**Airbnb Bookings Analysis- EDA**

**Upasana Kumari**

**Lonka ArunTeja**

**Zunaid,**

**Sukesh Shetty.**

**Data Science Trainees**

**Alma Better, Bangalore.**

**Abstract:**

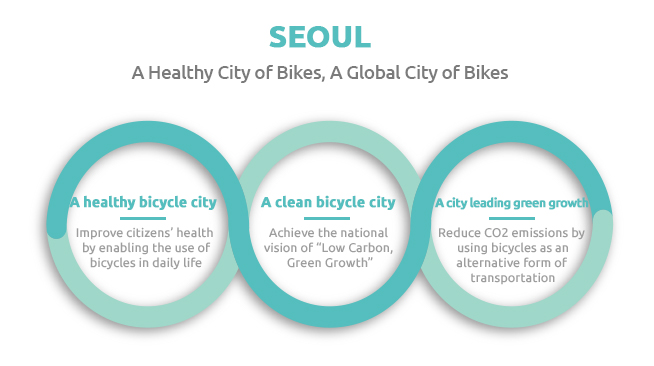
****

**Seoul Public Bike**

Seoul Public Bikes are designed to be used by all including women, the elderly and the infirm. Made of light-weight and durable materials, the bicycles prioritize driving stability and user convenience.

**Rental Stations** 

* Rental stations are installed by popular pedestrian areas, including subway entrances/exits, bus stops, residential complexes, public offices, schools, and banks.
* Rental stations are unmanned stands for the rental and return of bikes.
* Rental stations are installed in highly accessible areas near popular destinations.
* Users can rent and return bicycles at any rental station.
* **Docking Station**
* A docking station is a facility for parking bicycles. It has a lock that binds to the bicycle upon return.
* When renting a bicycle, separate the lock that is connected to the docking station from the bike terminal.



**1.Problem Statement**

* 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.
* The main goal is to create a prediction model that can be used to anticipate the number of bike rentals each hour based on weather conditions. As a result, it would be easier to anticipate fast and accurately.
* **Various Data collected:**

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

Attribute Information:

* 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 - NoFunc(Non Functional Hours), Fun(Functional hours)
* This Dataset contains 8760 lines and 14 columns.
* In a day we have 24 hours and we have 365 days a year so 365 multiplied by 24 = 8760, which represents the number of lines in the dataset. \*

## **Features description**

**Breakdown of Our Features:**

**Date**: *The date of the day, during 365 days from 01/12/2017 to 30/11/2018, formatting in DD/MM/YYYY, type: str*, we need to convert into datetime format.

**Rented Bike Count**: *Number of rented bikes per hour which our dependent variable and we need to predict that, type: int*

**Hour**: *The hour of the day, starting from 0-23 it's in a digital time format, type: int, we need to convert it into category data type.*

**Temperature(°C)**: *Temperature in Celsius, type: Float*

**Humidity (%)**: *Humidity in the air in %, type: int*

**Wind speed (m/s)**: *Speed of the wind in m/s, type: Float*

**Visibility (10m)**: *Visibility in m, type : int*

**Dew point temperature(°C)**: *Temperature at the beggining of the day, type : Float*

**Solar Radiation (MJ/m2)**: *Sun contribution, type : Float*

**Rainfall(mm)**: *Amount of raining in mm, type : Float*

**Snowfall (cm)**: *Amount of snowing in cm, type : Float*

**Seasons**: *Season of the year, type: str, there are only 4 seasons in data*.

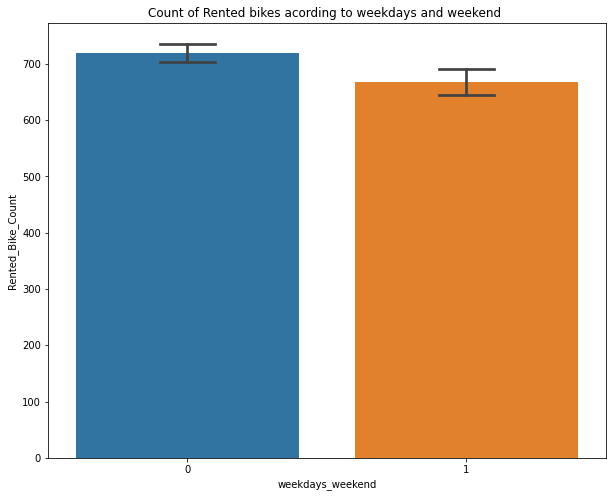
**Holiday**: *If the day is holiday period or not, type: str*

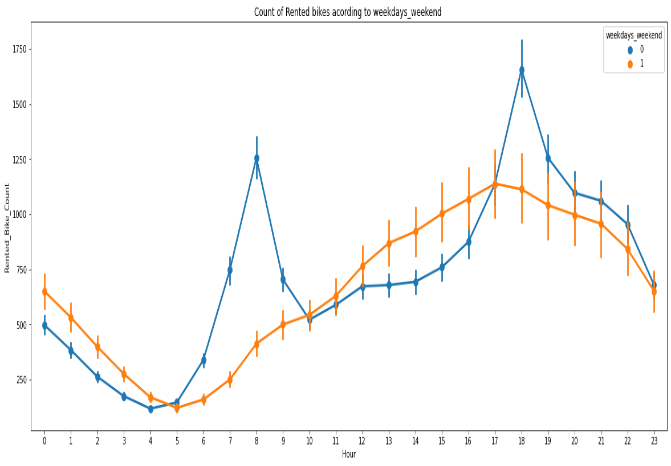
**Functioning Day**: *If the day is a Functioning Day or not, type: str*

we analyse our dependent variable, A dependent variable is a variable whose value will change depending on the value of another variable.

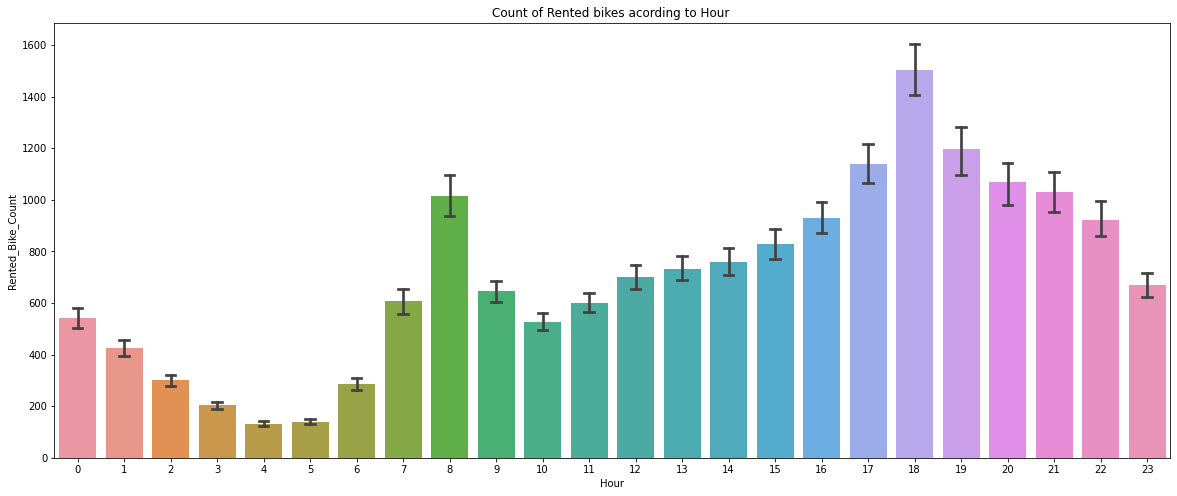
Analysation of categorical variables

Our dependent variable is "Rented Bike Count" so we need to analyze this column with the other columns by using some visualisation plot. First, we analyse the category data type then we proceed with the numerical data type.



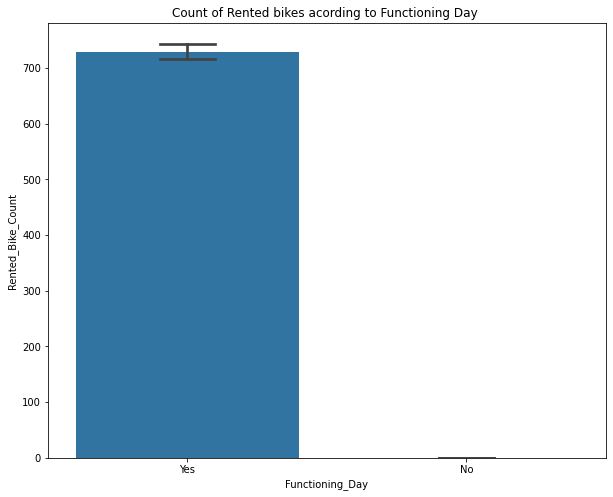
****

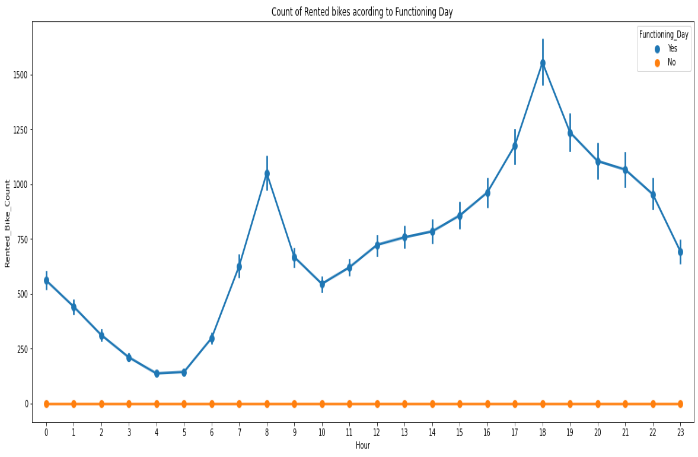
From the above point plot and bar plot we can say that in the week days which are represented in blue color show that the demand of the bike is higher because of the office. Peak Time are 7 am to 9 am and 5 pm to 7 pm The orange color represents the weekend days, and it show that the demand of rented bikes are very low especially in the morning hour but when the evening start from 4 pm to 8 pm the demand slightly increases.

****

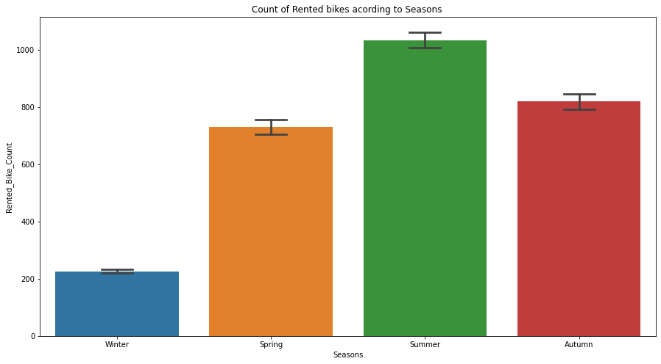
In the above plot which shows the use of rented bikes according to the hours and the data are from all over the year.

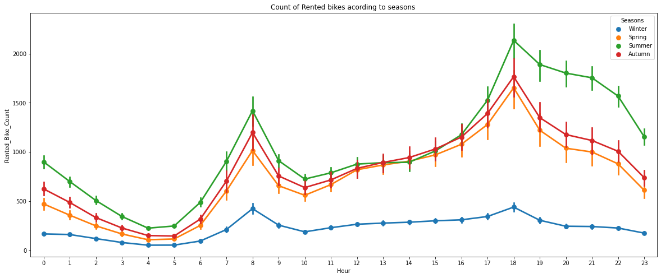
generally, people use rented bikes during their working hours from 7am to 9am and 5pm to 7pm.

****

****

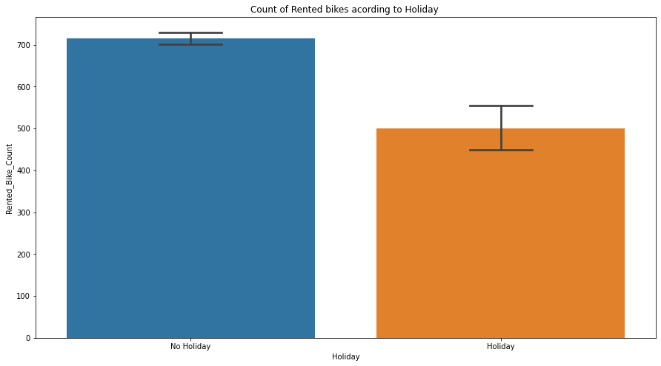
* In the above bar plot and point plot which shows the use of rented bike in functioning daya or not, and it clearly shows that,
* Peoples dont use reneted bikes in no functioning day.

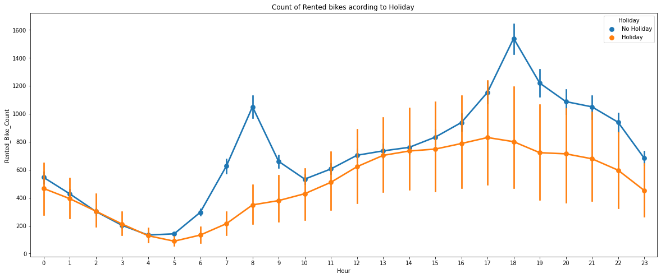




* In the above bar plot and point plot which shows the use of rented bike in in four different seasons, and it clearly shows that,
* In the summer season the use of rented bikes is high and peak time is 7am-9am and 7pm-5pm.
* In the winter season the use of rented bikes is very low because of snowfall***.***

#### **Holiday**



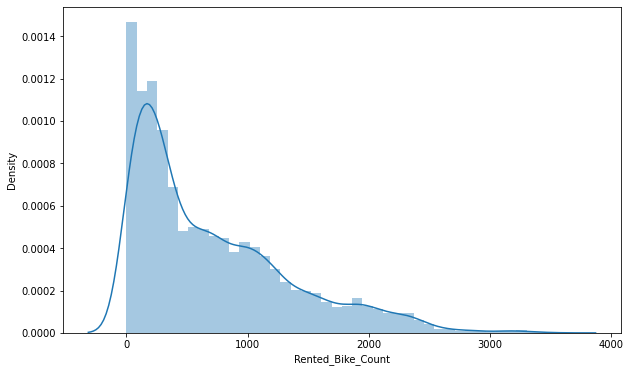
In the above bar plot and point plot which shows the use of rented bike in a holiday, and it clearly shows that,

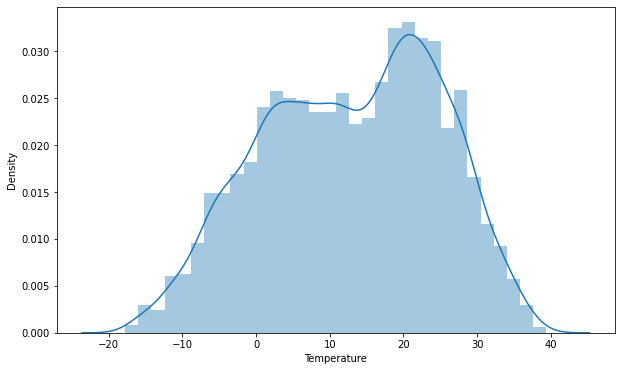
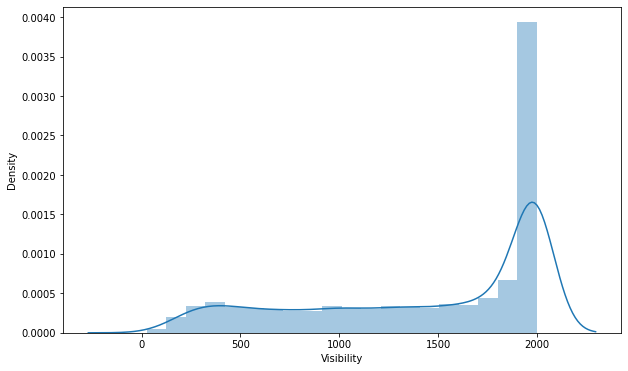
* plot shows that in holiday people uses the rented bike from 2pm-8pm

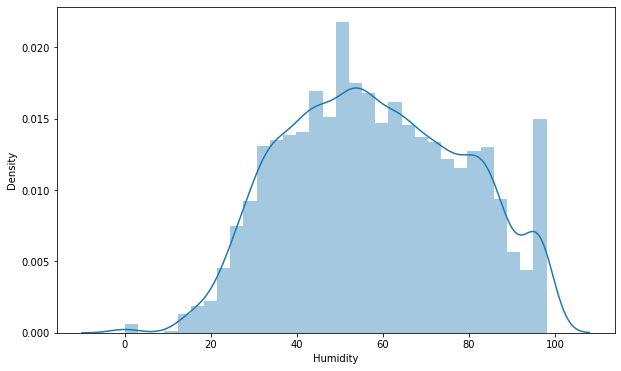
## Analyze of Numerical variables

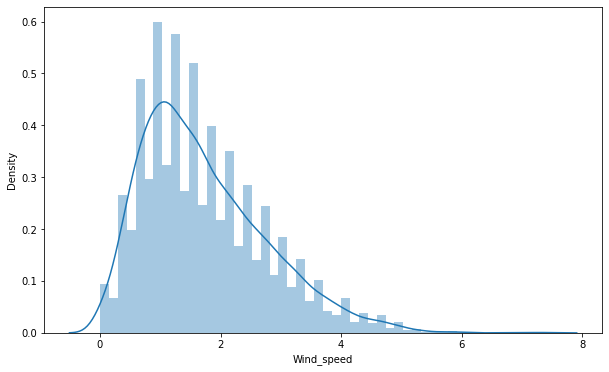
What is Numerical Data

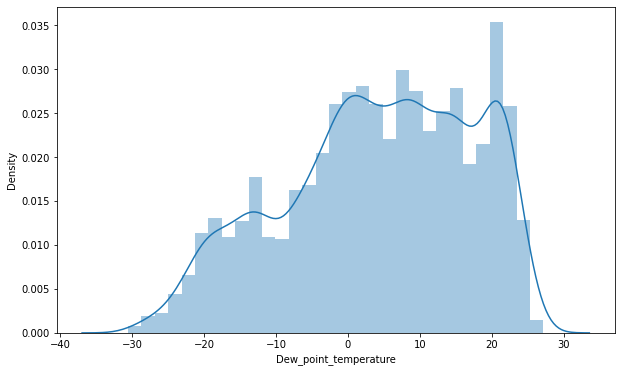
* *Numerical data is a data type expressed in numbers, rather than natural language description. Sometimes called quantitative data, numerical data is always collected in number form. Numerical data differentiates itself from other number form data types with its ability to carry out arithmetic operations with these numbers.*











#### 

#### **Regression plot**

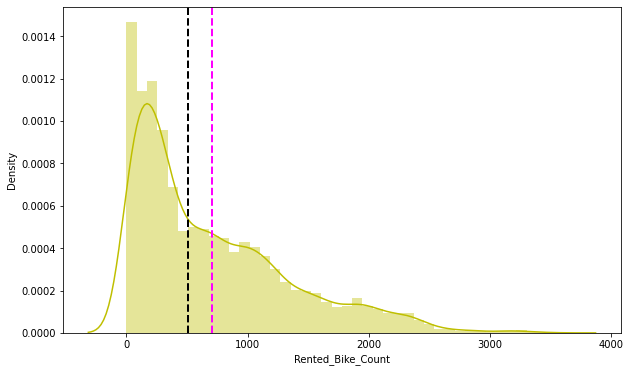
* The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses. Regression plots as the name suggests creates a regression line between 2 parameters and helps to visualize their linear relationships.

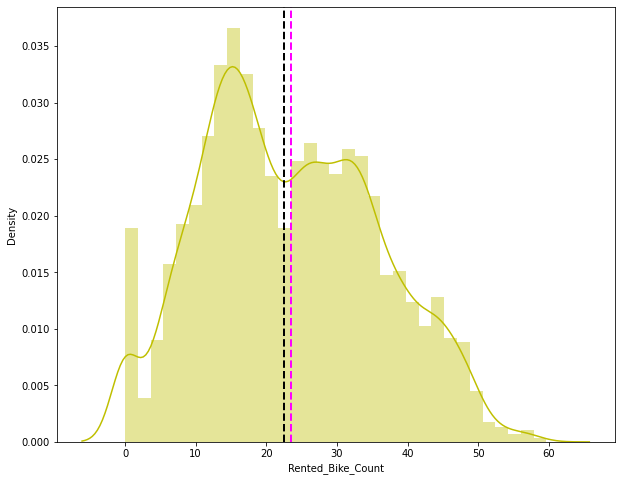
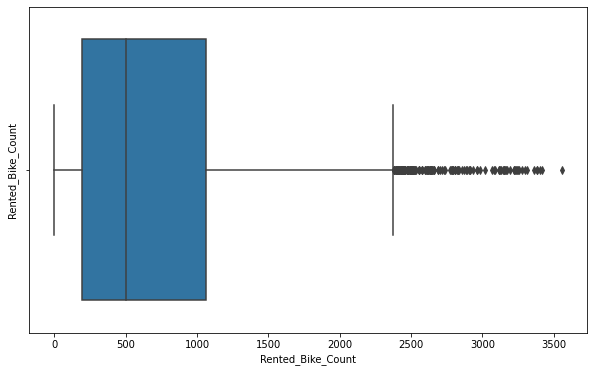
#### 

* From the above regression plot of all numerical features, we see that the columns 'Temperature', 'Wind\_speed','Visibility', 'Dew\_point\_temperature', 'Solar\_Radiation' are positively relation to the target variable.
* which means the rented bike count increases with increase of these features.
* 'Rainfall','Snowfall','Humidity' these features are negatively related with the target variable which means the rented bike count decreases when these features increase.

## Normalise Rented\_Bike\_Count column data

* The data normalization (also referred to as data pre-processing) is a basic element of data mining. It means transforming the data, namely converting the source data in to another format that allows processing data effectively. The main purpose of data normalization is to minimize or even exclude duplicated data



* ***The above graph shows that Rented Bike Count has moderate right skewness. Since the assumption of linear regression is that 'the distribution of dependent variables has to be normal', we should*** perform some operation to make it normal.
* Since we have a generic rule of applying Square root for the skewed variable in order to make it normal. After applying Square root to the skewed Rented Bike Count, here we get almost normal distribution.

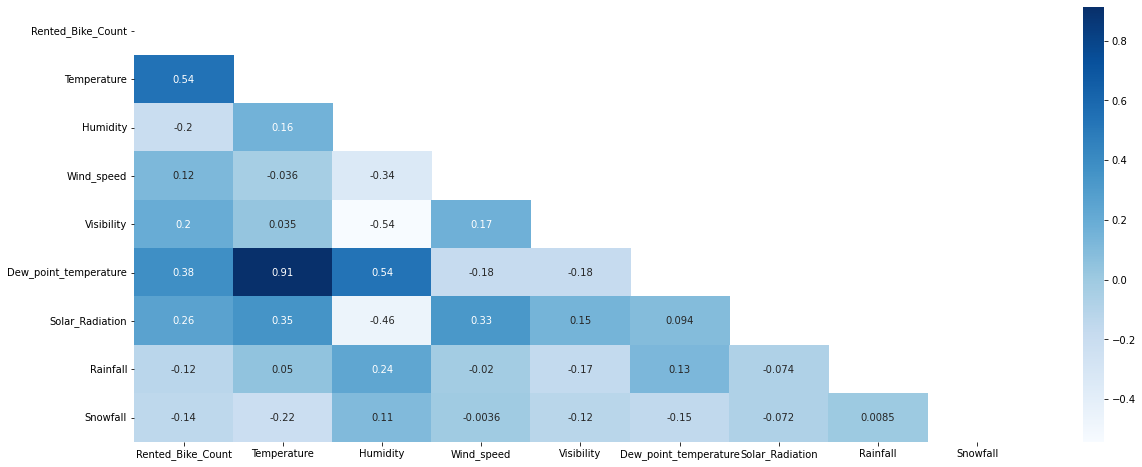
## Checking of Correlation between variables

### Checking in OLS Model

Ordinary least squares (OLS) regression is a statistical method of analysis that estimates the relationship between one or more independent variables and a dependent variable

* R square and Adj Square are near to each other. 40% of variance in the Rented Bike count is explained by the model.
* For the F statistic, P value is less than 0.05 for the 5% level of significance.
* P value of dew point temp and visibility are very high and they are not significant.
* Omnibus tests the skewness and kurtosis of the residuals. Here the value of Omnibus is high.. It shows we have skewness in our data.
* The condition number is large, 3.11e+04. This might indicate that there are strong multicollinearity or other numerical problems
* Durbin-Watson tests for autocorrelation of the residuals. Here the value is less than 0.5. We can say that there exists a positive autocorrelation among the variables.
* *From the OLS model we find that the 'Temperature' and 'Dew\_point\_temperature' are highly*correlated so we need to drop one of them.
* for dropping the we check the (P>|t|) value from above table and we can see that the 'Dew\_point\_temperature' value is higher so we need to drop Dew\_point\_temperature column
* For clarity, we use visualisation i.e heatmap in next step

### **Heatmap**

* we check correlation between variables using Correlation heatmap, it is graphical representation of correlation matrix representing correlation between different variables

We can observe on the heatmap that on the target variable line the most positively correlated variables to the rent are:

* the temperature
* the dew point temperature
* the solar radiation

***And most negatively correlated variables are:***

* Humidity
* Rainfall
* ***From the above correlation heatmap, we see that there is a positive*** correlation between columns 'Temperature' and 'Dew point temperature' i.e 0.91 so even if we drop this column then it dont affect the outcome of our analysis. And they have the same variations. so, we can drop the column 'Dew point temperature(°C)'.

## **Create the dummy variables**

A dataset may contain various types of values, sometimes it consists of categorical values. So, in-order to use those categorical values for programming efficiently we create dummy variables.

### **one hot encoding**

A one-hot encoding allows the representation of categorical data to be more expressive. Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical.

# **Model Training**

## **Train Test split for regression**

Before fitting any model it is a rule of thumb to split the dataset into a training and test set. This means some proportions of the data will go into training the model and some portion will be used to evaluate how our model is performing on any unseen data. The proportions may vary from 60:40, 70:30, 75:25 depending on the person but mostly used is 80:20 for training and testing respectively. In this step we will split our data into a training and testing set using scikit learn library.

* The mean squared error (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. It’s called the mean squared error as you’re finding the average of a set of errors. The lower the MSE, the better the forecast.
* MSE formula = (1/n) \* Σ(actual – forecast)2 Where:
* n = number of items,
* Σ = summation notation,
* Actual = original or observed y-value,
* Forecast = y-value from regression.
* Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors).
* Mean Absolute Error (MAE) are metrics used to evaluate a Regression Model. ... Here, errors are the differences between the predicted values (values predicted by our regression model) and the actual values of a variable.
* R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

# LINEAR REGRESSION

Regression models describe the relationship between variables by fitting a line to the observed data. Linear regression models use a straight line

Linear regression uses a linear approach to model the relationship between independent and dependent variables. In simple words it’s a best fit line drawn over the values of independent variables and dependent variables. In the case of a single variable, the formula is the same as a straight line equation having an intercept and slope.

y\_pred=β0+β1x

whereβ0 and β1are intercept and slope respectively.

In case of multiple features, the formula translates into:

y\_pred=β0+β1x1+β2x2+β3x3+.....

where x\_1,x\_2,x\_3 are the features values andβ0,β1,β2.....are weights assigned to each of the features. These become the parameters which the algorithm tries to learn using Gradient descent.

Gradient descent is the process by which the algorithm tries to update the parameters using a loss function. Loss function is nothing but the difference between the actual values and predicted values (aka error or residuals). There are different types of loss function but this is the simplest one. Loss function summed over all observations gives the cost function. The role of gradient descent is to update the parameters till the cost function is minimized i.e., a global minimum is reached. It uses a hyperparameter 'alpha' that gives a weightage to the cost function and decides on how big the steps to take. Alpha is called the learning rate. It is always necessary to keep an optimal value of alpha as high and low values of alpha might make the gradient descent overshoot or get stuck at a local minima. There are also some basic assumptions that must be fulfilled before implementing this algorithm. They are:

1. No multicollinearity in the dataset.
2. Independent variables should show linear relationship with dv.
3. Residual mean should be 0 or close to 0.
4. There should be no heteroscedasticity i.e., variance should be constant along the line of best fit.

Let us now implement our first model. We will be using Linear Regression from the scikit library.

MSE : 35.07751288189293

RMSE : 5.9226271942350825

MAE : 4.474024092996787

R2 : 0.7722101548255267

Adjusted R2 : 0.7672119649454145

**Looks like our r2 score value is 0.77 that means our model is able to capture most of the data variance. Let's save it in a data frame for later comparisons.**

MSE : 33.27533089591926

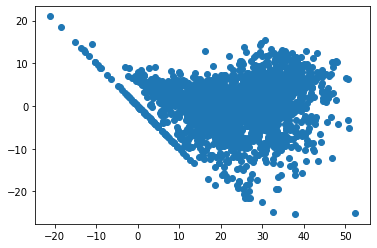
RMSE : 5.76847734639907

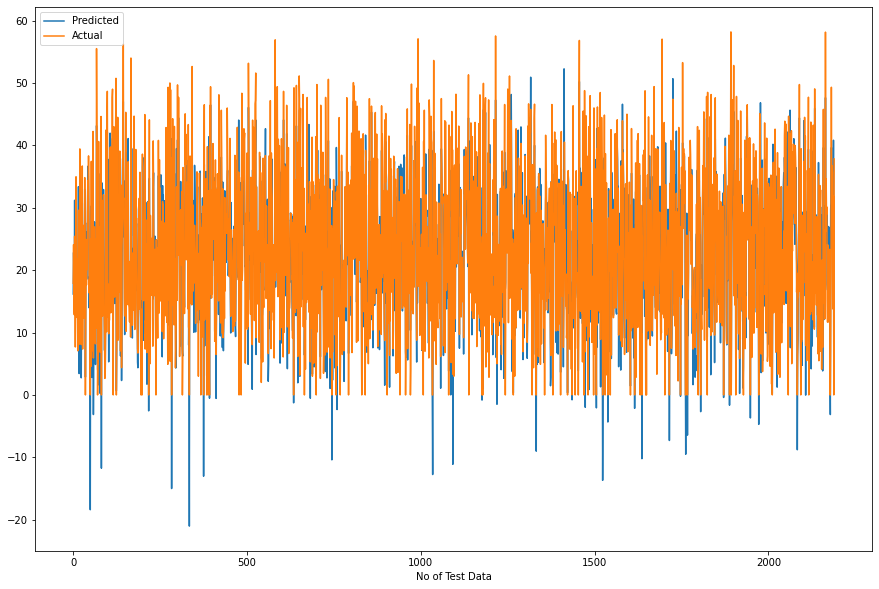
MAE : 4.410178475318181

R2 : 0.7893518482962683

Adjusted R2 : 0.7847297833429184

**The r2\_score for the test set is 0.78. This means our linear model is performing well on the data. Let us try to visualize our residuals and see if there is heteroscedasticity (unequal variance or scatter).**





# LASSO REGRESSION

MSE : 91.59423336097032

RMSE : 9.570487623991283

MAE : 7.255041571454952

R2 : 0.40519624904934015

Adjusted R2 : 0.3921449996120475

**Looks like our r2 score value is 0.40 that means our model is not able to capture most of the data variance. Let's save it in a data frame for later comparisons.**

MSE : 96.7750714044618

RMSE : 9.837432155011886

MAE : 7.455895061963607

R2 : 0.3873692800799008

Adjusted R2 : 0.37392686932535146

**The r2\_score for the test set is 0.38. This means our linear model is not performing well on the data. Let us try to visualize our residuals and see if there is heteroscedasticity (unequal variance or scatter).**

# RIDGE REGRESSION

MSE : 35.07752456136463

RMSE : 5.922628180239296

MAE : 4.474125776125378

R2 : 0.7722100789802107

Adjusted R2 : 0.7672118874358922

**Looks like our r2 score value is 0.77 that means our model is able to capture most of the data variance. Let's save it in a data frame for later comparisons.**

MSE : 33.27678426818438

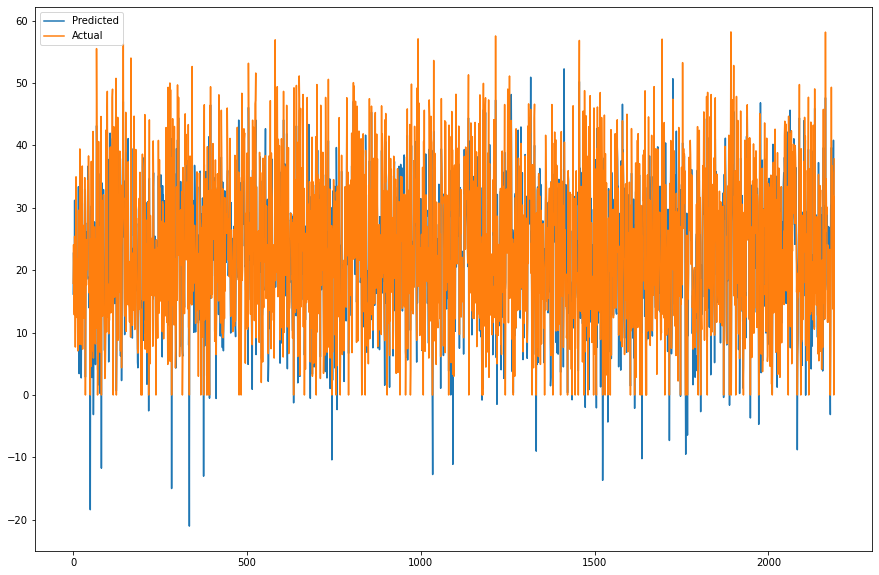
RMSE : 5.768603320404722

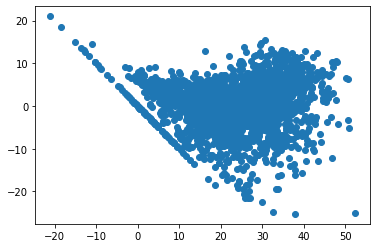
MAE : 4.410414932539515

R2 : 0.7893426477812578

Adjusted R2 : 0.7847203809491939

**The r2\_score for the test set is 0.78. This means our linear model is performing well on the data. Let us try to visualize our residuals and see if there is heteroscedasticity (unequal variance or scatter).**





# ELASTIC NET REGRESSION

MSE : 57.5742035398887

RMSE : 7.587766703048315

MAE : 5.792276538970546

R2 : 0.6261189054494012

Adjusted R2: 0.6179151652795234

**Looks like our r2 score value is 0.62 that means our model is able to capture most of the data variance. Let’s save it in a data frame for later comparisons.**

MSE : 59.45120536350042

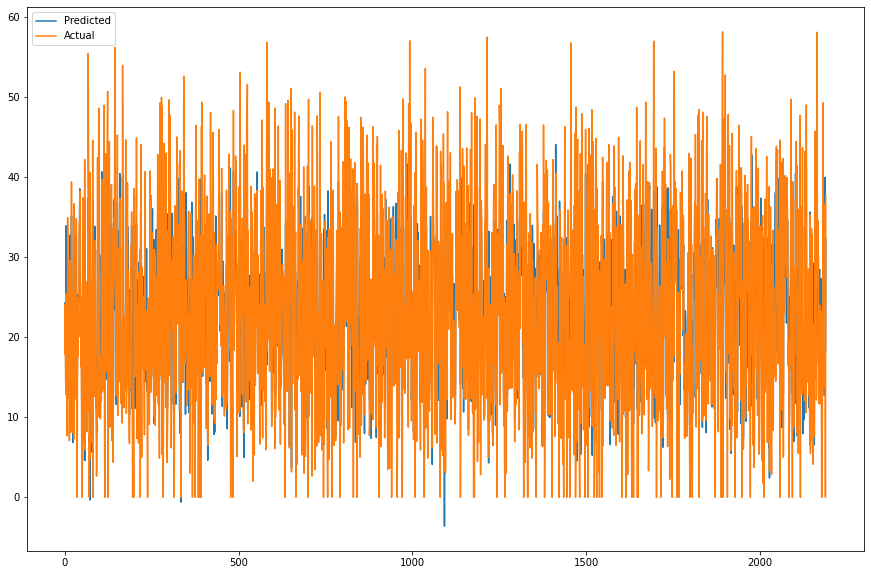
RMSE : 7.710460775044538

MAE : 5.873612334800099

R2 : 0.6236465216363589

Adjusted R2 : 0.6153885321484546

**The r2\_score for the test set is 0.86. This means our linear model is performing well on the data. Let us try to visualize our residuals and see if there is heteroscedasticity (unequal variance or scatter).**



# DECISION TREE

Model Score: 0.6420186924582427

MSE : 55.1257845456407

RMSE : 7.4246740363224495

MAE : 5.3866040021161785

R2 : 0.6420186924582427

Adjusted R2 : 0.6341638271667103

**Looks like our r2 score value is 0.69 that means our model is able to capture most of the data variance. Let's save it in a dataframe for later comparisons.**

MSE : 68.616666648556

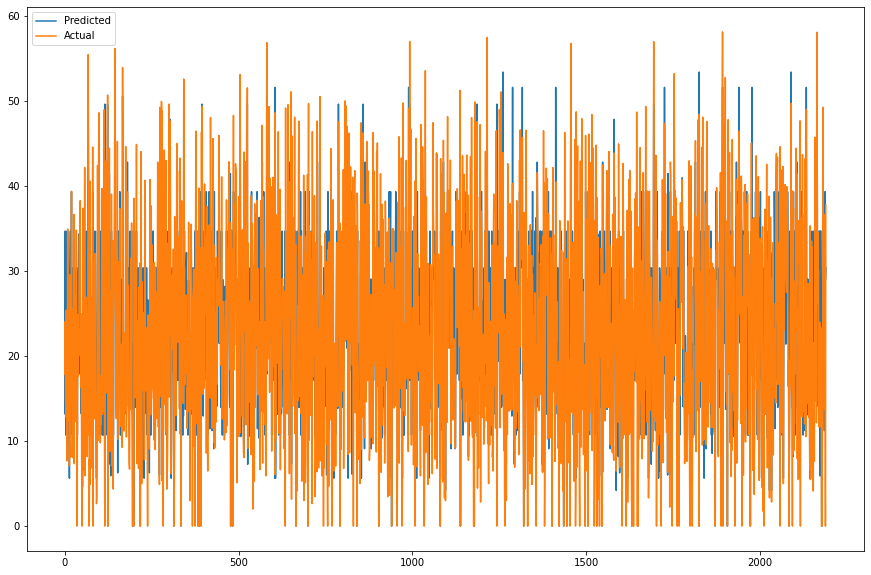
RMSE : 8.28351777015997

MAE : 5.89079554333329

R2: 0.5656249354574567

Adjusted R2: 0.5560938299329472

**The r2\_score for the test set is 0.66. This means our linear model is performing well on the data. Let us try to visualize our residuals and see if there is heteroscedasticity (unequal variance or scatter).**



# RANDOM FOREST

Model Score: 0.9897545829621024

MSE : 1.5776987242427414

RMSE : 1.2560647770886426

MAE : 0.8014773366158234

R2 : 0.9897545829621024

Adjusted R2 : 0.9895297768926434

**Looks like our r2 score value is 0.98 that means our model is able to capture most of the data variance. Let's save it in a dataframe for later comparisons.**

MSE : 12.804193712672028

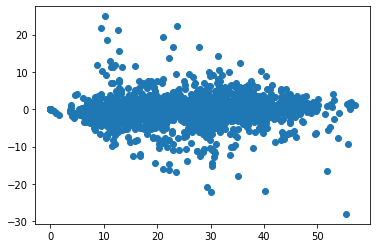
RMSE: 3.578294805165168

MAE: 2.2015707435525167

R2: 0.9189435636848415

Adjusted R2: 0.9171650144286265

**The r2\_score for the test set is 0.91. This means our linear model is performing well on the data. Let us try to visualize our residuals and see if there is heteroscedasticity (unequal variance or scatter).**



Feature Feature Importance

0 Temperature 0.32

1 Humidity 0.16

34 Functioning\_Day\_Yes 0.15

10 Hour\_4 0.03

4 Solar\_Radiation 0.03

5 Rainfall 0.03

24 Hour\_18 0.03

11 Hour\_5 0.03

25 Hour\_19 0.02

46 weekdays\_weekend\_1 0.02

9 Hour\_3 0.02

3 Visibility 0.01

44 month\_11 0.01

2 Wind\_speed 0.01

32 Seasons\_Winter 0.01

8 Hour\_2 0.01

27 Hour\_21 0.01

26 Hour\_20 0.01

28 Hour\_22 0.01

23 Hour\_17 0.01

12 Hour\_6 0.01

14 Hour\_8 0.01

13 Hour\_7 0.00

45 month\_12 0.00

43 month\_10 0.00

42 month\_9 0.00

41 month\_8 0.00

40 month\_7 0.00

39 month\_6 0.00

38 month\_5 0.00

37 month\_4 0.00

36 month\_3 0.00

35 month\_2 0.00

19 Hour\_13 0.00

20 Hour\_14 0.00

31 Seasons\_Summer 0.00

30 Seasons\_Spring 0.00

29 Hour\_23 0.00

15 Hour\_9 0.00

16 Hour\_10 0.00

17 Hour\_11 0.00

18 Hour\_12 0.00

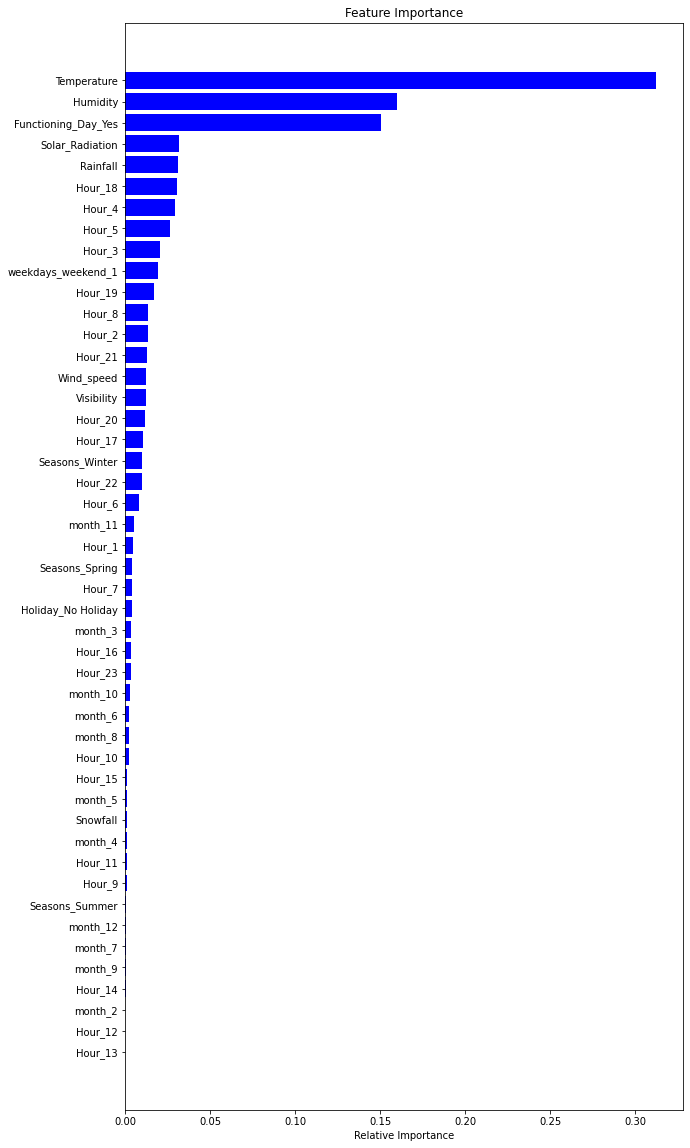
6 Snowfall 0.00

7 Hour\_1 0.00

22 Hour\_16 0.00

21 Hour\_15 0.00

33 Holiday\_No Holiday 0.00



# GRADIENT BOOSTING

Model Score: 0.8789016499095265

MSE: 18.648017131847936

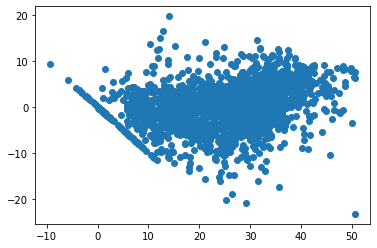
RMSE: 4.318334995324927

MAE: 3.2690035692731247

R2: 0.8789016499095265

Adjusted R2: 0.8762444965695394

**Looks like our r2 score value is 0.87 that means our model is able to capture most of the data variance. Let’s save it in a data frame for later comparisons.**



Feature Feature Importance

0 Temperature 0.32

34 Functioning\_Day\_Yes 0.17

1 Humidity 0.13

5 Rainfall 0.07

4 Solar Radiation 0.05

32 Seasons Winter 0.03

24 Hour\_18 0.03

10 Hour\_4 0.03

11 Hour\_5 0.02

27 Hour\_21 0.02

25 Hour\_19 0.02

46 weekdays\_weekend\_1 0.02

9 Hour\_3 0.02

12 Hour\_6 0.01

44 month\_11 0.01

43 month\_10 0.01

28 Hour\_22 0.01

26 Hour\_20 0.01

8 Hour\_2 0.01

23 Hour\_17 0.01

14 Hour\_8 0.01

20 Hour\_14 0.00

2 Wind\_speed 0.00

45 month\_12 0.00

15 Hour\_9 0.00

42 month\_9 0.00

41 month\_8 0.00

40 month\_7 0.00

39 month\_6 0.00

38 month\_5 0.00

37 month\_4 0.00

36 month\_3 0.00

35 month\_2 0.00

33 Holiday\_No Holiday 0.00

21 Hour\_15 0.00

3 Visibility 0.00

31 Seasons\_Summer 0.00

30 Seasons\_Spring 0.00

13 Hour\_7 0.00

16 Hour\_10 0.00

6 Snowfall 0.00

17 Hour\_11 0.00

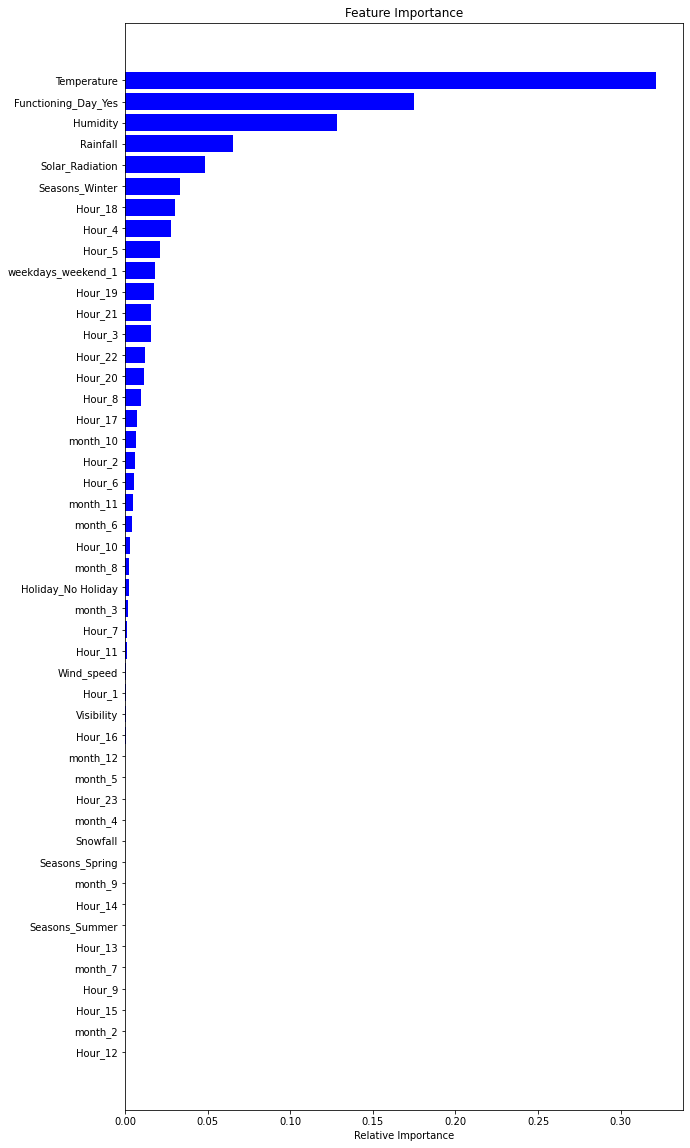
7 Hour\_1 0.00

18 Hour\_12 0.00

19 Hour\_13 0.00

22 Hour\_16 0.00

29 Hour\_23 0.00



# Hyperparameter tuning

Before proceeding to try the next models, let us try to tune some hyperparameters and see if the performance of our model improves.

Hyperparameter tuning is the process of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a model argument whose value is set before the learning process begins. The key to machine learning algorithms is hyperparameter tuning.

**Using GridSearchCV**

GridSearchCV helps to loop through predefined hyperparameters and fit the model on the training set. So, in the end, we can select the best parameters from the listed hyperparameters.

Feature Feature Importance

0 Temperature 0.31

34 Functioning\_Day\_Yes 0.16

1 Humidity 0.15

4 Solar\_Radiation 0.04

5 Rainfall 0.04

10 Hour\_4 0.03

24 Hour\_18 0.03

32 Seasons\_Winter 0.02

25 Hour\_19 0.02

11 Hour\_5 0.02

46 weekdays\_weekend\_1 0.02

9 Hour\_3 0.02

8 Hour\_2 0.01

43 month\_10 0.01

41 month\_8 0.01

2 Windspeed 0.01

3 Visibility 0.01

12 Hour\_6 0.01

28 Hour\_22 0.01

27 Hour\_21 0.01

26 Hour\_20 0.01

23 Hour\_17 0.01

14 Hour\_8 0.01

22 Hour\_16 0.00

35 month\_2 0.00

45 month\_12 0.00

44 month\_11 0.00

13 Hour\_7 0.00

42 month\_9 0.00

40 month\_7 0.00

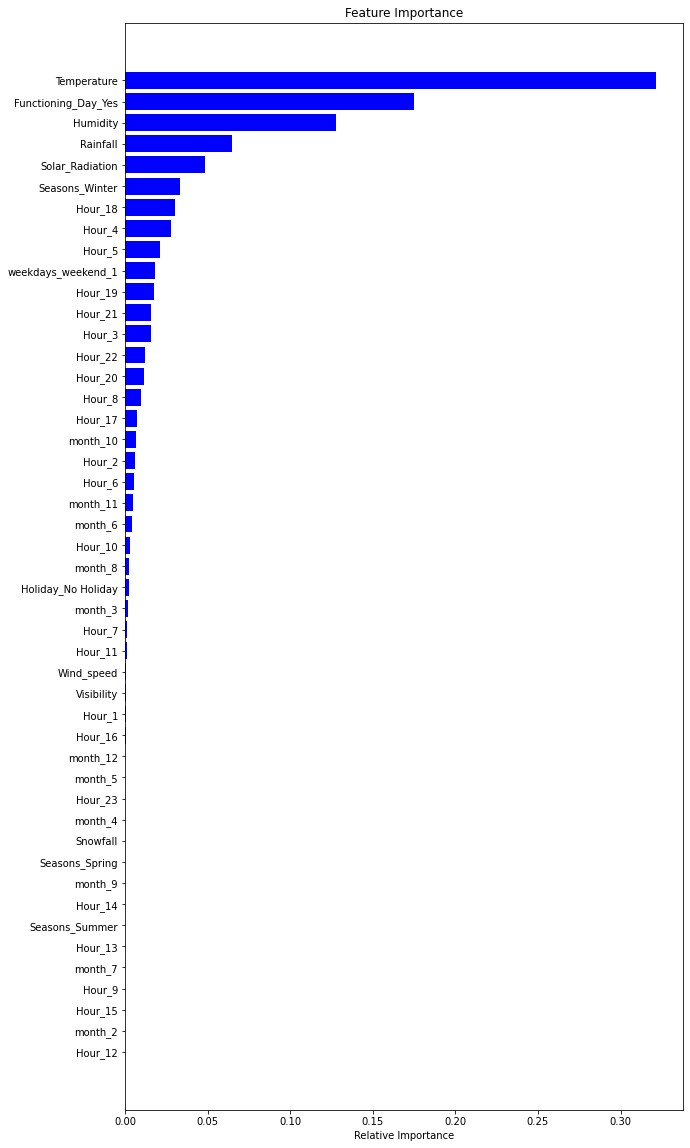
39 month\_6 0.00

38 month\_5 0.00

37 month\_4 0.00

36 months\_3 0.00

15 Hour\_9 0.0



**CONCLUSION**

During the time of our analysis, we initially did EDA on all the features of our dataset. We first analysed our dependent variable, 'Rented Bike Count' and also transformed it. Next, we analysed categorical variables and dropped the variable who had the majority of one class. We also analysed numerical variables, found out the correlation, distribution and their relationship with the dependent variable. We also removed some numerical features who had mostly 0 values and hot encoded the categorical variables.

Next we implemented 7 machine learning algorithms: Linear Regression, lasso, ridge, elastic net, decision tree, Random Forest and XGBoost. We did hyperparameter tuning to improve our model performance. The results of our evaluation are:

• No overfitting is seen.

• Random forest Regressor and GB gridsearchcv gives the highest R2 score of 99% and 95% respectively for the Train Set and 92% for Test set.

• Feature Importance value for Random Forest and Gradient Boost are different.

• We can deploy this model.

However, this is not the ultimate end. As this data is time dependent, the values for variables like temperature, windspeed, solar radiation etc., will not always be consistent. Therefore, there will be scenarios where the model might not perform well. As Machine learning is an exponentially evolving field, we will have to be prepared for all contingencies and also keep checking our model from time to time. Therefore, having a quality knowledge and keeping pace with the ever-evolving ML field would surely help one to stay a step ahead in future.