Capstone Project-2

Bike Sharing Demand Prediction

(Supervised Machine Learning regression)

BY

Sk Samim Ali,

Mohd. Izhar,

Sarath Haridas

(Cohort – Florence)

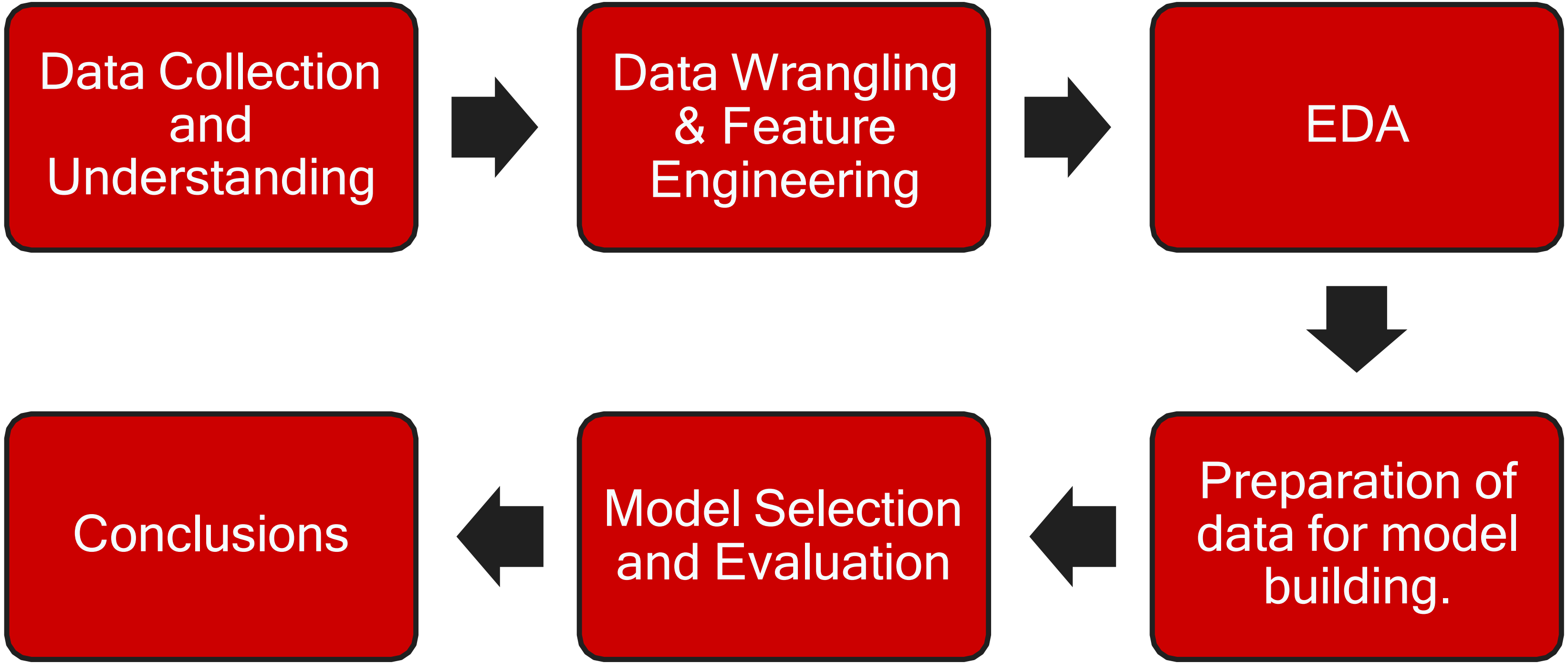
Problem Statement:

- Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. The client is Seoul Bike, who participates in a bike share program in Seoul, South Korea. An accurate prediction of bike count is critical to the success of the Seoul bike share program. It is important to make the rental bikes 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 main objective of this project is the prediction of the bike count required at each hour for the stable supply of rental bikes.

Work Flow :

- So we will divide our work flow into the following steps.



Data Collection and Understanding:

AI

- We had Seoul Bike Data for our analysis and model building
- The dataset contains weather information (Temperature, Humidity, Wind speed, Visibility, Dew point, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.
- In this we had total 8760 observations and 14 features including target variable.

Data Description:

Date : year-month-day.

Hour - Hour of the day.

Temperature-Temperature in Celsius.

Humidity - %.

Wind speed - m/s.

Visibility - m.

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).

Rented Bike count - Count of bikes rented at each hour (Target Variable i.e Y variable).

Data Wrangling and Feature Engineering:

As we know we had 8760 observations and 14 features.

- **Categorical Features:** Seasons, Holiday and Functioning day.
- **Numerical Columns:**

Date, Hour, Temperature, Humidity, Wind speed, Visibility, Dew point temperature, Solar radiation, Rainfall, Snowfall, Rented Bike count .

Rename Columns: We renamed columns because they had units mentioned in brackets and was difficult to copy the feature name while working.

- **Since the variables having units with name, so renaming columns for better variable analysis**

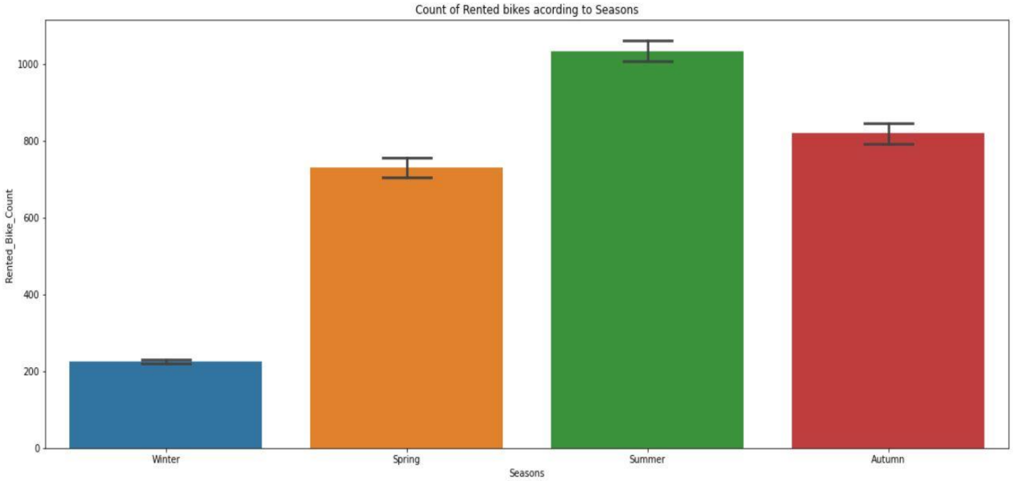
[13]
```
#Renaming some columns name
bike_df = bike_df.rename(columns = {'Rented Bike Count' : 'Rented_Bike_Count',
                                    'Temperature(°C)':'Temperature',
                                    'Humidity(%)':'Humidity',
                                    'Wind speed (m/s)': 'Wind_speed',
                                    'Visibility (10m)':'Visibility',
                                    'Dew point temperature(°C)': 'Dew_point_temperature',
                                    'Solar Radiation (MJ/m2)': 'Solar_Radiation',
                                    'Rainfall(mm)':'Rainfall',
                                    'Snowfall (cm)':'Snowfall',
                                    'Functioning Day':'Functioning_Day'})
```

Data Wrangling and Feature Engineering:

- We had zero null values in our dataset.
- Zero Duplicate entries were found.
- We changed the data type of Date column from ‘object’ to ‘datetime64[ns]’. This was done for feature engineering.
- We created two new columns with the help of Date column ‘Month’ and ‘Day’. Which were further used for EDA. And later we dropped Date column.

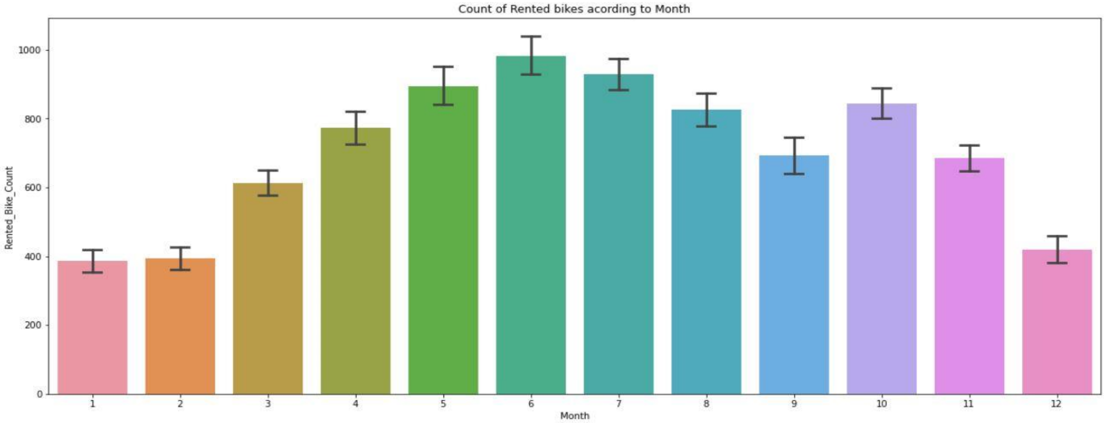
```
# Changing the "Date" column into three "year","month","day" column
bike_df['Date']=bike_df['Date'].astype('datetime64[ns]')

bike_df['year'] = bike_df['Date'].dt.year
bike_df['Month'] = bike_df['Date'].dt.month
bike_df['day'] = bike_df['Date'].dt.day_name()
```

EDA (Exploratory Data Analysis):



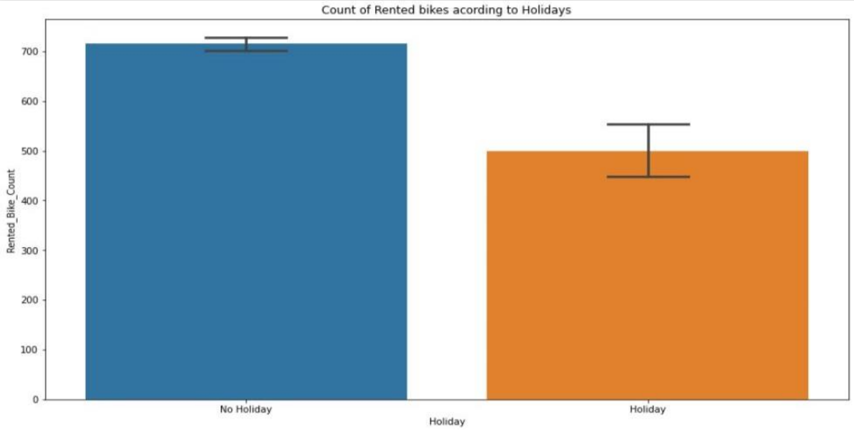
Relation of rented bike count with categorical features:

Summer season had the highest rented bike count. People are more likely to take rented bikes in summer. Bike rentals in winter is very less compared to other seasons.



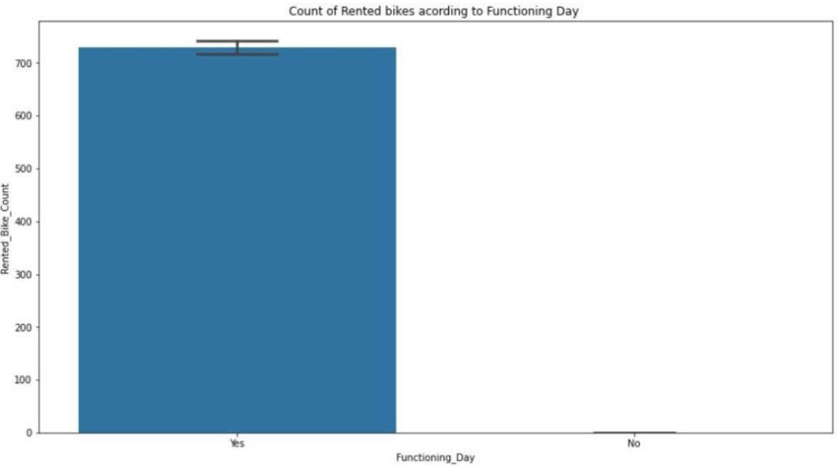
From March count of rented bikes started increasing and it was highest in June.

EDA (Exploratory Data Analysis):



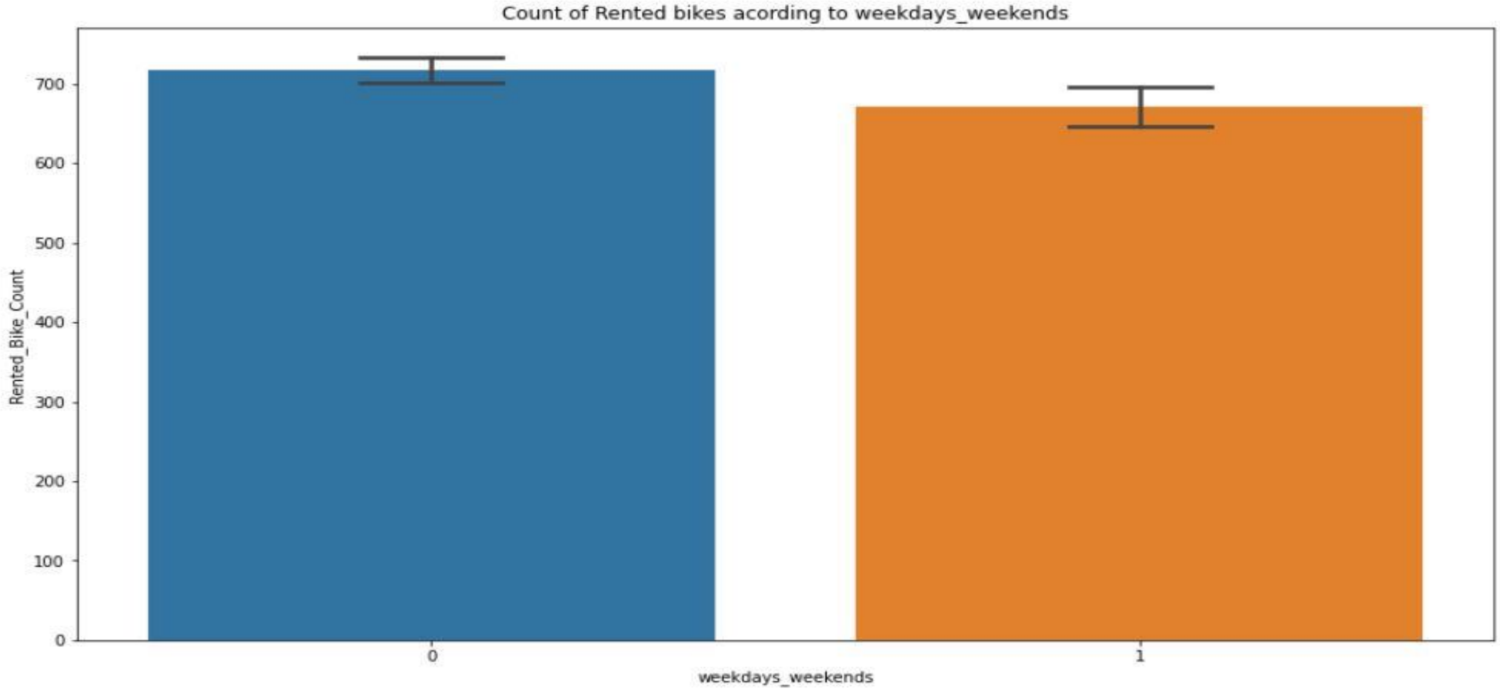
Conclusions:

Higher number of bikes were rented on Non-Holidays. Which is almost 700 bikes.



Negligible number of bikes were rented on non-functioning day. More than 700 bikes rented on functioning day

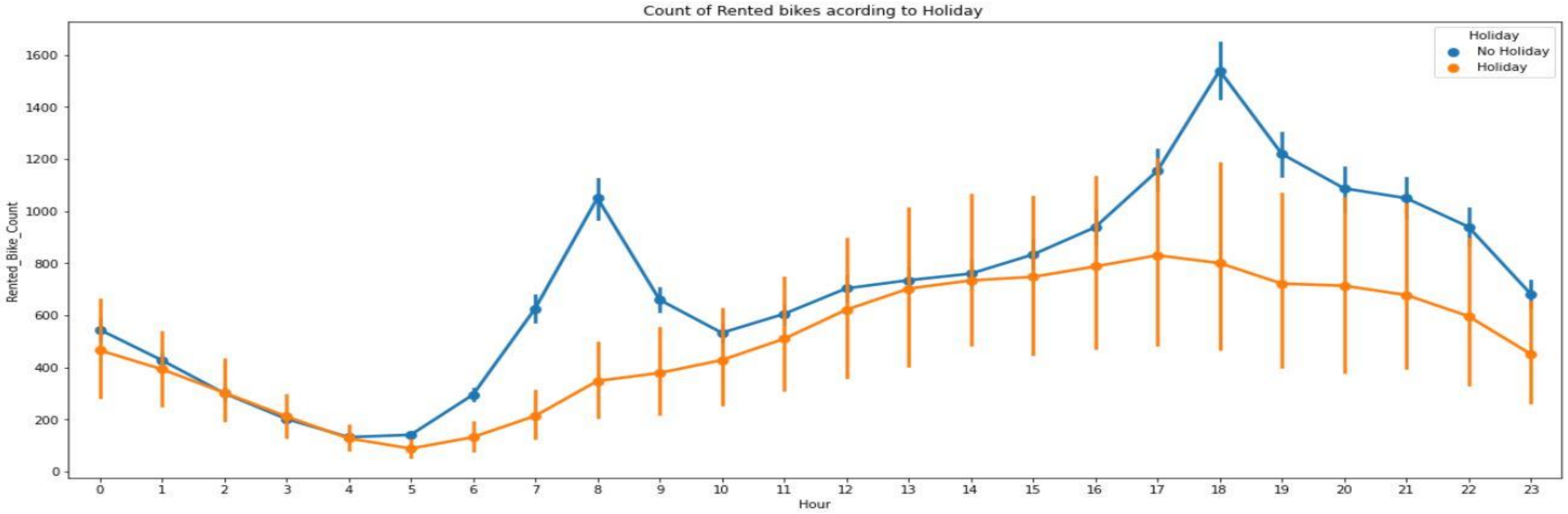
EDA (Exploratory Data Analysis):



Observation:
More than 700 bikes were rented on weekdays. On weekends, almost 650 bikes were rented.

EDA (Exploratory Data Analysis):

The trend of rented bike count according to hours in different scenarios



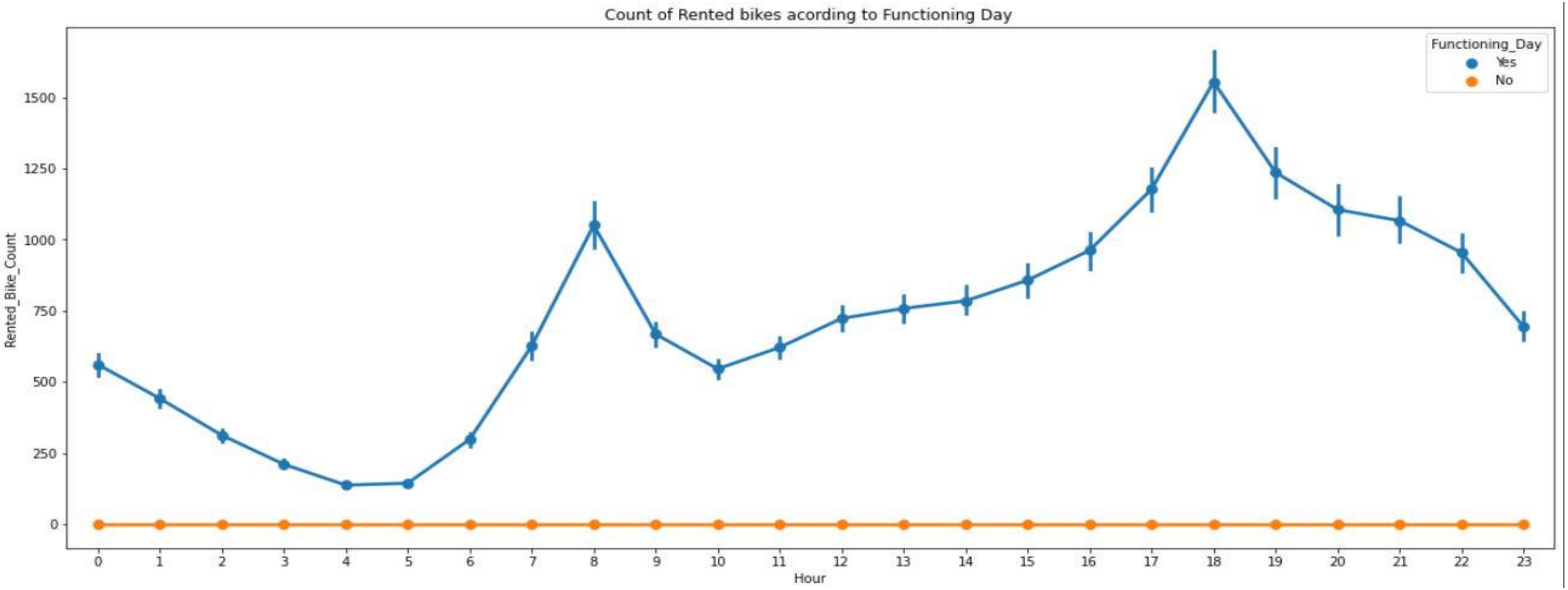
Observations:

1. **Here we observed that, Bike rental trend according to hours is almost similar in all scenarios.**

2. **There is sudden peak between 6/7AM to 10 AM. Office/College going time could be the reason for this sudden peak on Non-Holiday. But on Holiday the case is different, very less bike rentals happened.**

3. **Again there is peak between 4PM to 7 PM. May be its office leaving time for the above people.(Non-Holiday).**

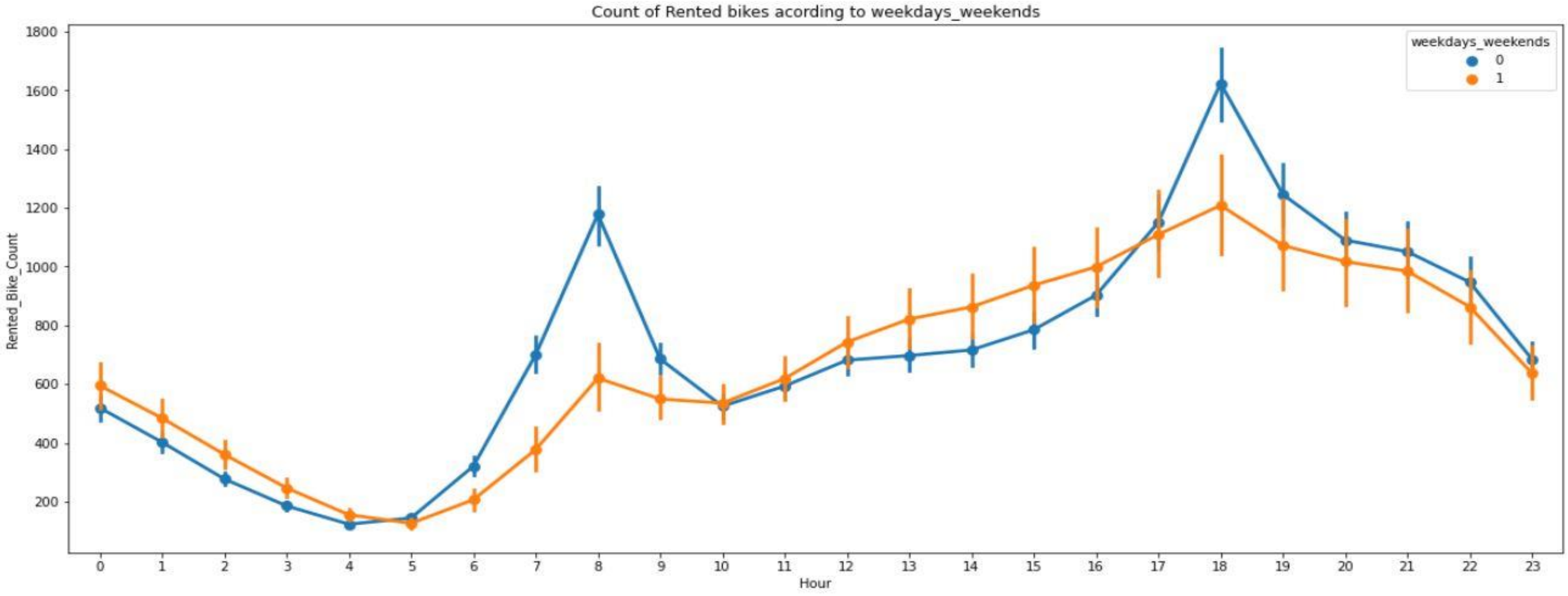
EDA (Exploratory Data Analysis):



Observation:

Here the trend for functioning day is same as of Non-holiday. Only the difference is on Non functioning day there were zero bike rentals.

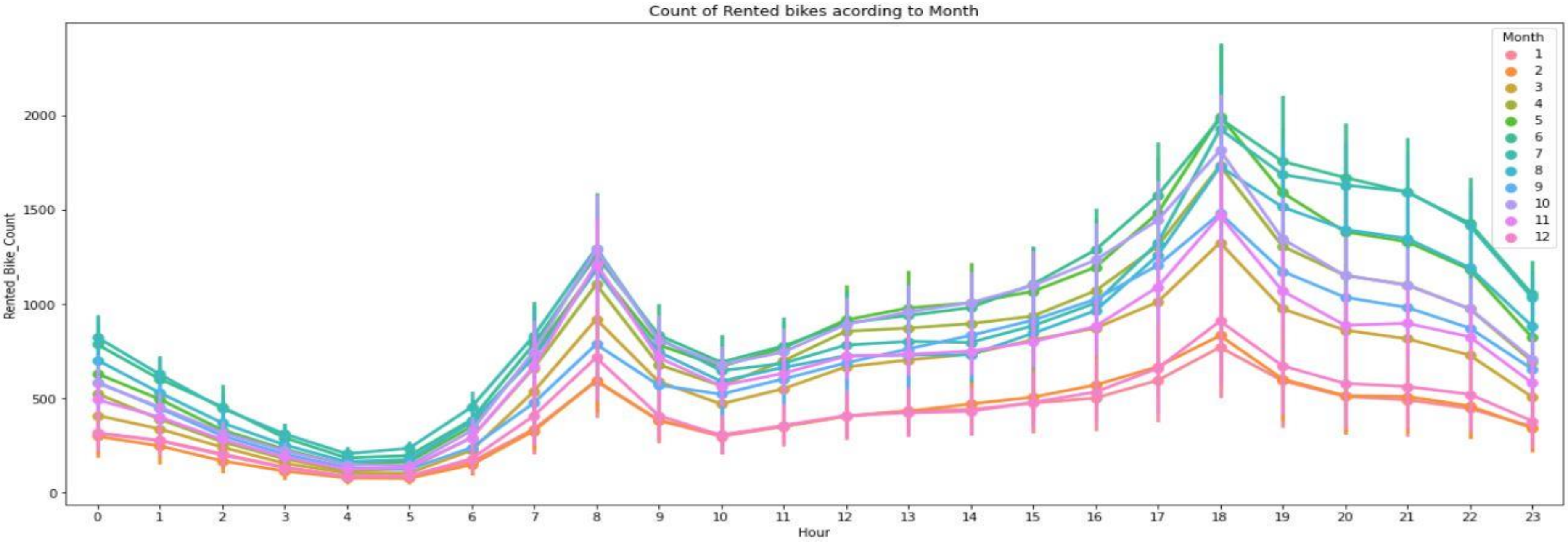
EDA (Exploratory Data Analysis):



Observation:

From the above plot we can say that on weekdays, which is represented in blue color, the demand for bikes is higher because of the office or college. Peak times are between 7 am to 9 am and 5 pm to 7 pm. The orange color represents the weekends, and it shows that the demand for rented bikes is very low, especially in the morning hours but during the evening, starting from 4 pm to 8 pm, the demand slightly increases.

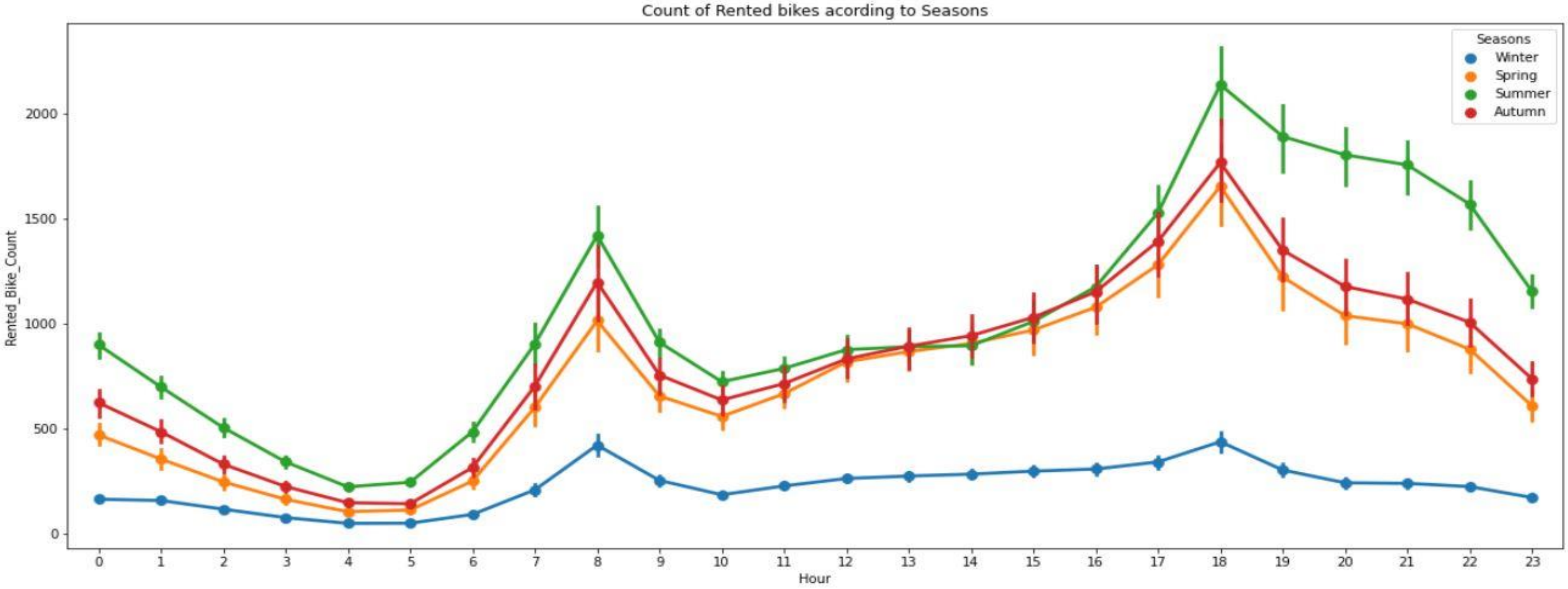
EDA (Exploratory Data Analysis):



Observation:

From the above bar and point plot, we can clearly say that from the month 5th to 10th the demand for the rented bike is high as compared to other months. These months constitute the summer season.

EDA (Exploratory Data Analysis):



Observation:

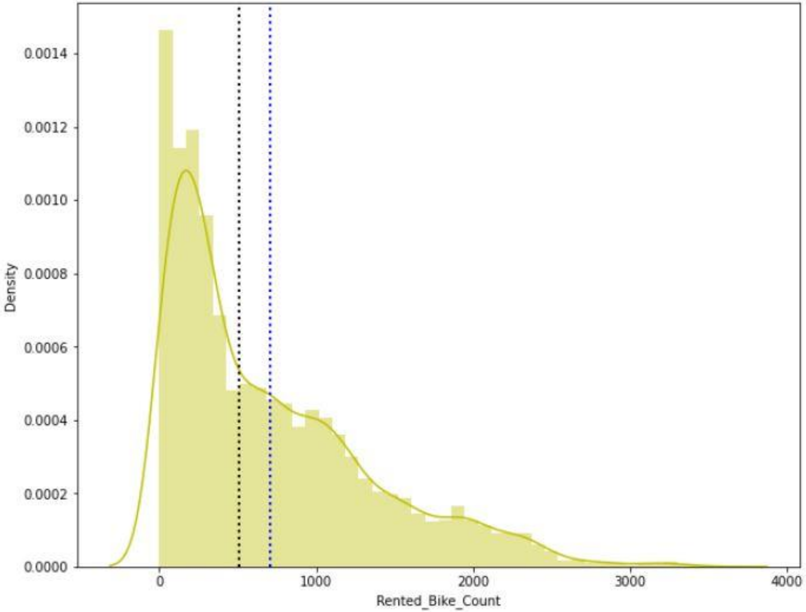
In the above bar plot and point plot which shows the use of rented bikes in four different seasons, and it clearly shows that :

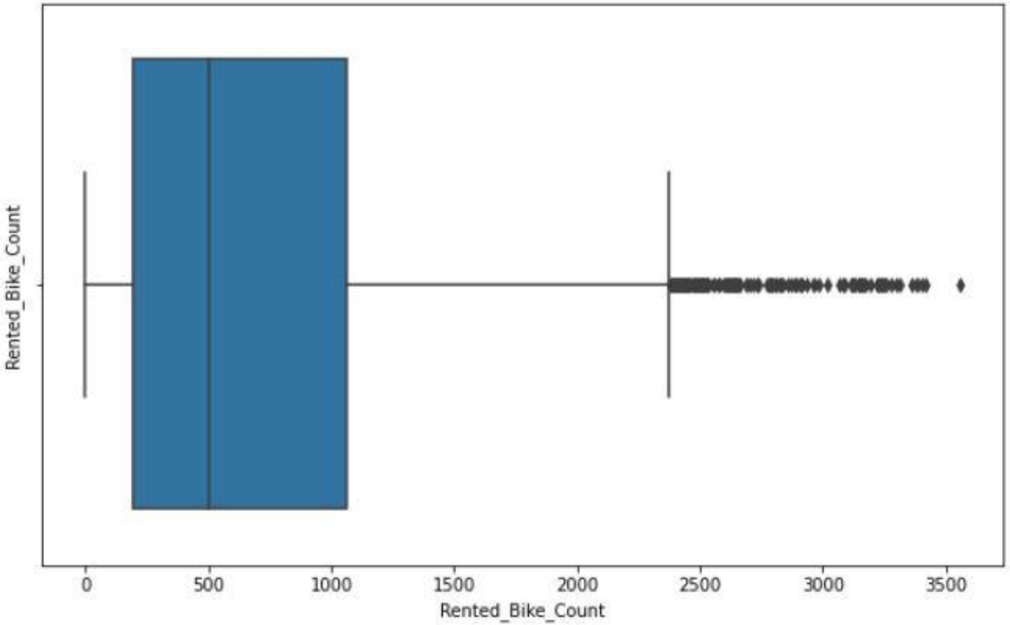
- **In Summer, Autumn and Spring seasons the use of rented bike is high and peak time is 7am-9am and 5pm-7pm.**
- **But In winter season the use of rented bike is very low because of snowfall.**

EDA (Exploratory Data Analysis):

AI

Distribution of target variable- Bike Rent Count

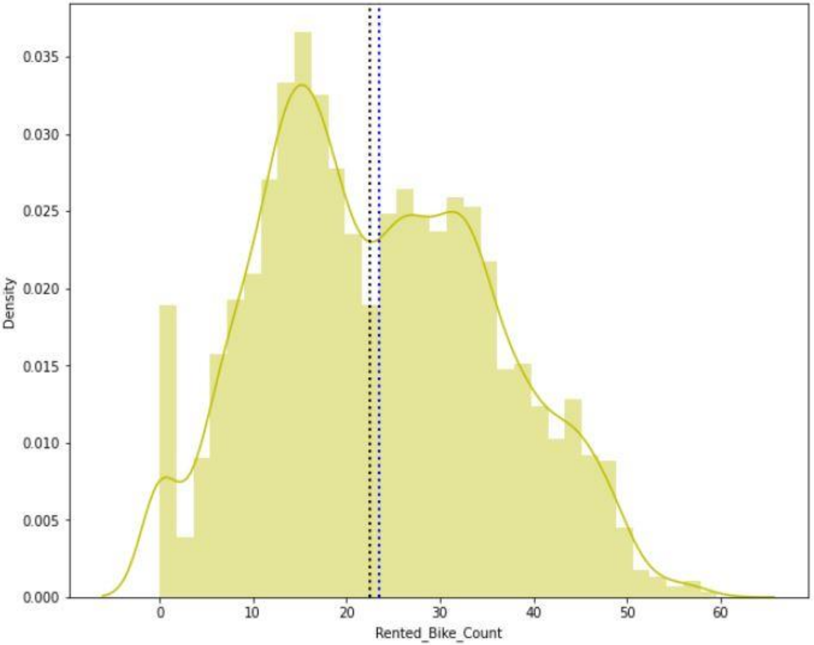


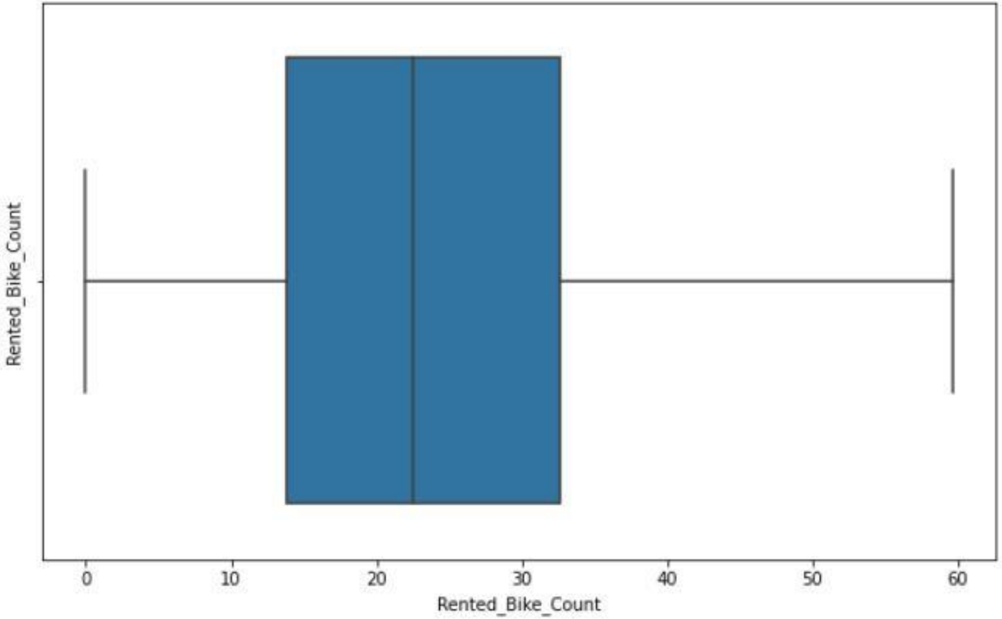


Observation:

The above graph shows that Rented Bike Count has moderate right skewness. Since the assumption of linear regression is that 'the distribution of dependent variable has to be normal', so we should perform some operation to make it normal.

EDA (Exploratory Data Analysis):



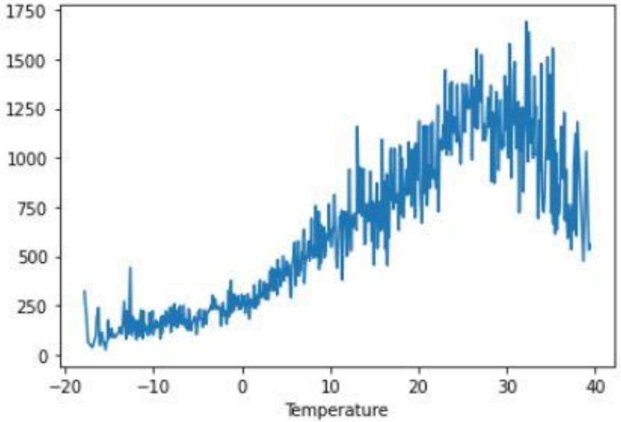


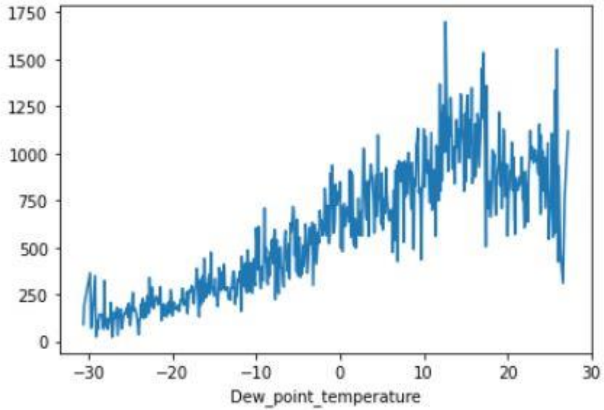
Observation:

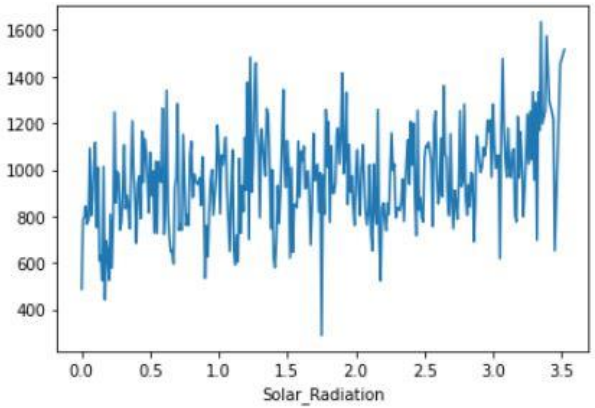
1. **We have the generic rule of applying Square root for the skewed variable in order to make it normal. After applying Square root to the skewed Rented Bike Count, here we get an almost normal distribution.**
2. **After applying Square root to the Rented Bike Count column, we find that there are no outliers present.**

EDA (Exploratory Data Analysis):

Numeric variables vs. Rented Bike count





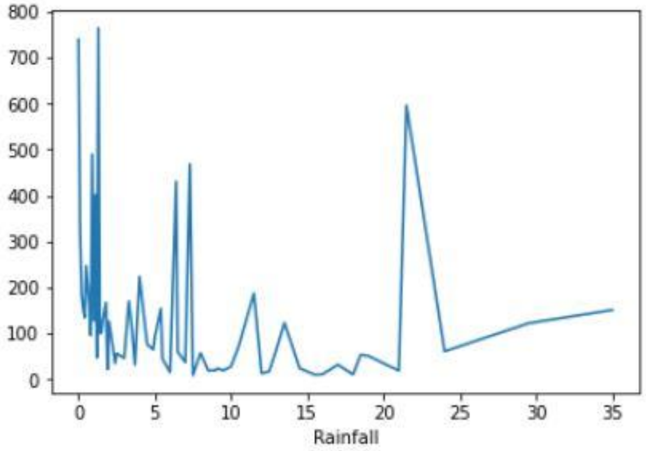


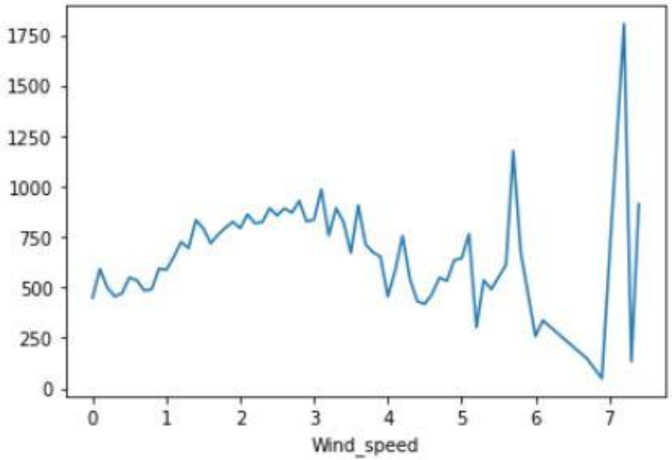
Observation:

- **From the above plot we see that people like to ride bikes when it is pretty hot around 25°C in average**
- **From the above plot of "Dew_point_temperature' is almost same as the 'temperature' there is some similarity present we can check it in our next step**
- **from the above plot we see that, the amount of rented bikes is huge, when there is solar radiation, the counter of rents is around 1000**

EDA (Exploratory Data Analysis):

Numeric variables vs. Rented Bike count

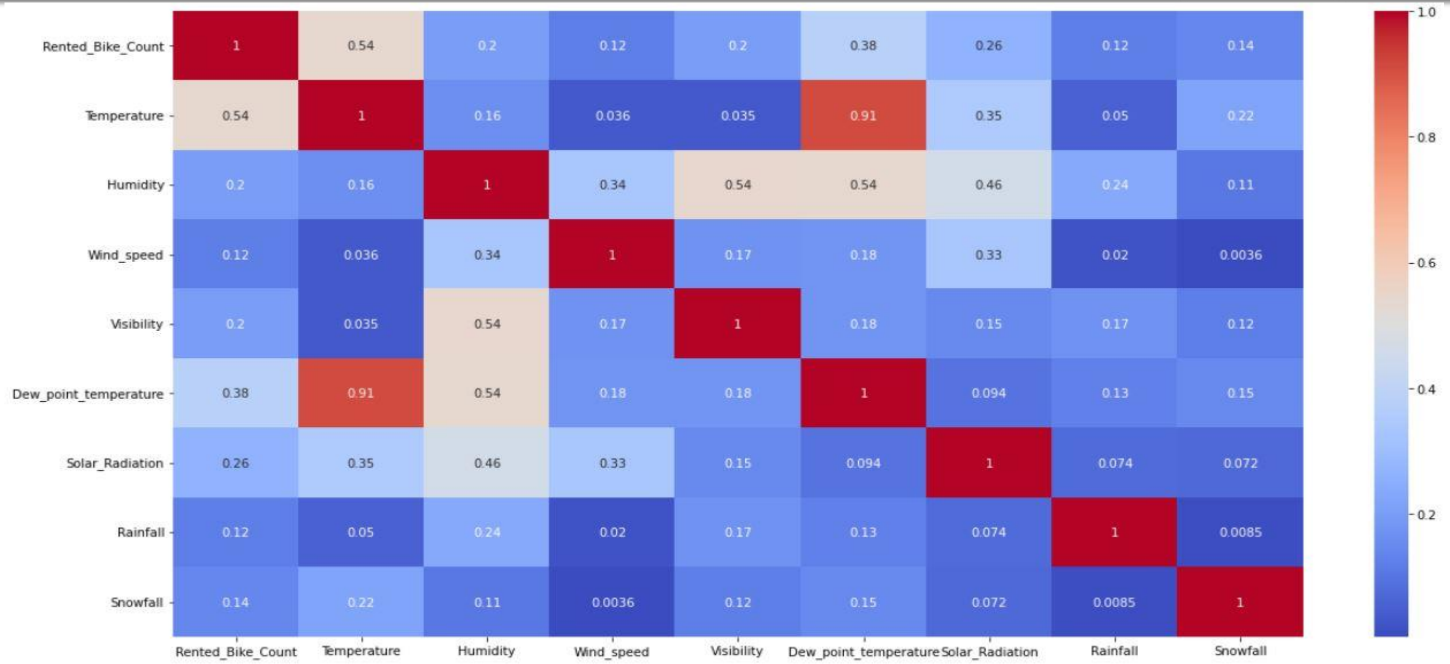




Observation:

- **From the rainfall plot, if it rains a lot the demand for rent bikes is not decreasing, here for example even if we have 20mm of rain there is a big peak of rented bikes.**
- **From the wind speed plot, we can see that the demand for rented bikes is uniformly distributed despite wind speed but when the speed of the wind was 7 m/s then the demand for bikes also increase which clearly means peoples prefer to ride bikes when it's windy**

EDA (Exploratory Data Analysis):



Observation:

- **Temperature and Dew point Temperature are highly correlated.**
- **As per our regression assumption, there should not be colinearity between independent variables.**
- **We can see from the heatmap that "Temperature" and "Dew Point Temperature" are highly correlated. We can drop one of them. As the correlation between temperature and our dependent variable "Bike Rented Count" is high. So we will Keep the Temperature column and drop the "Dew Point Temperature" column.**

Model selection and Evaluation

AI

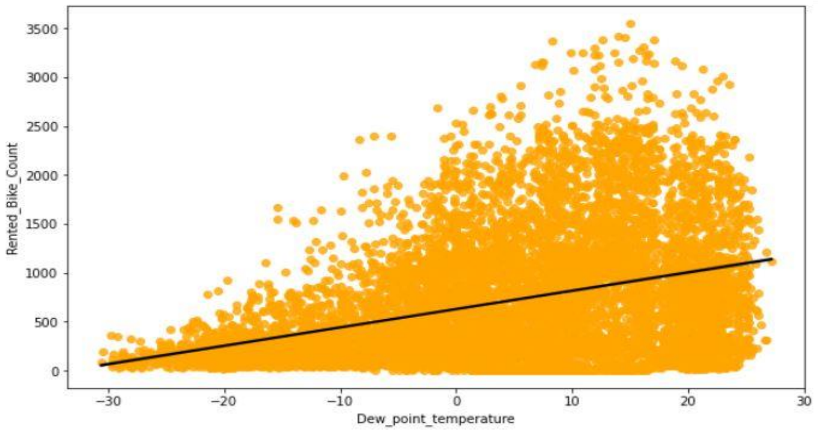
As this is the regression problem we are trying to predict continuous value. For this, we used the following regression models.

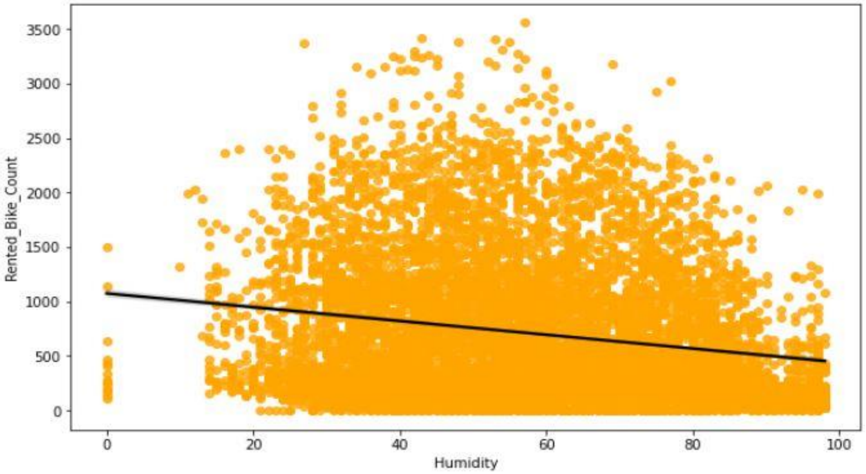
- Linear Regression
- Lasso regression (regularized regression)
- Ridge Regression(regularized regression)
- Decision Tree regression.
- Random forest regression
- Gradient Boosting regression.

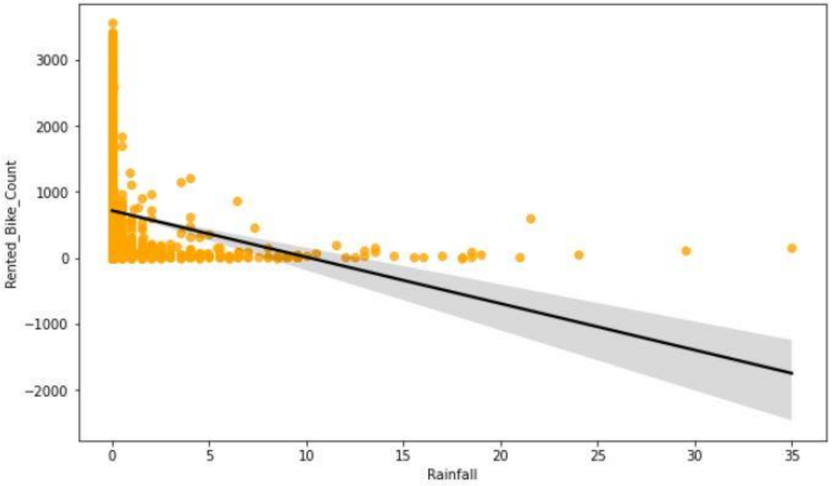
Assumptions of regression line:

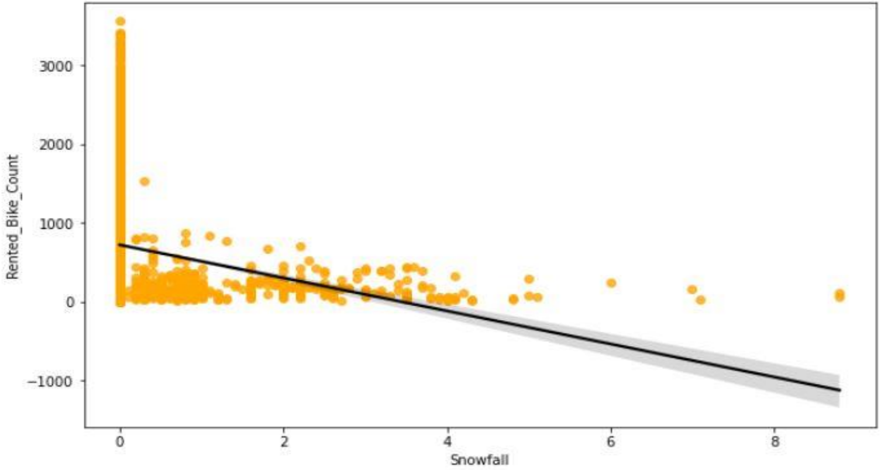
- The relation between the dependent and independent variables should be almost linear.
- The mean of residuals should be zero or close to 0 as much as possible. It is done to check whether our line is actually the line of “best fit”.
- There should be homoscedasticity or equal variance in a regression model. This assumption means that the variance around the regression line is the same for all values of the predictor variable (X).
- There should not be multicollinearity in the regression model. Multicollinearity generally occurs when there are high correlations between two or more independent variables.
- Before and after applying these models we checked our regression assumptions by the distribution of residuals, scatter plot of actual and predicted values, removing multi-colinearity among independent variables.

Regression plot for Numerical Variables

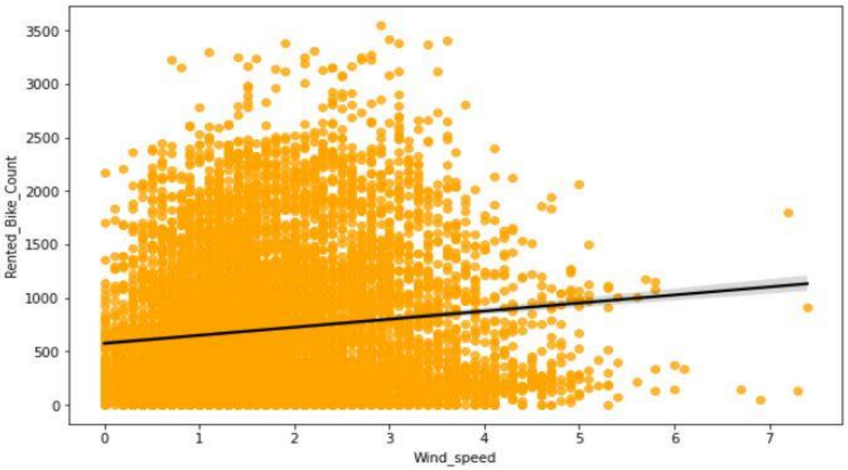


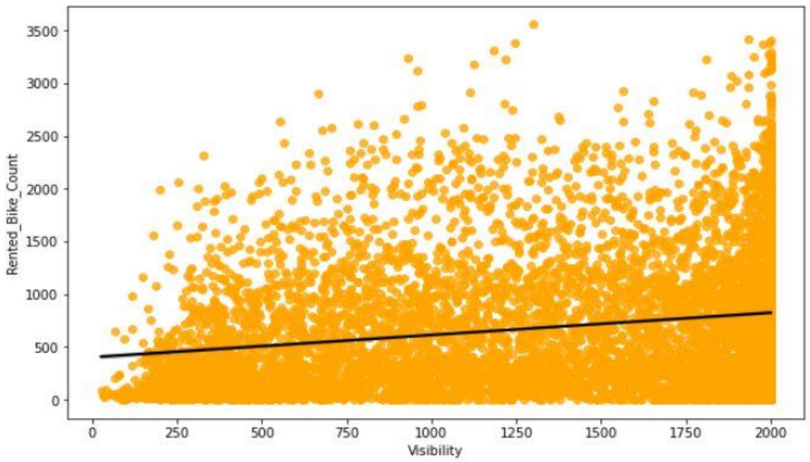


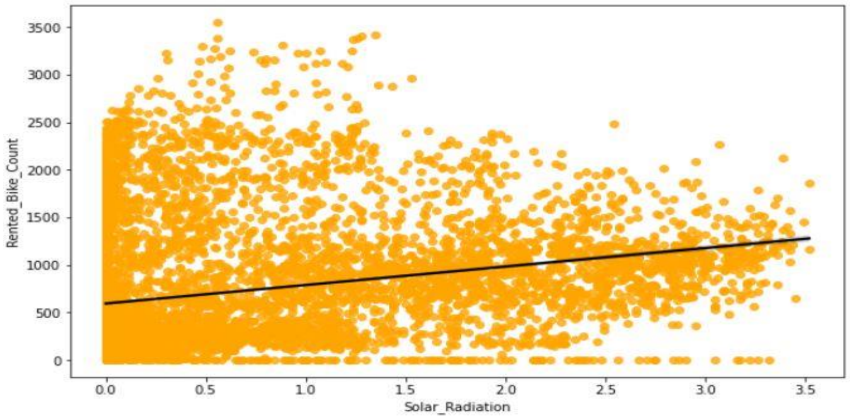


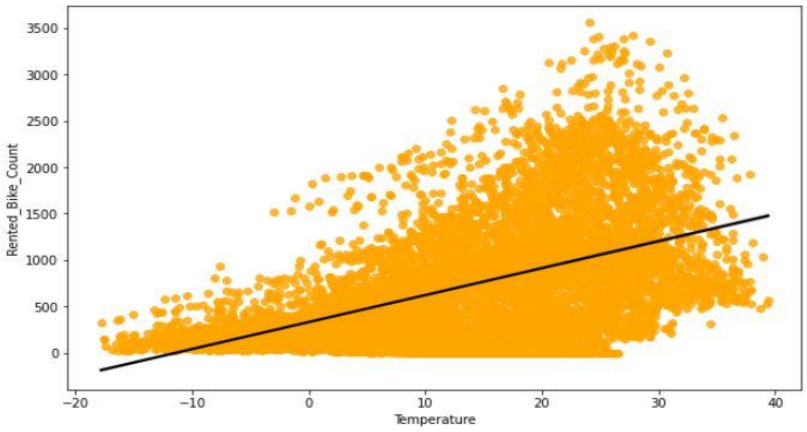


Regression plot for Numerical Variables









Regression plot for Numerical Variables

- **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 related to the target variable. Which means the rented bike count increases with an 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.**

Model selection and Evaluation

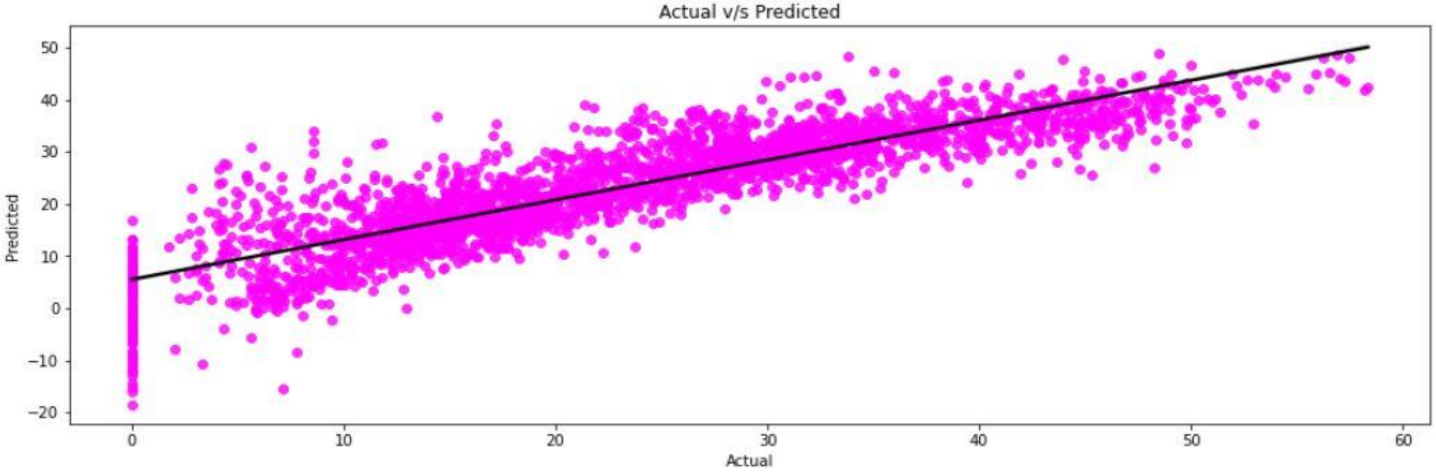
- **Linear Regression, Lasso, Ridge and Elasticnet Regression :**
  - **Linear Regression**

Score on train set :

```
Mean_Squared_Error_for_Linear_Regression :  38.09414735223389
Mean_Absolute_Error_for_Linear_Regression : 4.69650618949340З
Root_Mean_Square_Error_For_Linear_Regression : 6.1720456375689485
r2_score_for_LR : 0.754759122743357
Adjusted_r2_score_for_LR : 0.7502915563747282
```

Score on test set :

```
Mean_Squared_Error_for_Linear_Regression :  35.67429301457273
Mean_Absolute_Error_for_Linear_Regression : 4.557275256369084
Root_Mean_Square_Error_For_Linear_Regression : 5.972796080109611
r2_score_for_LR : 0.7686482214946567
Adjusted_r2_score_for_LR : 0.7668609871768146
```



Model selection and Evaluation

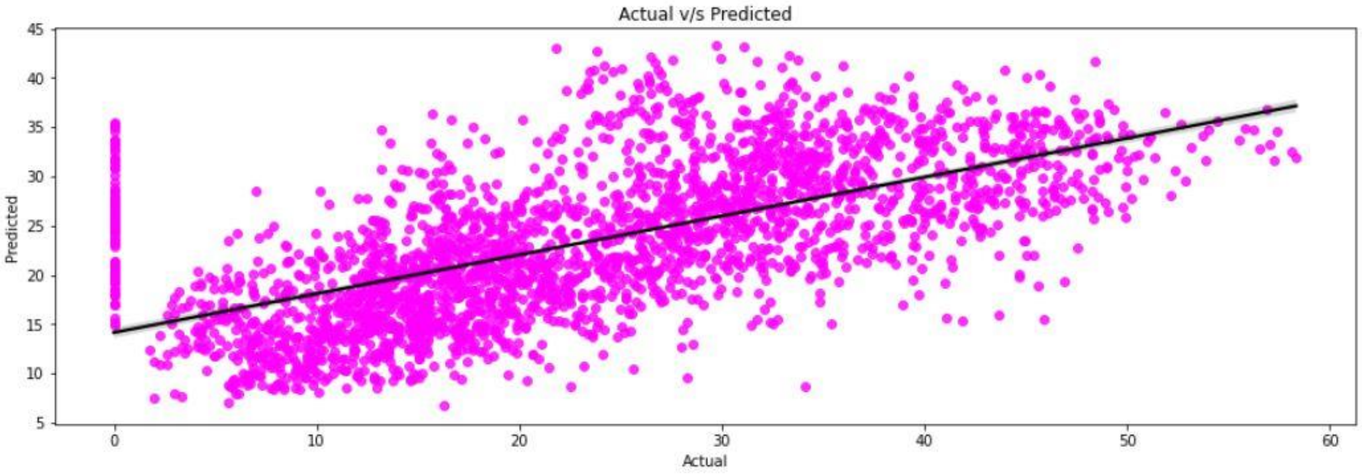
- **Linear Regression, Lasso, Ridge and Elasticnet Regression :**
  - **Lasso Regression**

Score on train set :

```
Mean_Squared_Error_for_Lasso_Regression :  93.4985716869806
Mean_Absolute_Error_for_Lasso_Regression : 7.352671897027052
Root_Mean_Square_Error_For_Lasso_Regression : 9.669465946316818
r2_score_for_lasso : 0.39807888254483725
Adjusted_r2_score_for_lasso : 0.3871136528857704
```

Score on test set :

```
Mean_Squared_Error_for_Lasso_Regression :  92.03178332737271
Mean_Absolute_Error_for_Lasso_Regression : 7.23521308666217
Root_Mean_Square_Error_For_Lasso_Regression : 9.593319724025292
r2_score_for_lasso : 0.4031635961753036
Adjusted_r2_score_for_lasso : 0.3922909950203577
```



Model selection and Evaluation

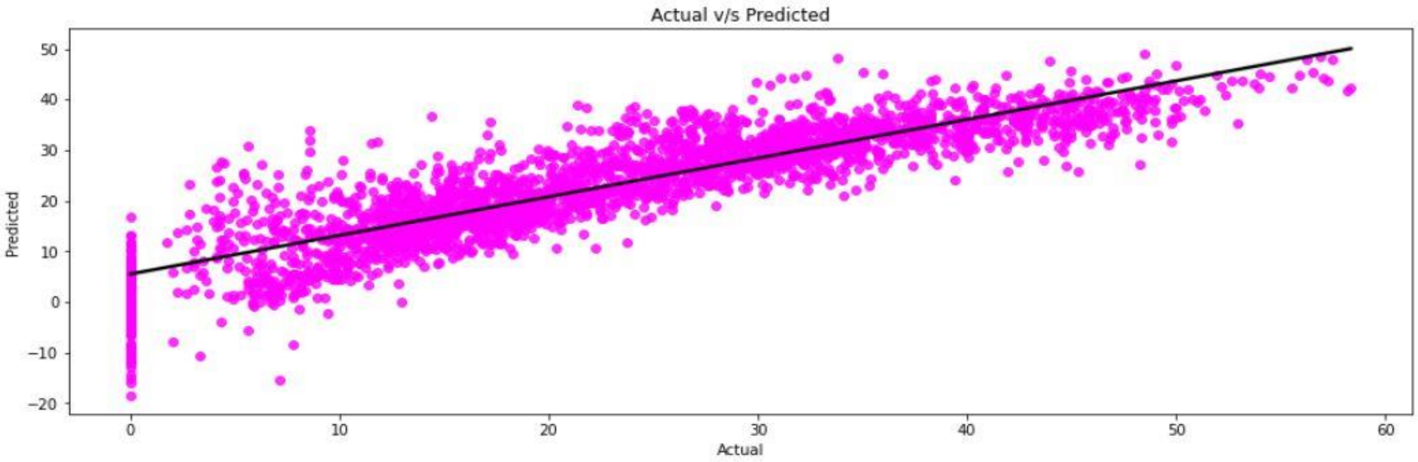
- **Linear Regression, Lasso, Ridge and Elasticnet Regression :**
  - **Ridge Regression**

Score on train set :

```
Mean_Squared_Error_for_Ridge_Regression :  38.09416275499652
Mean_Absolute_Error_for_Ridge_Regression : 4.696611837350968
Root_Mean_Square_Error_For_Ridge_Regression : 6.172046885353069
r2_score_for_ridge : 0.7547590235841086
Adjusted_r2_score_for_ridge : 0.7502914554090905
```

Score on test set :

```
Mean_Squared_Error_for_Ridge_Regression :  35.67380170112001
Mean_Absolute_Error_for_Ridge_Regression : 4.557363318023901
Root_Mean_Square_Error_For_Ridge_Regression : 5.9727549507007245
r2_score_for_ridge : 0.7686514077173254
Adjusted_r2_score_for_ridge : 0.7644369178579125
```



Model selection and Evaluation

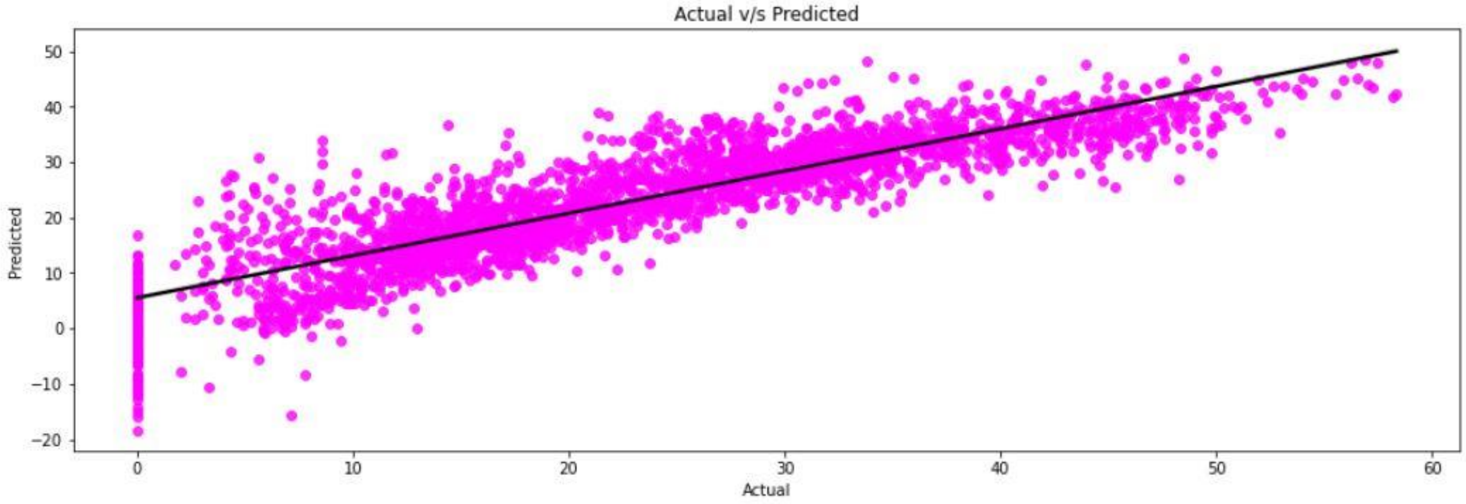
- Linear Regression, Lasso, Ridge and Elasticnet Regression :
  - Elastic Net Regression

Score on train set :

```
Mean_Squared_Error_for_elasticnet_Regression :  38.09432889810408
Mean_Absolute_Error_for_elasticnet_Regression : 4.696795768785435
Root_Mean_Square_Error_For_elasticnet_Regression : 6.172060344658345
r2_score_for_elastic : 0.754757953995096
Adjusted_r2_score_for_elastic : 0.7502903663353169
```

Score on test set :

```
Mean_Squared_Error_for_elasticnet_Regression :  35.67240922607158
Mean_Absolute_Error_for_elasticnet_Regression : 4.557501249456823
Root_Mean_Square_Error_For_elasticnet_Regression : 5.972638380654866
r2_score_for_elasticnet : 0.76866043807369
Adjusted_r2_score_for_elasticnet : 0.7644461127207689
```



Model selection and Evaluation

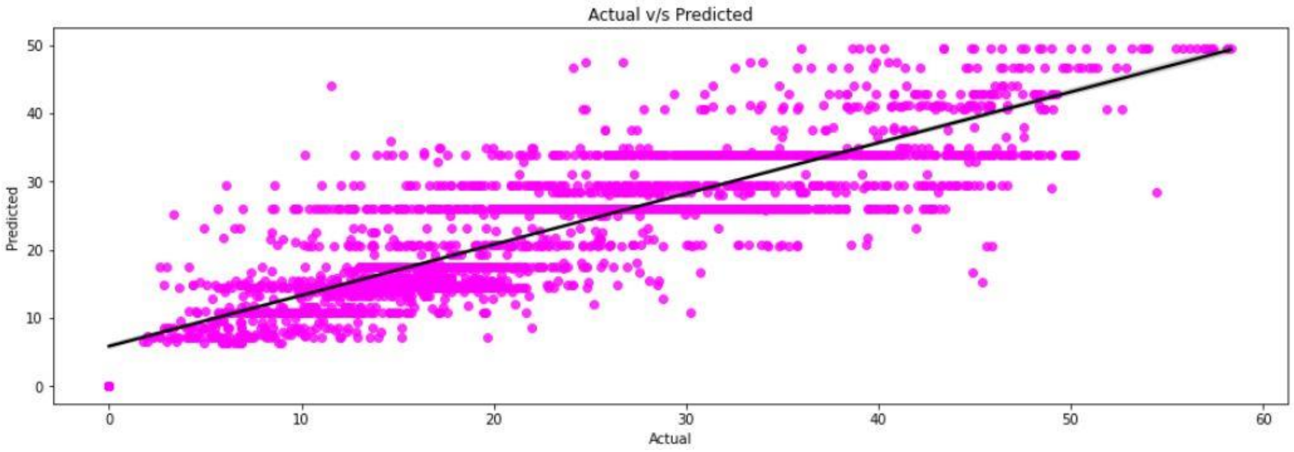
- **Decision Tree Regression :**
  (criterion='mse', max_depth=9, max_leaf_nodes=100, max_features=9, random_state = 2)

Score on train set :

```
Model Score: 0.7640894376677909
Mean_Squared_Error_for_Decision_tree_Regression :  36.64483598314194
Mean_Absolute_Error_for_Decision_tree_Regression : 4.409605810490887
Root_Mean_Square_Error_For_Decision_tree_Regression : 6.053497830440013
r2_score_for_Decision_tree : 0.7640894376677909
Adjusted_r2_score_for_Decision_tree : 0.7597918421524368
```

Score on test set :

```
Model Score: 0.73847377587056
Mean_Squared_Error_for_Decision_tree_Regression :  40.32717280525675
Mean_Absolute_Error_for_Decision_tree_Regression : 4.586685705041654
Root_Mean_Square_Error_For_Decision_tree_Regression : 6.350367926762728
r2_score_for_Decision_tree : 0.73847377587056
Adjusted_r2_score_for_Decision_tree : 0.7337095384542485
```



Model selection and Evaluation

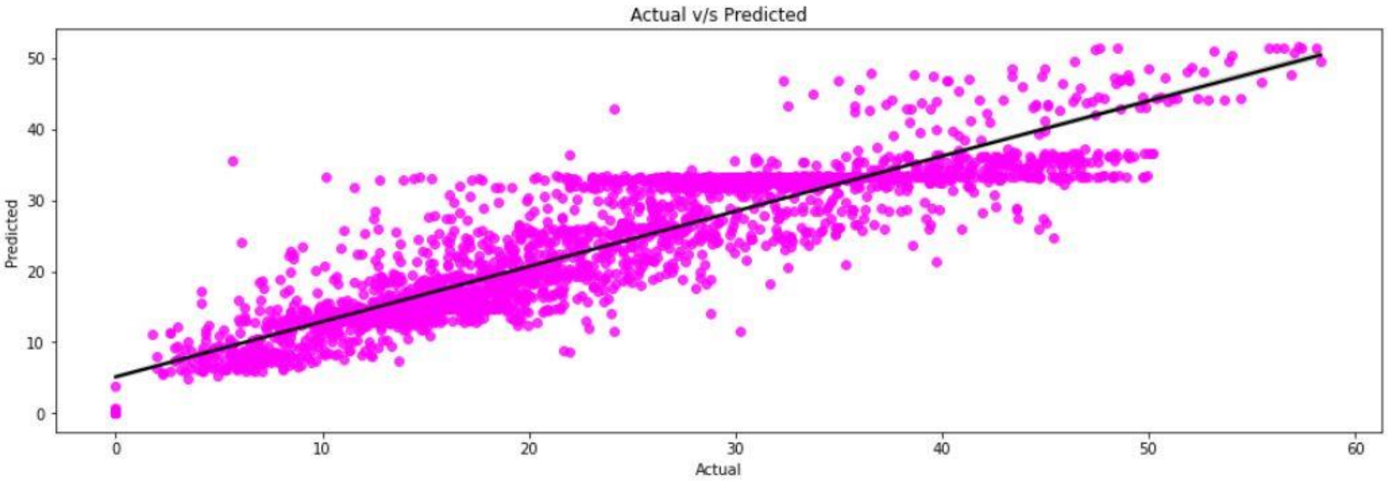
- **Random Forest Regression (HyperParameter Tuned – ‘max_depth=9’, ‘n_estimator=100’)**

Score on train set :

```
Model Score: 0.8450693626166363
Mean_Squared_Error_for_Random_Forest__Regression :  24.065933036444896
Mean_Absolute_Error_for_Random_Forest_Regression : 3.5671660988383254
Root_Mean_Square_Error_For_Random_Forest_Regression : 4.905704132583303
r2_score_for_Random_Forest : 0.8450693626166363
Adjusted_r2_score_for_Random_Forest : 0.8422469827883348
```

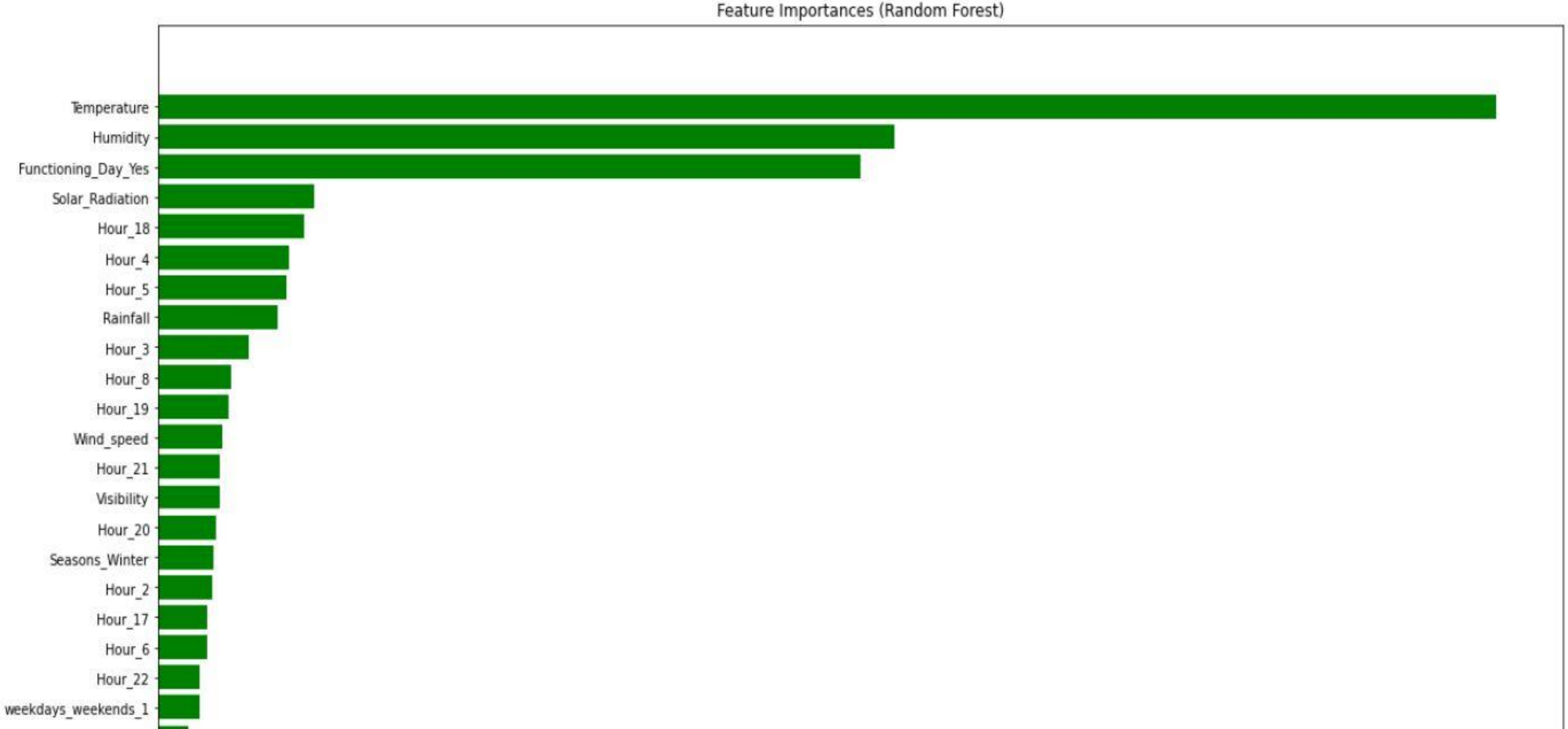
Score on test set :

```
Model Score: 0.818171637317473
Mean_Squared_Error_for_Random_Forest_Regression :  28.03781466735799
Mean_Absolute_Error_for_Random_Forest_Regression : 3.837883144451421
Root_Mean_Square_Error_For_Random_Forest_Regression : 5.2950745667420005
r2_score_for_Random_Forest : 0.818171637317473
Adjusted_r2_score_for_Random_Forest : 0.5679109722455142
```



Model selection and Evaluation

Feature Importance (Random Forest Regression):



Model selection and Evaluation

- **Gradient Boosting Regression:**

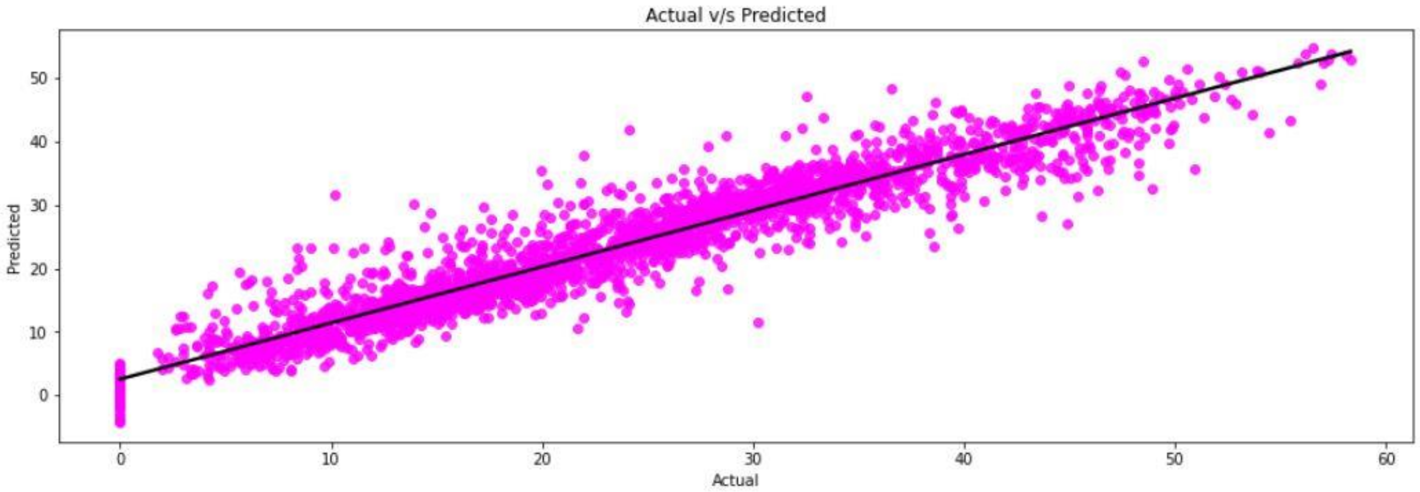
GridSearchCV(estimator=gb_reg, param_grid= param_dict, cv=5, verbose=2)

Score on train set :

```
Model Score: 0.9315330030230752
Mean_Squared_Error_for_Gradient_Boosting_optimal :  10.63522484823959
Mean_Absolute_Error_for_Gradient_Boosting_optimal : 2.2708225048436144
Root_Mean_Square_Error_For_Gradient_Boosting_optimal : 3.261169245568158
r2_score_for_Gradient_Boosting_optimal : 0.9315330030230752
Adjusted_r2_score_for_Gradient_Boosting_optimal : 0.9302857360238832
```

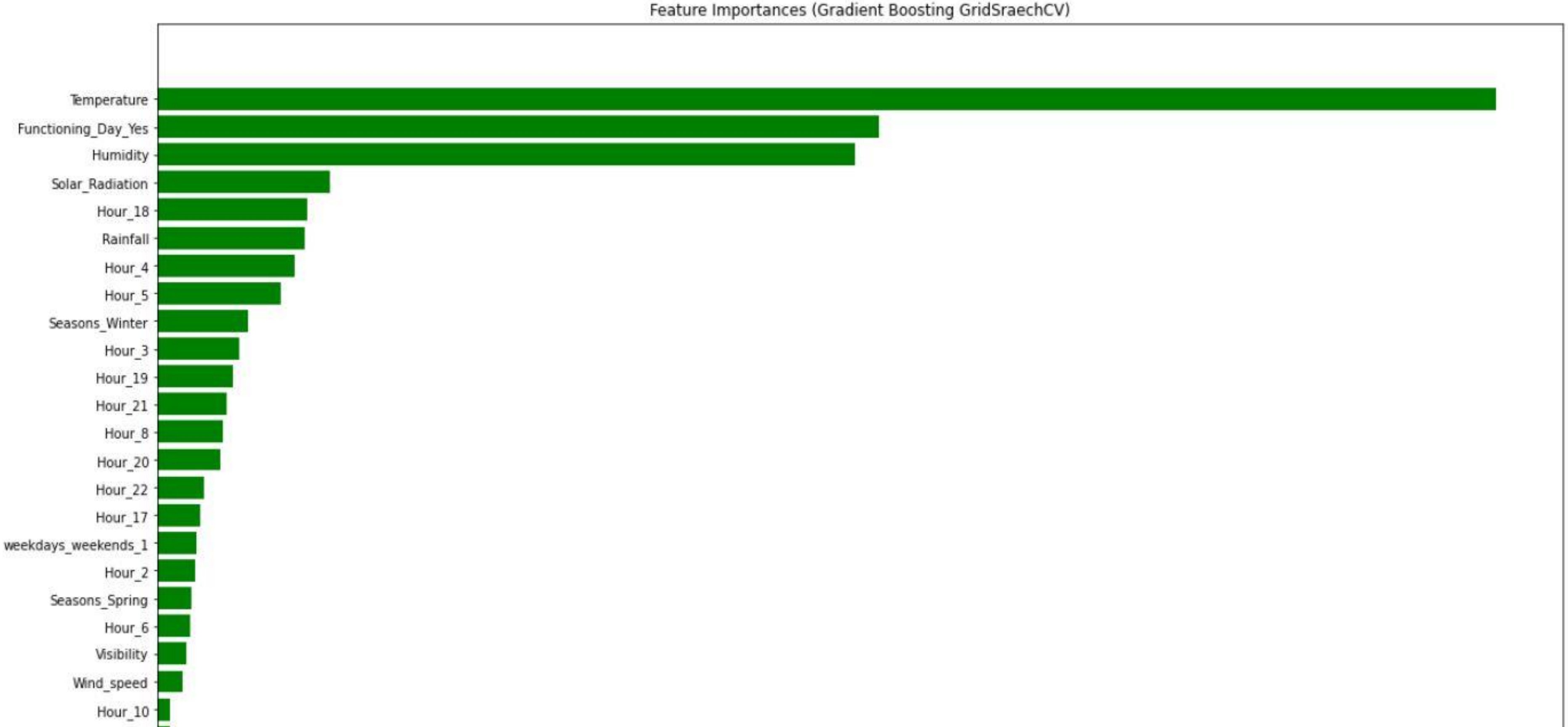
Score on test set :

```
Model Score: 0.9128367342078767
Mean_Squared_Error_for_Gradient_Boosting_optimal :  13.44051860791498
Mean_Absolute_Error_for_Gradient_Boosting_optimal : 2.577457568399969
Root_Mean_Square_Error_For_Gradient_Boosting_optimal : 3.6661312862355295
r2_score_for_Gradient_Boosting_optimal : 0.9128367342078767
Adjusted_r2_score_for_Gradient_Boosting_optimal : 0.9112488762651519
```



Model selection and Evaluation

Feature Importance (Gradient Boosting - GridSearchCV):



Conclusion:

		Model	MAE	MSE	RMSE	R2_score	Adjusted_R2
Training set	0	Linear regression	4.697	38.094	6.172	0.755	0.750
	1	Lasso regression	7.353	93.499	9.669	0.398	0.390
	2	Ridge regression	4.697	38.094	6.172	0.755	0.750
	3	Elastic net Regression	4.697	4.697	6.172	0.755	0.750
	4	Decision Tree Regression	5.431	55.060	7.420	0.646	0.639
	5	Random Forest Regression	3.593	24.385	4.938	0.843	0.840
	6	Gradient Boosting	3.496	21.453	4.632	0.862	0.859
	7	Gradient Boosting GridSearchCV	2.271	10.635	3.261	0.932	0.930
Test set	0	Linear regression	4.697	35.674	6.172	0.769	0.770
	1	Lasso regression	7.235	92.032	9.593	0.403	0.390
	2	Ridge regression	4.557	35.674	5.973	0.769	0.760
	3	Elastic net Regression	4.557	35.672	5.973	0.769	0.764
	4	Decision Tree Regression	5.562	57.569	7.587	0.627	0.620
	5	Random Forest Regression	3.863	28.327	5.322	0.816	0.813
	6	Gradient Boosting	3.579	23.015	4.797	0.851	0.848
	7	Gradient Boosting GridSearchCV	2.577	13.441	3.666	0.913	0.911

As we have calculated MAE, MSE, RMSE and R2 score for each model. Based on r2 score will decide our model performance.
Our assumption: if the difference of R2 score between Train data and Test is more than 5 % we will consider it as over fitting.

Linear, Lasso, Ridge and Elastic Net:
Linear, Ridge and Elastic regression models have almost similar R2 scores 75% on training and 76% test data.But Lasso Regression is not performing well.

Decision Tree Regression:
On Decision tree regression model, we got r2 score as 76% on training data and 73% on test data. Thus our model memorized the data. So it was not a over fitted model, which is quite good for us.

Random Forest:
On Random Forest regression model, we got r2 score as 84% on training data and 81% on test data. Thus our model memorized the data. So it was not a over fitted model, which is quite good for us.

Gradient Boosting Regression(Gradient Boosting Machine): On Gradient Boosting Regression model, without hyper-parameter tuning we got r2 score as 86% on training data and 85% on test data. Our model performed well without hyper-parameter tuning. After hyper-parameter tuning we got r2 score as 93% on training data and 91% on test data, thus we improved the model performance by hyper-parameter tuning.

Conclusion:

AI

Thus Gradient Boosting Regression GridSearchCV(estimator= gb_reg, param_grid= param_dict, cv=5, verbose=2)

and

Random forest Regression (HyperParameter Tuned – ‘max_depth=9’, ‘n_estimator=100’) gives good r2 scores. We can deploy these models.

However, this is not the end. As this data is time-dependent, the values for variables like temperature, wind speed, 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 quality knowledge and keeping pace with the ever-evolving ML field would surely help one to stay a step ahead in the future.

Signing off…

THANK YOU