Capstone Project – 2

Seoul Bike Sharing Demand Prediction

By

SHAIK AHMAD BASHA

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.

Our main objective behind this project is to explore and analyze the data to discover the key understandings. And to predict the count of bikes required at each hour by using regression models.

Work Flow:

1. Data Understanding

2. Data Cleaning And Manipulation

3. Exploratory Data Analysis

4. Feature Engineering

5. Pre Processing

6. Model Implementation

1. Model Explainability

Data Understanding :

The dataset has 8760 rows and 14 columns. It contains weather information like temperature, humidity etc. , number of bikes rented per hour and date information.

The features of the dataset are :

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/m2

Rainfall – mm

Snowfall – cm

Seasons - Winter, Spring, Summer, Autumn

Holiday - Holiday/No holiday

Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours)

Data Cleaning and Manipulation :

In Data cleaning and manipulation, we will check for null values, duplicated values and manipulate the data for our need.

Hence, the data has zero null values and zero duplicated values.

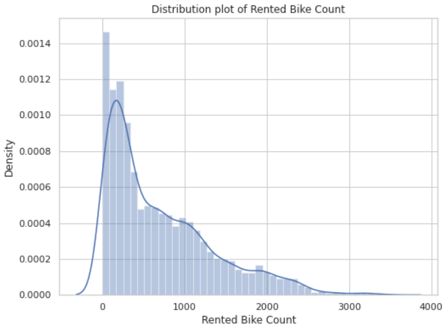
In the dataset,

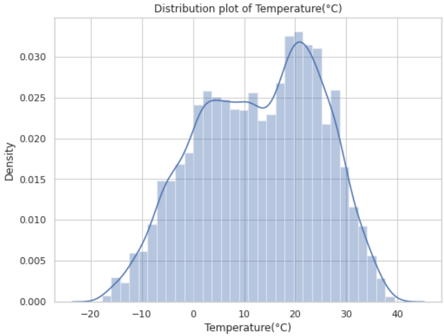
- ‘Date’ column have string datatype values, so we will convert them into datetime feature.
- ‘Hour’ column have numerical values, but we will convert them into categorical because we will not perform any mathematical operations on them.
- I created a new column ‘Day’ which contains day name based on the date.
- I also create a new column ‘weekend’ which contains 0(is not weekend) and 1(is weekend)

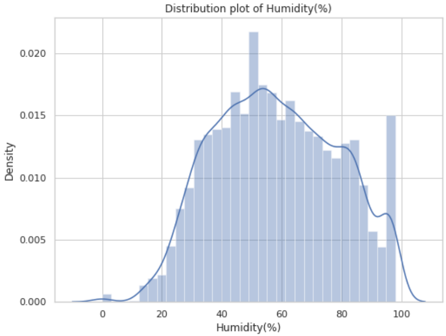
Exploratory Data Analysis :

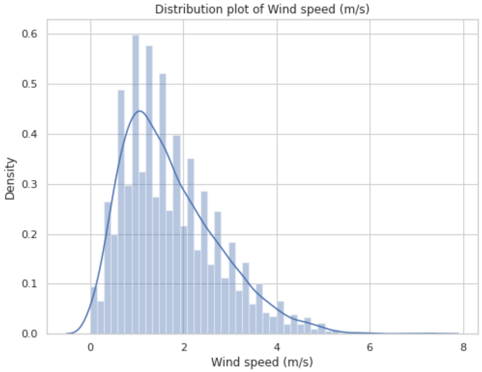
Numerical Feature	Categorical Feature	Datetime Feature
Rented Bike Count (Dependent feature)	Hour	Date
Temperature(°C)	Seasons	
Humidity(%)	Holiday	
Wind speed (m/s)	Functioning Day	
Visibility (10m)	Day	
Dew point temperature(°C)		
Solar Radiation (MJ/m2)		
Rainfall(mm)		
Snowfall (cm)		

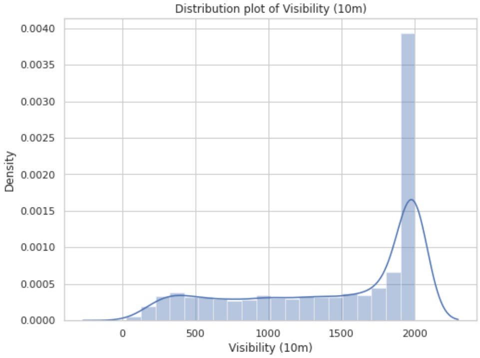
Distribution of numerical features :

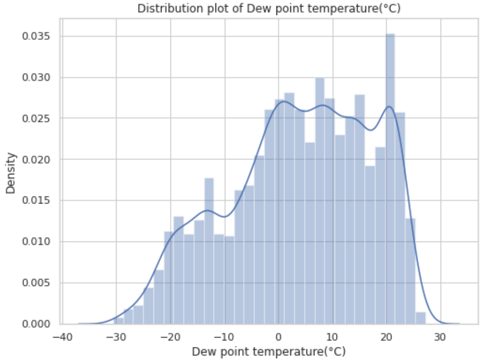


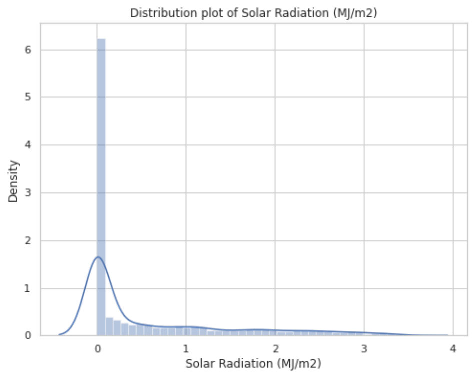


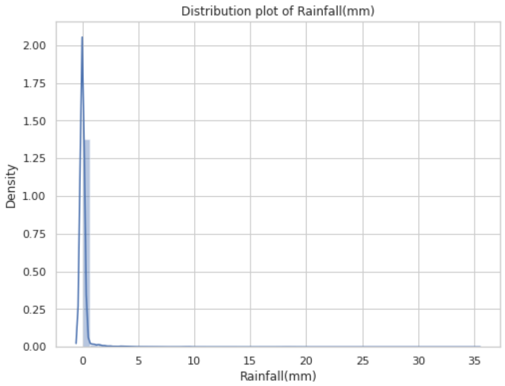


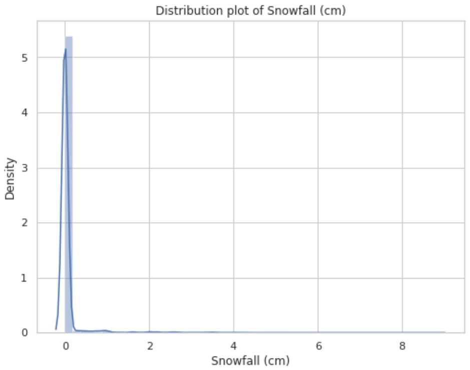








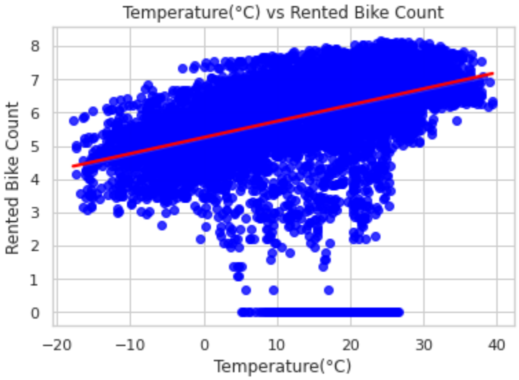


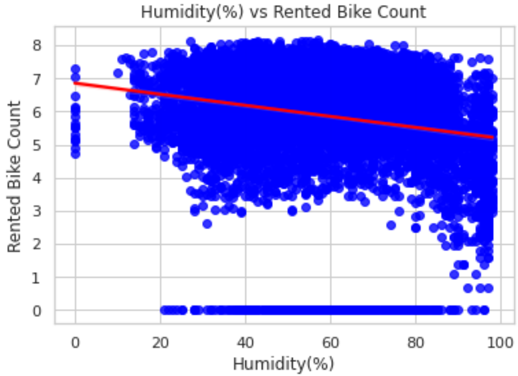


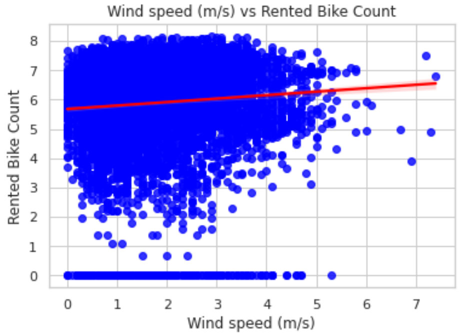
From the above distribution plots, we can observe that

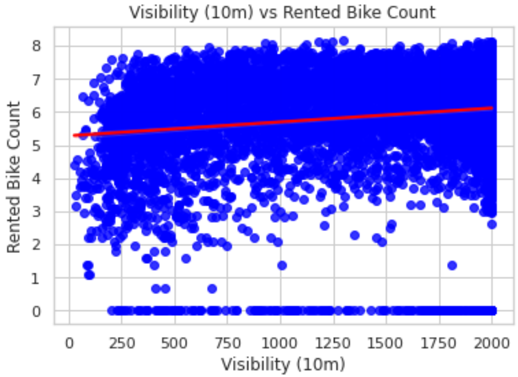
- Most number of the bike count ranges from 0 to 500.
- Temperature mostly varies from 20 to 30 .
- Humidity mostly varies from 20 to 100 .
- Wind speed mostly varies from 2 to 4 m/s .
- Visibility of 2000 count is high.
- Solar Radiation is mostly 0. And a few are in range of 1 to 4 .
- Mostly there is no rainfall and snowfall. And a very few have rainfall and snowfall.

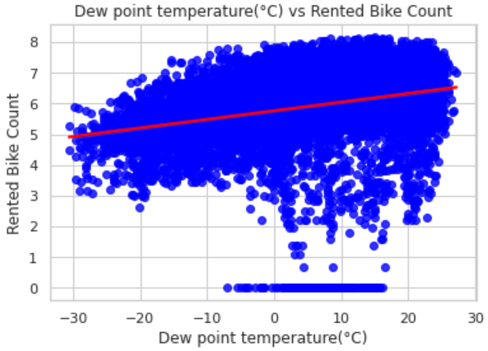
Relationship between numerical features and dependent feature (Scatter plots with regression line):

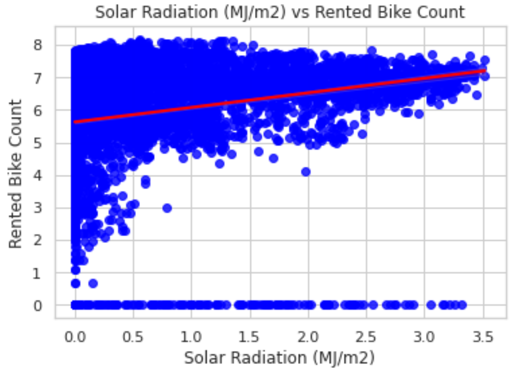


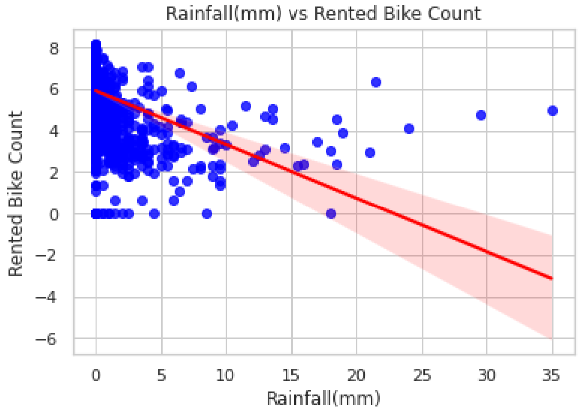


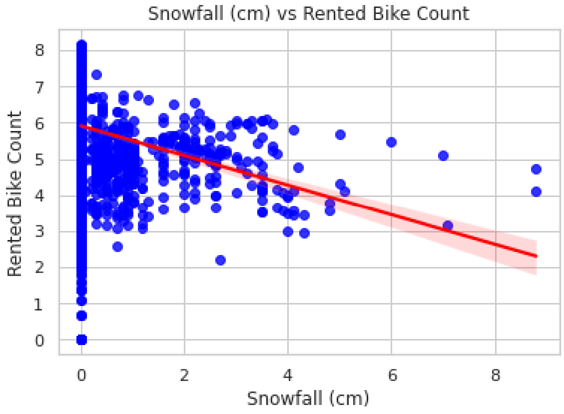








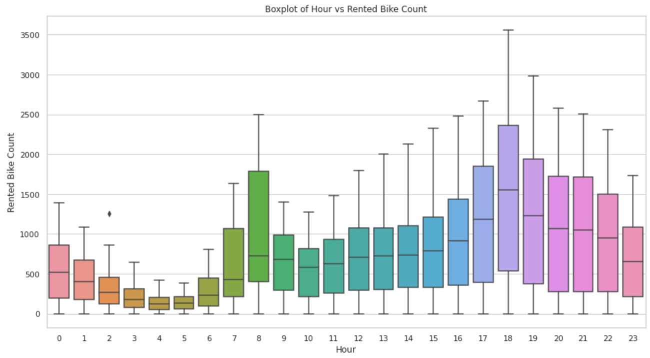


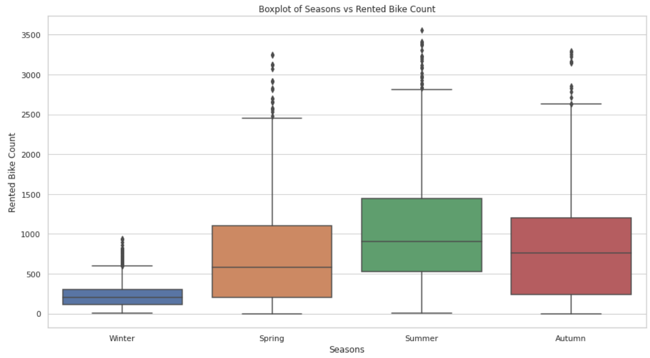


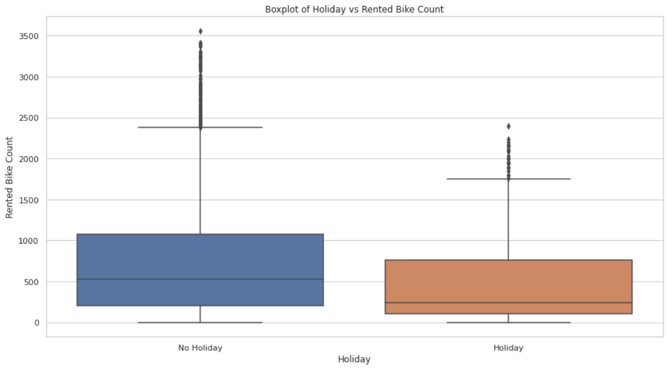
From the above scatter plots, we can observe that

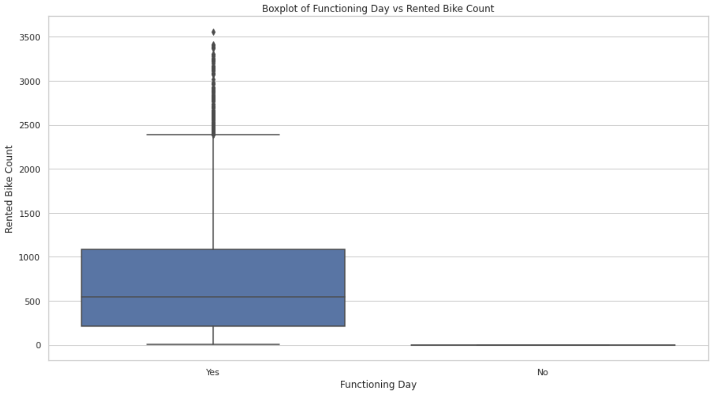
- **In every feature, there are outliers.**
- **The distribution between numerical features and dependent feature (Rented Bike Count) has been spread out entire area which means there is no specific relation between them..**

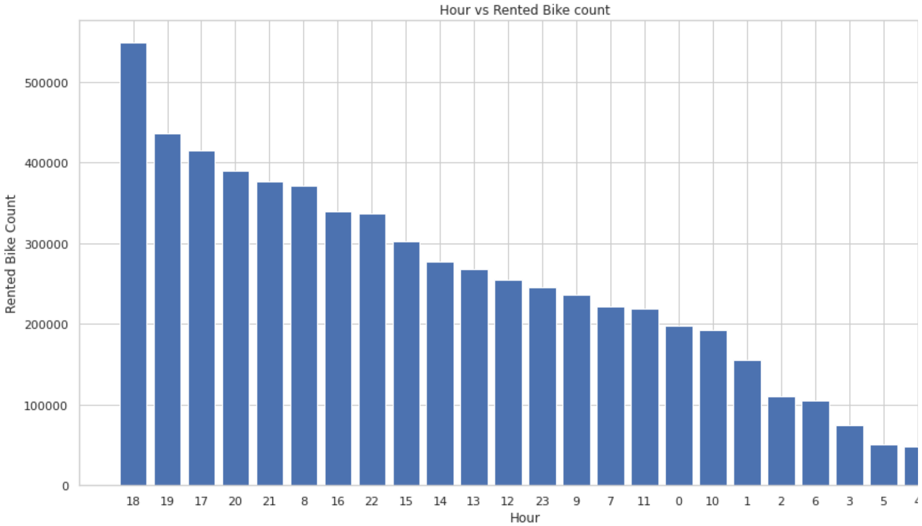
Relationship between Categorical features and dependent feature :

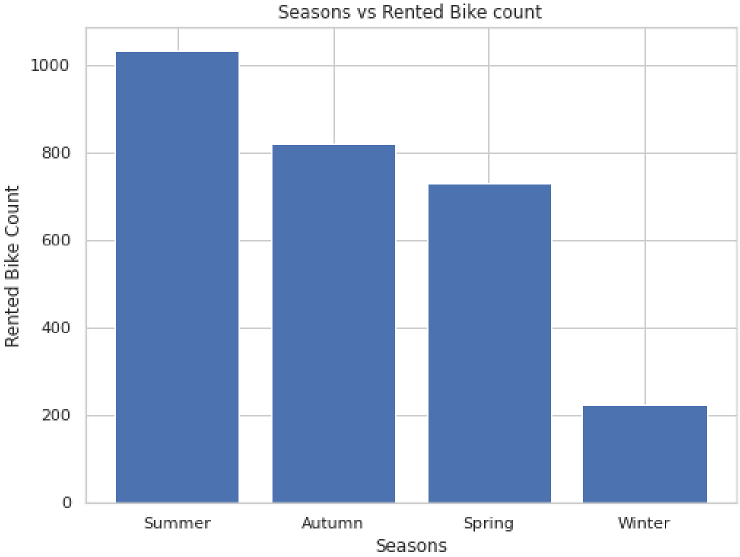






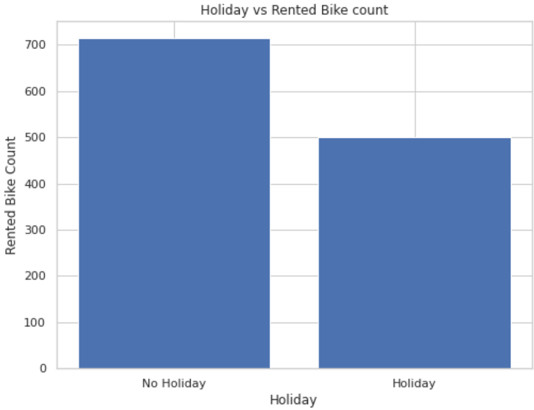


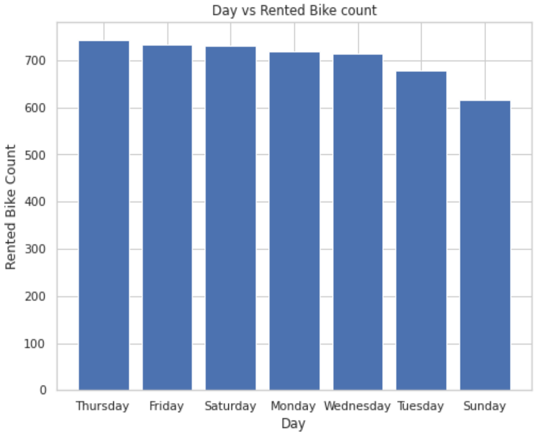


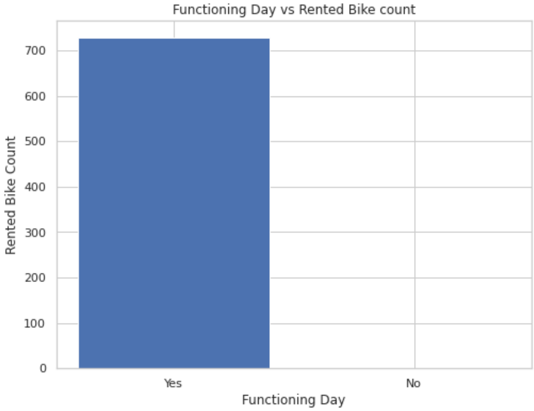


From the above visualizations plots, we can observe that

- **More number of bikes are rented in the hour of 18 followed by 19th hour. And the least is 4th hour.**
- **In summer season, most number of bikes are rented and the least is winter season.**







From the above visualizations plots, we can observe that

- ❖ **In working days(No Holiday), most number of bikes are rented.**
- ❖ **Thursday has high count of rented bike and the least is Sunday.**
- ❖ **In a functioning day, most number of bikes are rented**

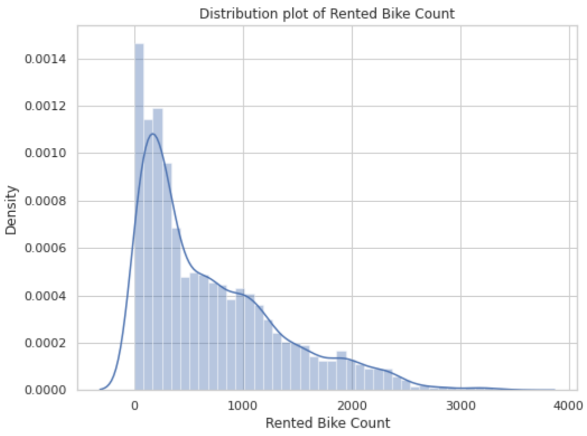
Correlation between numerical features:

- From the above correlation heatmap, we can observe that multicollinearity exits between two features namely Temperature and Dew Point Temperature.
- Humidity, Rainfall, Snowfall are negatively correlated with dependent variable. That means, if the above feature values increases, dependent feature value will decreases and vice versa.
- Temperature, Wind speed, Visibility, Dew point temperature, Solar Radiation are the features which are positively correlated with dependent feature (Rented Bike Count). That means, if the above features values increase, dependent feature value will also increases.

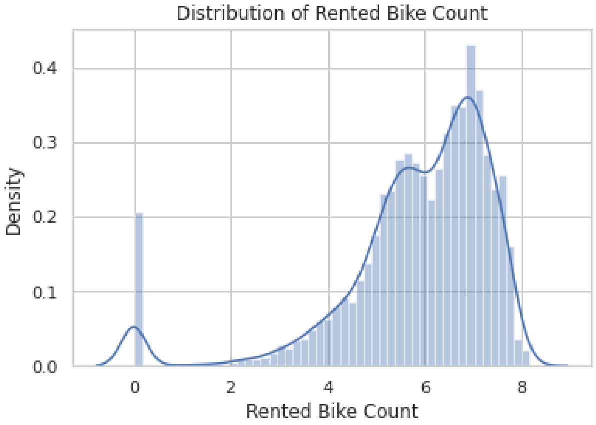
Feature Engineering :

❖ **From Applied log to the values of 'Rented Bike Count' column because the distribution of the feature is positively skewed.**

Before Transformation



After Transformation



Removing Outliers :

```
# Removing Outliers
# Creating a for loop for storing the indeces of outliers
for i in num_feat:
  indeces = []
  x = data[i]
  mean = data[i].mean()
  std = data[i].std()
  index = data[(np.abs(x)) - (mean) >= (3 * std)].index
  indeces.append(index)
```

```
# Displaying the list of indeces of outliers
list(indeces)[0]
```

```
Int64Index([ 222,  223,  224,  225,  226,  227,  228,  229,  230,  415,
            ...
            8620, 8621, 8622, 8623, 8624, 8625, 8626, 8627, 8628, 8629],
           dtype='int64', length=173)
```

```
# Dropping the outliers
new_data.drop(list(indeces)[0] , inplace = True)
```

- At first, created a ‘for’ loop for all the numerical features which appends the outlier index value into a variable.
- After storing the list of outlier indexes, simply dropped them from the data.

- To remove multi-collinearity, we should drop ‘Dew Point Temperature’ feature because it is highly correlated with Temperature feature.
- After storing the list of outlier indexes, simply dropped them from the data.
- Before fitting the data into model, the data should have only numerical values, so we have to change categorical features into numerical features by One Hot Encoding using get_dummies

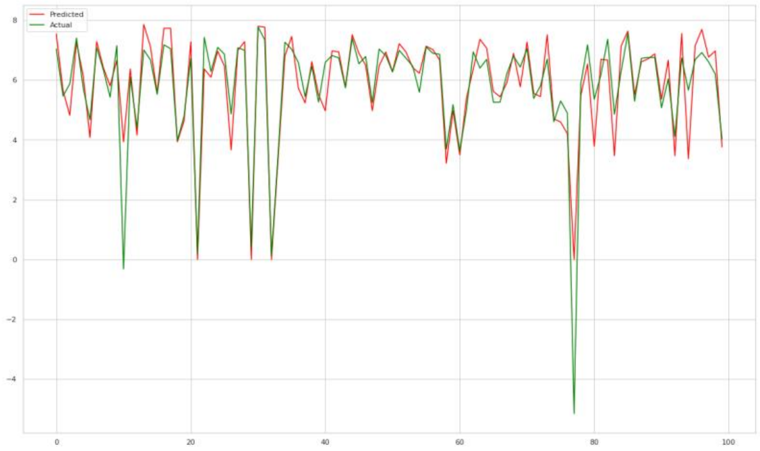
Pre Processing:

- After converting the categorical features into numerical features, the data has only numerical values.
- Feature scaling is a important preprocessing step. So we have to apply MinMaxScaler to the data.
- Features that are measured at different scales do not contribute equally to the model fitting & model learned function and might end up creating a bias.
- Thus, to deal with this potential problem feature-wise normalization such as MinMax Scaling is usually used prior to model fitting.

Model Implementation and Explainability:

Linear Regression

Actual vs predicted graph



By fitting the data into Linear Regression Model, we get

- Training score : 84.02%
- Testing score: 83.61%

Evaluation Metrics for train and test data

```
The evaluation metrics for training dataset.....
```

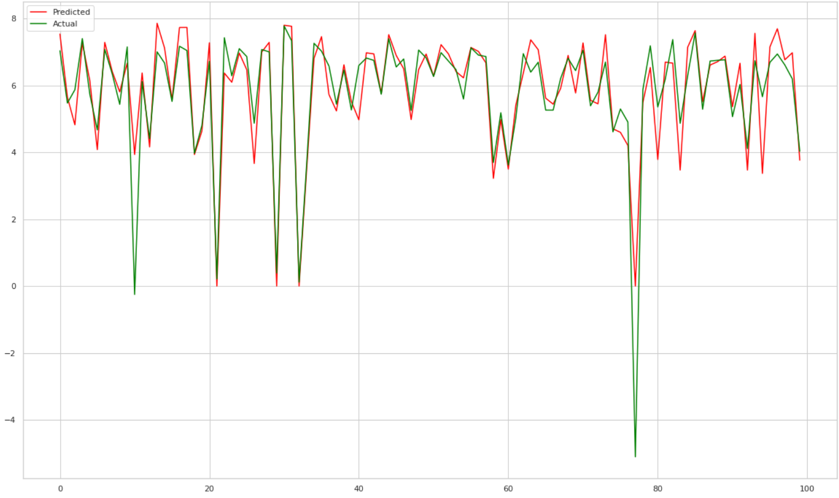
	Metric	value
0	r2_score	0.84022
1	Mean Square Error	0.39996
2	Root mean square error	0.63242
3	Adjusted r2	0.83919
4	Mean absolute error	0.43906

```
The evaluation metrics for test dataset.....
```

	Metric	value
0	r2_score	0.83619
1	Mean Square Error	0.42967
2	Root mean square error	0.65549
3	Adjusted r2	0.83300
4	Mean absolute error	0.44334

Lasso Regression

Actual vs predicted graph



Evaluation Metrics for train and test data

The evaluation metrics for training dataset.....

	Metric	value
0	r2_score	0.84022
1	Mean Square Error	0.39994
2	Root mean square error	0.63241
3	Adjusted r2	0.83920
4	Mean absolute error	0.43911

The evaluation metrics for test dataset.....

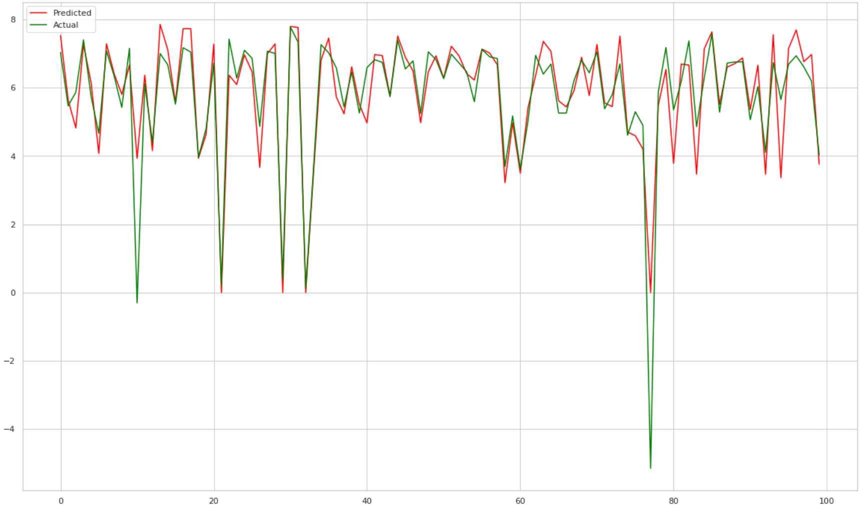
	Metric	value
0	r2_score	0.83654
1	Mean Square Error	0.42875
2	Root mean square error	0.65479
3	Adjusted r2	0.83335
4	Mean absolute error	0.44342

By fitting the data into Lasso Regression Model, we get

- **Training score : 84.02%**
- **Testing score: 83.65%**

Ridge Regression

Actual vs predicted graph



Evaluation Metrics for train and test data

```
The evaluation metrics for training dataset.....
```

	Metric	value
0	r2_score	0.84023
1	Mean Square Error	0.39992
2	Root mean square error	0.63239
3	Adjusted r2	0.83921
4	Mean absolute error	0.43878

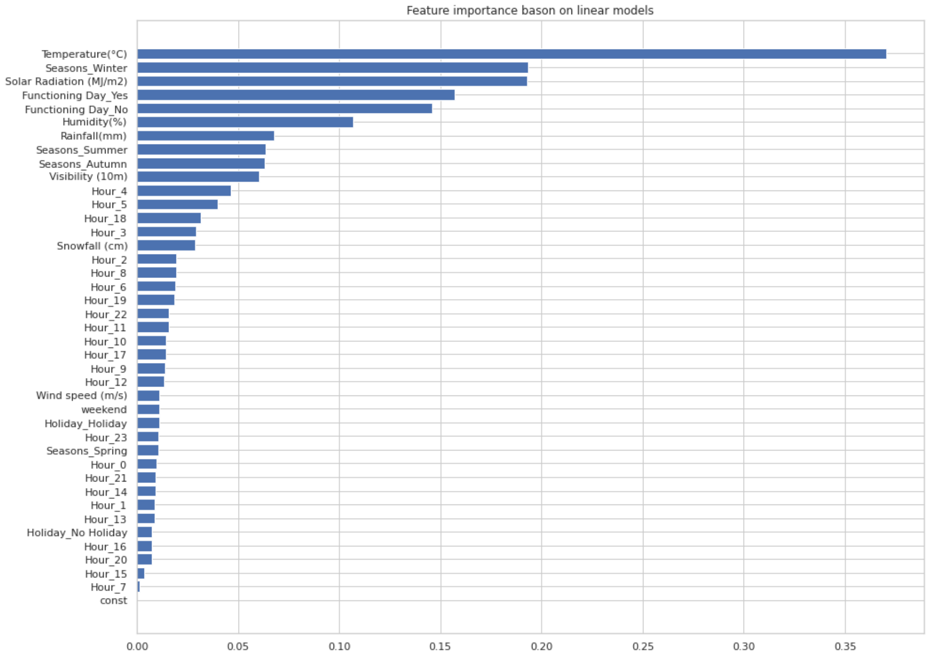
```
The evaluation metrics for test dataset.....
```

	Metric	value
0	r2_score	0.83618
1	Mean Square Error	0.42968
2	Root mean square error	0.65550
3	Adjusted r2	0.83299
4	Mean absolute error	0.44329

By fitting the data into Ridge Regression Model, we get

- **Training score : 84.02%**
- **Testing score: 83.61%**

Feature Importance For Linear Models :

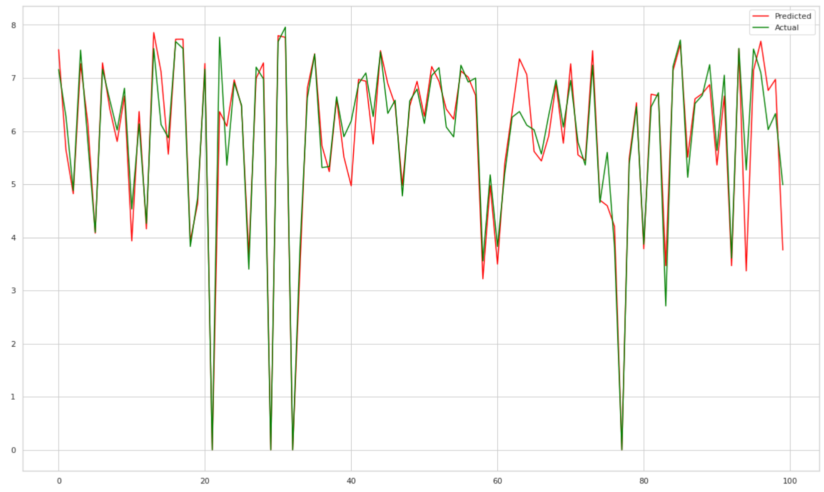


For Linear models,

The feature 'Temperature' has more importance followed by Seasons(Winter), Solar Radiation, Function Day_Yes, Function Day_no, Humidity etc. And the least importance is Hour_7

Decision Tree Regressor :

Actual vs predicted graph



Evaluation Metrics for train and test data

The evaluation metrics for training dataset.....

	Metric	value
0	r2_score	1.00000
1	Mean Square Error	0.00000
2	Root mean square error	0.00000
3	Adjusted r2	1.00000
4	Mean absolute error	0.00000

The evaluation metrics for test dataset.....

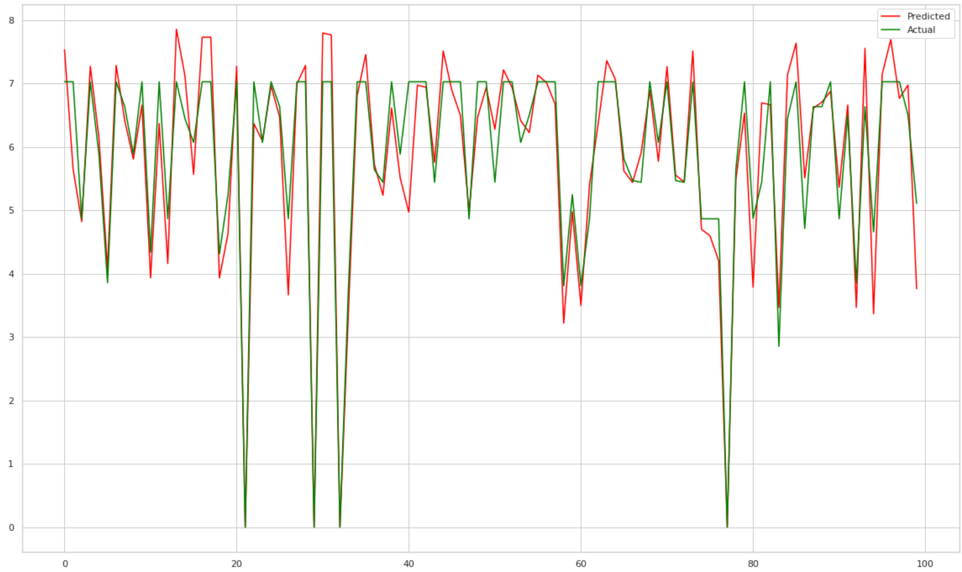
	Metric	value
0	r2_score	0.87567
1	Mean Square Error	0.32611
2	Root mean square error	0.57106
3	Adjusted r2	0.87325
4	Mean absolute error	0.34911

By fitting the data into Decision Tree Regressor Model, we get

- **Training score : 100% (Which means the model is overfitted so we will perform hyper parameter tuning for the model)**
- **Testing score: 87.56%**

Decision Tree Regressor After Hyperparameter Tuning :

Actual vs predicted graph



Evaluation Metrics for train and test data

```
The evaluation metrics for training dataset.....
```

	Metric	value
0	r2_score	0.86087
1	Mean Square Error	0.34826
2	Root mean square error	0.59014
3	Adjusted r2	0.85998
4	Mean absolute error	0.43132

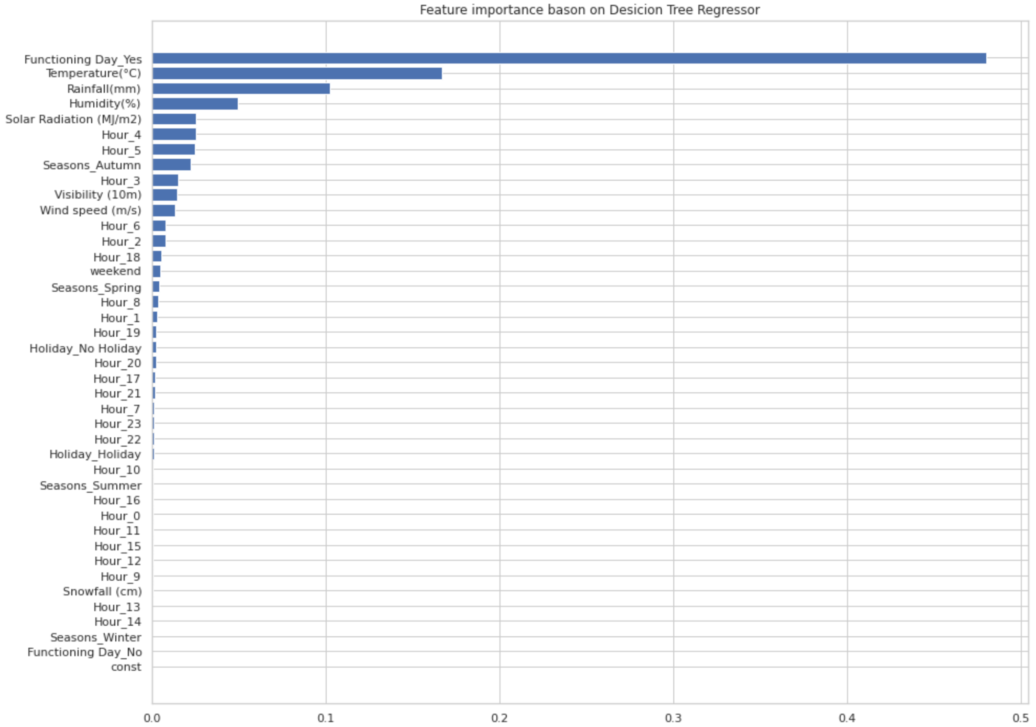
```
The evaluation metrics for test dataset.....
```

	Metric	value
0	r2_score	0.85188
1	Mean Square Error	0.38852
2	Root mean square error	0.62331
3	Adjusted r2	0.84899
4	Mean absolute error	0.45328

After hyper parameter tuning, we get

- Training score : 86.08%
- Testing score: 85.18%

Feature Importance For Decision Tree Regressor :

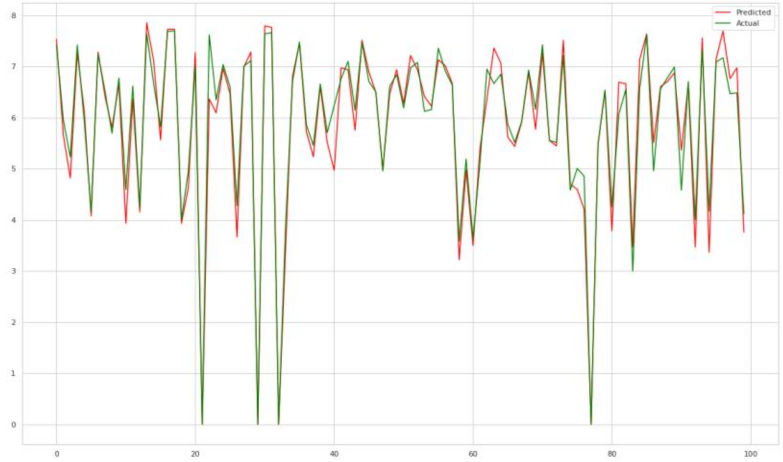


For Decision Tree Regressor model,

The feature ‘Function Day_Yes’ has more importance followed by Temperature, Rainfall, Humidity, Solar Radiation etc. And the least importance is Hour_7

Random Forest Regressor :

Actual vs predicted graph



Evaluation Metrics for train and test data

```
The evaluation metrics for training dataset.....
```

	Metric	value
0	r2_score	0.98975
1	Mean Square Error	0.02565
2	Root mean square error	0.16015
3	Adjusted r2	0.98969
4	Mean absolute error	0.09826

```
The evaluation metrics for test dataset.....
```

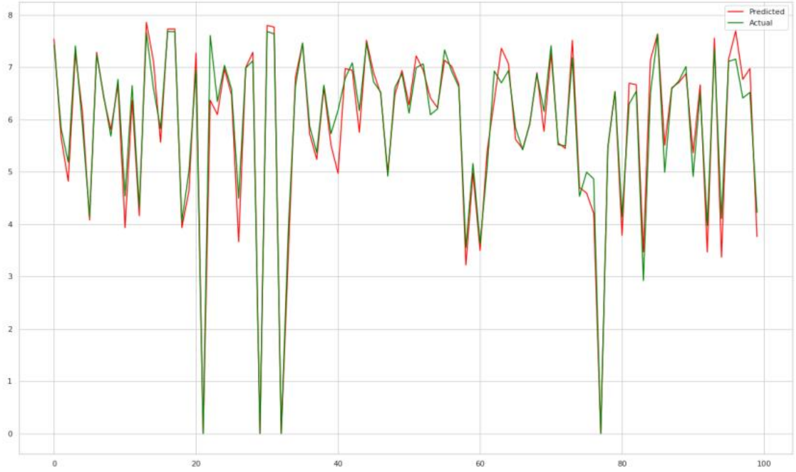
	Metric	value
0	r2_score	0.93594
1	Mean Square Error	0.16802
2	Root mean square error	0.40990
3	Adjusted r2	0.93469
4	Mean absolute error	0.25234

By fitting the data into Random Forest Regressor Model, we get

- **Training score : 98.97% (Which means the model is slightly overfitted, so we will perform hyper parameter tuning for the model)**
- **Testing score: 87.56%**

Random Forest Regressor After Hyperparameter Tuning :

Actual vs predicted graph



After Hyperparameter Tuning

Train data score : 97.34%

Test data score : 93.63%

Evaluation Metrics for train and test data

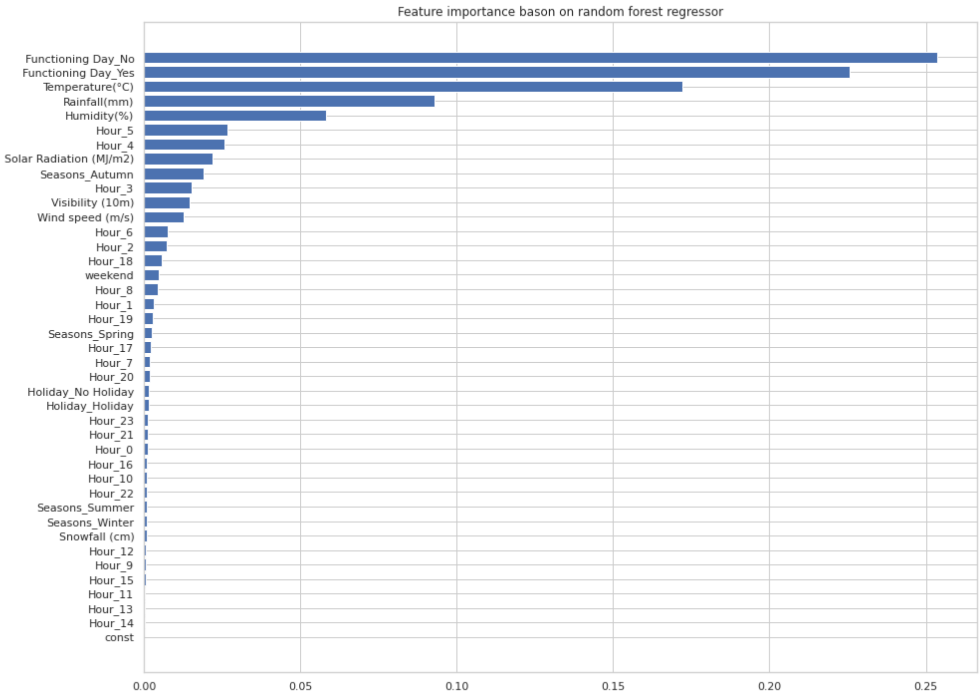
The evaluation metrics for training dataset.....

	Metric	value
0	r2_score	0.97340
1	Mean Square Error	0.06658
2	Root mean square error	0.25804
3	Adjusted r2	0.97323
4	Mean absolute error	0.15399

The evaluation metrics for test dataset.....

	Metric	value
0	r2_score	0.93637
1	Mean Square Error	0.16689
2	Root mean square error	0.40852
3	Adjusted r2	0.93514
4	Mean absolute error	0.25372

Feature Importance For Random Forest Regressor:



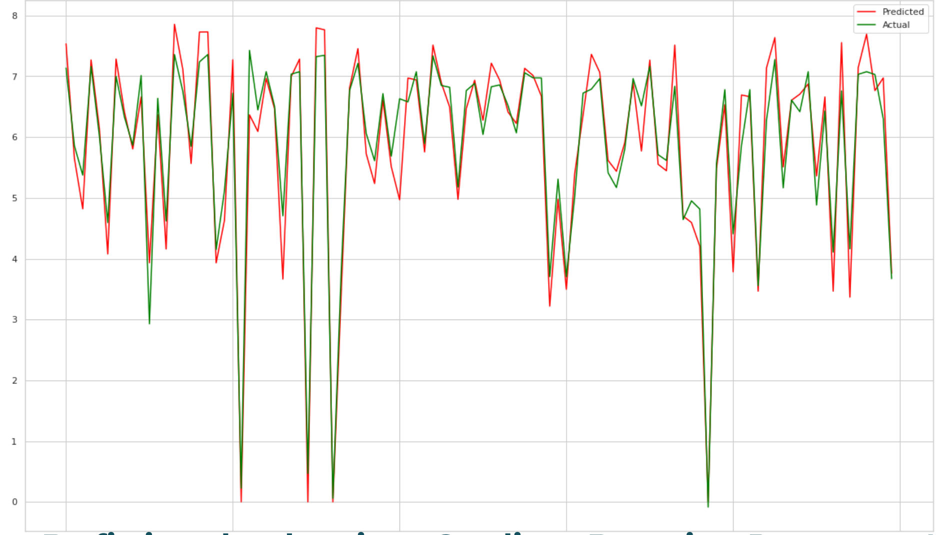
For Random Forest Regressor model,

The feature 'Function Day' has more importance followed by Temperature, Rainfall, Humidity etc.

And the least importance is Hour_14

Gradient Boosting Regressor :

Actual vs predicted graph



Evaluation Metrics for train and test data

The evaluation metrics for training dataset.....

	Metric	value
0	r2_score	0.91641
1	Mean Square Error	0.20922
2	Root mean square error	0.45741
3	Adjusted r2	0.91588
4	Mean absolute error	0.32379

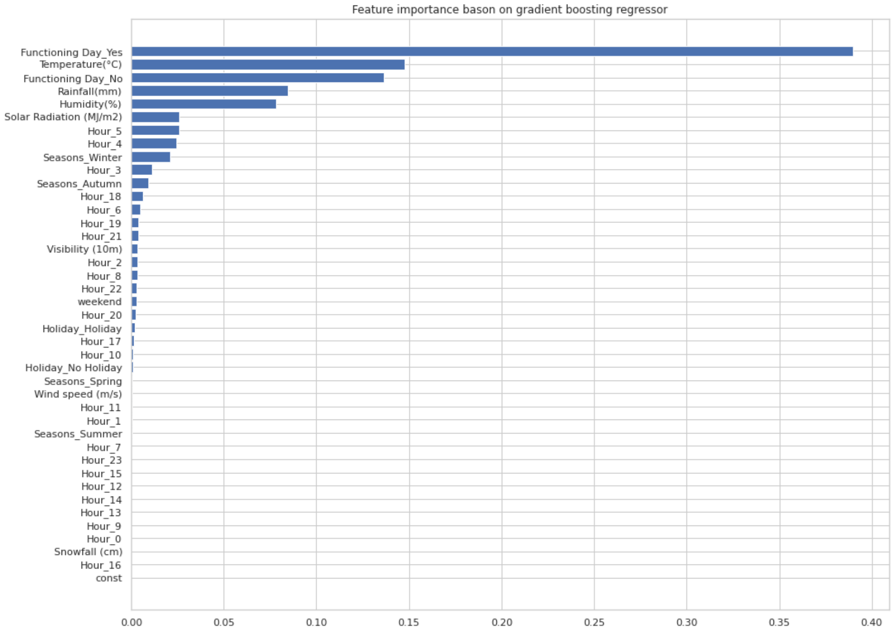
The evaluation metrics for test dataset.....

	Metric	value
0	r2_score	0.91248
1	Mean Square Error	0.22957
2	Root mean square error	0.47913
3	Adjusted r2	0.91077
4	Mean absolute error	0.33309

By fitting the data into Gradient Boosting Regressor Model, we get

- Training score : 91.64%
- Testing score: 91.24%

Feature Importance For Gradient Boosting Regressor :

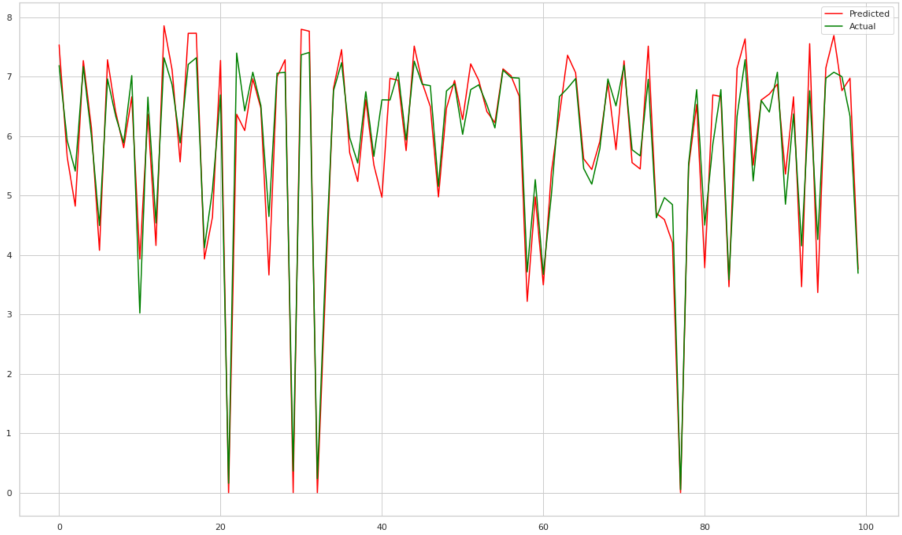


For Decision Tree Regressor model,

The feature 'Function Day_Yes' has more importance followed by Temperature, Function_Day_No, Rainfall, Humidity, Solar Radiation etc. And the least importance is Hour_16

XG Boost Regressor :

Actual vs predicted graph



Evaluation Metrics for train and test data

The evaluation metrics for training dataset.....

	Metric	value
0	r2_score	0.91670
1	Mean Square Error	0.20851
2	Root mean square error	0.45663
3	Adjusted r2	0.91617
4	Mean absolute error	0.32319

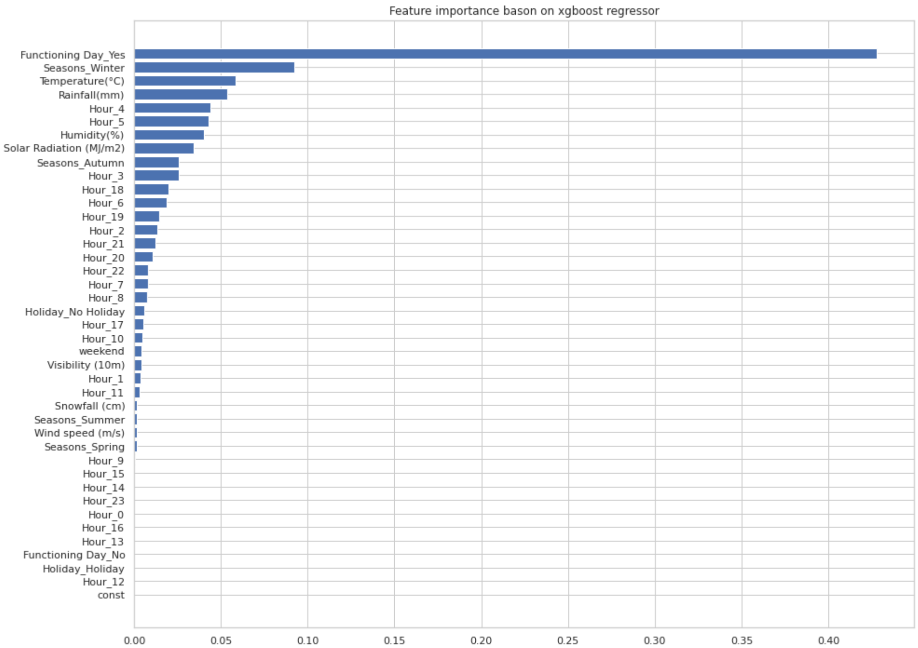
The evaluation metrics for test dataset.....

	Metric	value
0	r2_score	0.91223
1	Mean Square Error	0.23022
2	Root mean square error	0.47982
3	Adjusted r2	0.91052
4	Mean absolute error	0.33356

By fitting the data into XGBoost Regressor Model, we get

- **Training score : 91.67%**
- **Testing score: 91.22%**

Feature Importance For XGBoost Regressor :



For XGBoost Regressor model,

The feature 'Function Day_Yes' has more importance followed by Season_Winter, Temperature, Rainfall, etc. And the least importance is Hour_12

Conclusion :

- From Exploratory Data Analysis, we can conclude that,
- Most Number of bike rented count ranges from 0 to 500.
- Temperature mostly varies from 20 to 30.
- Humidity mostly varies from 20 to 100.
- Wind speed mostly varies from 2 to 4 m/s.
- Visibility of 2000 count is high.
- Solar Radiation is mostly 0. And a few are in range of 1 to 4.
- Mostly there is no rainfall and snowfall. And a very few have rainfall and snowfall.
- The distribution between numerical features and dependent feature (Rented Bike Count) has been spread out entire area which means there is no specific relation between them.

- Most number of bikes are rented in the hour of 18 followed by 19th hour. And the least is 4th hour
- In summer season, most number of bikes are rented and the least is winter season.
- In working days(No Holiday), most number of bikes are rented.
- Thursday has high count of rented bike.
- In a functioning day, most number of bikes are rented.
- Multicollinearity exits between two features namely Temperature and Dew Point Temperature.
- Where Temperature is 54% correlated with dependent feature,
- Dew Point Temperature is 38% correlated with dependent feature.

  Hence, we can remove Dew Point Temperature.

After fitting the data into various regression models, we can conclude that

- ❖ Tree based models performs well than linear models because, the independent features are not linearly related to the dependent feature ('Rented Bike Count').
- ❖ For Linear Models

1. Linear regression
    - Train data score : 84.01%
    - Test data score : 83.60%
2. Lasso regression
    - Train data score : 84.02%
    - Test data score : 83.65%
3. Ridge Regression
    - Train data score : 84.02%
    - Test data score : 83.61%

- ❖ For Linear Models (Linear Regression, Lasso Regression, Ridge Regression), 'Temperature' has more importance and least is 'Hour 7'.
- ❖ For Decision Tree Regressor,
    - ❖ Train data score : 100% (which means the model is overfitted)
    - ❖ Test data score : 86.88%
- ❖ After Hyperparameter Tuning
    - ❖ Train data score : 88.13%
    - ❖ Test data score : 87.06%

'Functioning Day_Yes' has more importance and the least is 'Functioning Day_No'

- For Random Forest Regressor,
  - Train data score : 98.95% (which means the model is slightly overfitted)
  - Test data score : 93.66%
- After Hyperparameter Tuning
  - Train data score : 97.25%
  - Test data score : 93.67%
- 'Functioning Day_Yes' has more importance and the least is 'Hour_14'
- For Gradient Boosting Regressor,
  - Train data score : 91.61%
  - Test data score : 91.25%
- 'Functioning Day_Yes' and has more importance and the least is 'Hour_16'
- For XGBoost Regressor,
  - Train data score : 91.61%
  - Test data score : 91.22%
- 'Functioning Day_No' has more importance and the least is 'Hour_16'

THANK YOU