56242998627246

R2test\_poly : 0.7165302521568099

Adjusted\_R2test\_poly : 0.6991212009761372

R2 errors on the test data = 0.71 and training data = 0.79 are almost the same. So, we can conclude that the data on Polynomial regression model has not been overfitted. The Efficiency of this model shows a greater difference than the Linear regression.

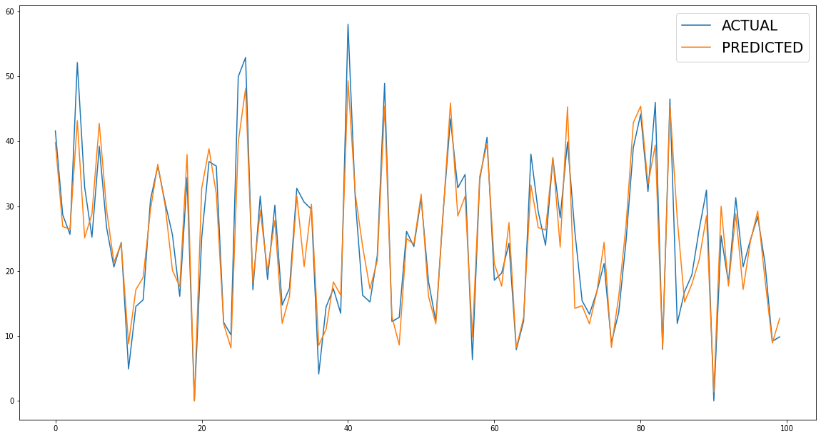
## 7.7 Random Forest Regressor

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification.

Random forest does not split tree nodes based on all variables; instead, it chooses random variable subsets as candidates to find the optimal split at every node of every tree. Then the information from the n trees is aggregated for classification and prediction. Random forests also provide the importance of each feature by accumulated Gini gains of all splits in all trees representing the variable discrimination ability.

The same data set is then trained and tested using Random Forest Regression based on the values of the evaluation matrix the errors are calculated and the graph is plotted as shown.



Here,we fit random forest model with criterion='mse',max\_depth=13,max\_leaf\_nodes=80, n\_estimators=180, random\_state=21.

For that, we got R2 of 0.9210 for training data and 0.90009 for test data, even the mean squared error is less as compared to linear regression and decision tree regressor .

Train errors,

MSE\_random: 12.331080726433425

MAE\_random: 2.564619664742169

RMSE\_random: 3.511563857661345

R2\_random : 0.9210712641261734

Adjusted\_R2\_random : 0.9204901045431171

Test errors,

MSEtest\_random: 14.984355312252857

MAEtest\_random: 2.758981618176812

RMSEtest\_random: 3.870963098797618

R2test\_random : 0.9009924221703873

Adjusted\_R2test\_random : 0.9002634202167409

Higher scores indicate higher importance given to the feature. For random forest regressor, Functioning Day and humidity has gotten highest importance

## 7.8 Decision Tree Regressor

## A Decision Tree is a supervised learning algorithm. It is a graphical representation of all the possible solutions. All the decisions were made based on some conditions. . It is a classification algorithm used for supervised learning.

## Following are the two ways:

**i) Gini Index:** Gini Index is the measure of impurity or the purity that is used in building a decision tree in the CART Algorithm.

**ii) Information Gain:** Information gain is the measure of how much information a feature gives about the class. It is the decrease in entropy after splitting the dataset based on the attribute.

Constructing a decision tree is all about finding the attribute that has the highest information gain.

The same data set is then trained and tested using Decision Tree Regression based

on the values of the evaluation matrix the errors are calculated and the graph is plotted as shown.

Train errors,

MSE\_dtree: 13.070675961683936

MAE\_dtree: 2.6260996582763205

RMSE\_dtree: 3.615338982956361

R2\_dtree : 0.8486002179004213

Adjusted R2\_dtree : 0.9113064563330937

Test errors,

MSEtest\_dtree: 13.070675961683936

MAEtest\_dtree: 2.6260996582763205

RMSEtest\_dtree: 3.615338982956361

R2test\_dtree : 0.8486296675657885

Adjusted\_R2test\_dtree:

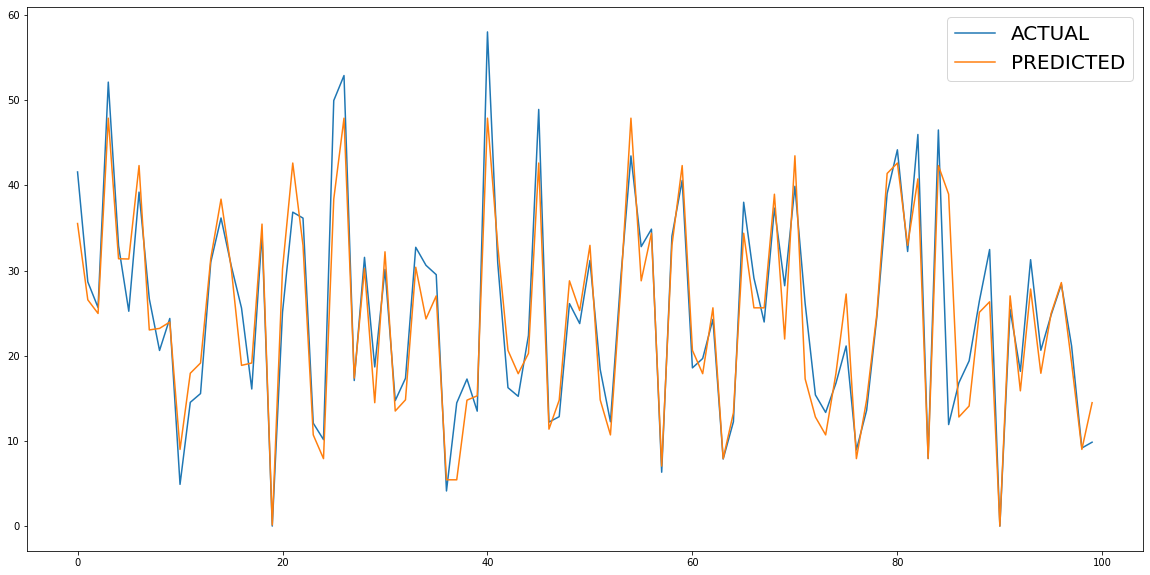
0.8393327190688187

R2 errors on the test data = 0.84 and training data = 0.91. So, we can conclude that the data on Decision Tree regression model has not been overfitted. The Efficiency of this model shows a better than polynomial regression.

The Decision Tree Regression Model seems to approximate the Rented Bike Count better than the Linear Regression Model and the Polynomial Regression Model.

But it dominates over Random Forest regressor.

In random forest we got the r2score on test dataset is 0.90009 which is better than that of the r2score on test dataset of Decision Tree regressor which is 0.8



# 8. Conclusion:

* Now, we have come to the end of our project Bike Sharing Demand Prediction. What we did let's take a short recap. We have found our dataset info where we have found there are 8760 rows and 14 columns with 13 features as independent and one as dependent according to our problem statement that is Rented bike count on which we have done our prediction. We didn’t find any null values or duplicates on the given dataset.
* Then we go for exploratory data analysis, in this section, we add some column and drop unnecessary column as per analysis of data like column ‘Date’. In addition to this we divided the dataset into numerical and categorical feature. Then we analyze the dataset by plotting graph for each independent feature with dependent feature.
* From this we concluded the followings.

1. Less demand of rented bike counts on winter season
2. Slightly Higher demand during Nonholidays
3. Almost no demand on Nonfunctioning days

* Now here comes the feature engineering,

in which we got to know that which feature gives more or less importance to dependent feature. Also in this part of study we got some conclusions. They are listed below :

1. The demand for rented bike from month 5 to 10 is higher than other months and these months fall in summer.
2. The demand for rented bike is less on winter seasons
3. Highest number of bike have rented in summer season
4. Almost equal percentage of bike rented in spring and autumn
5. Sligthly Higher demand during Non holidays
6. Almost no demand on Non functioning day
7. High increase of bikes rented between 8:00 am to 9:00 pm means people prefer rented bike during peak hours.
8. We can clearly see that the demand peaks at 8:00 AM and 6:00 PM, so we can say that there is a very high demand during the opening and closing hours of the office.

* Using box plot we found out there are some outliers in rented bike count which one is the dependent feature, then by using square root transformation we free it from outliers.
* In correlation analysis, we got that there is high correlation between Temperature and Dew Point Temperature.
* Last but not the least we go for model training in which we split the dataset into train dataset (for hyperparameter tuning) and test data set (for model evaluation) in

proportion of 75% train dataset and 25% test dataset.

* We have performed 8 models using these train and test data set out of which we concluded that Random Forest Regressor Model is best fit model to our dataset with mean squared error on test dataset is 14.98 and mean absolute error is 2.75 and r2score is 0.9.

**9. References:**

1.GeekforGeeks

2.Kaggle

3.Analytics Vidya

4.Statistics