

Bike Sharing Demand Prediction

Rochak P V,
Data science trainee,
AlmaBetter

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 the bike count required at each hour for the stable supply of rental bikes.

2. Objective

As mentioned in the problem statement, the objective of this project is to develop a supervised machine learning regression model that can predict the target variable (rented bikes count) for a city, with the help of given feature variables. Regression models are used to predict continuous target variables.

In this project, the steps involved are:

1. Data Description: Understanding the characteristics of the dataset in hand.
2. Initial data prepping, which includes,
 - Handling NaN values
 - Necessary feature changes
3. Exploratory Data Analysis to understand the underlying patterns in the data.
4. Final preprocessing.
 - Feature conditioning and outlier handling.
 - Feature selection using correlation and multicollinearity analysis.
 - One hot encoding on categorical features.
5. Model Implementation and prediction

3. Methodology

Regression analysis:

Regression analysis is a reliable method of identifying which variables have an impact on a topic of interest. The process of performing a regression allows you to confidently determine which factors matter most, which factors can be ignored, and how these factors influence each other

1. Linear regression:

Linear regression is one of the most basic types of regression in supervised machine learning. The linear regression model consists of a predictor variable and a dependent variable related linearly to each other. We try to find the relationship between the independent variable(input) and a corresponding dependent variable (output).

2. Regularized Linear Regression (Ridge, Lasso and elastic net):

Regularized linear regression models are very similar to least squares, except that the coefficients are estimated by minimizing a slightly different objective function. we minimize the sum of RSS and a "penalty term" that penalizes coefficient size.

Ridge regression (or "L2 regularization") minimizes:

$$RSS + \lambda \sum_{j=1}^p \beta_j^2$$

Lasso regression (or "L1 regularization") minimizes:

$$RSS + \lambda \sum_{j=1}^p |\beta_j|$$

Where λ is a tuning parameter that seeks to balance the fit of the model to the data and the magnitude of the model's coefficients:

- A tiny λ imposes no penalty on the coefficient size and is equivalent to a normal linear regression.
- Increasing λ penalizes the coefficients and thus shrinks them toward zero.

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

3. Random forest:

Random forest is a supervised learning algorithm that uses an ensemble learning method for classification and regression. Random forest is a bagging technique. The trees in random forests run in parallel, meaning is no interaction between these trees while building the trees. Random forest operates by constructing a multitude of decision trees at training time and outputting the class that's the mode of the classes (classification) or mean prediction (regression) of the individual trees.

4. Libraries Used:

- **NumPy:**

NumPy is a library for the Python language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- **Pandas:**

Pandas is a fast, powerful, flexible and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language.

- **Seaborn:**

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

- **Plotly:**

Plotly again is a library used to generate visualizations. Plotly in addition has many interactive visualizations that can be helpful to draw out insights.

- **Matplot:**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

- **Scikit Learn:**

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5. Data Description: Understanding the characteristics of the dataset in hand

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information. The dataset contains 8760 rows and 14 columns.

Column Information:

1. **Date:** year-month-day
2. **Rented Bike count** – Number of bikes rented at each hour
3. **Hour** - Hour of the day
4. **Temperature**-Temperature at the time of booking in Celsius
5. **Humidity** – Humidity is a measure of the amount of water vapor in the air expressed in %
6. **Windspeed** -Speed of wind at the time of booking expressed in m/s
7. **Visibility** – Driving visibility expressed in m.
8. **Dew point temperature** - The dew point is the temperature expressed in Celsius, at which air is saturated with water vapor, which is the gaseous state of water.
9. **Solar radiation** - Amount of solar radiation at the time of booking expressed in MJ/m²
10. **Rainfall** - Amount of rainfall at the time of booking expressed in mm
11. **Snowfall** - Amount of snowing at the time of booking expressed in cm
12. **Seasons** - Winter, Spring, Summer, Autumn
13. **Holiday** - whether the day is considered a holiday expressed as Holiday/No holiday
14. **Functional Day** - NoFunc(Non Functional Hours), Fun(Functional hours)

Here our target feature is ‘Rented Bike count’.

A brief statistics description of all numerical features is given below,

	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m ²)	Rainfall(mm)	Snowfall (cm)
count	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000
mean	704.602055	11.500000	12.882922	58.226256	1.724909	1436.825799	4.073813	0.569111	0.148687	0.075068
std	644.997468	6.922582	11.944825	20.362413	1.036300	608.298712	13.060369	0.868746	1.128193	0.436746
min	0.000000	0.000000	-17.800000	0.000000	0.000000	27.000000	-30.600000	0.000000	0.000000	0.000000
25%	191.000000	5.750000	3.500000	42.000000	0.900000	940.000000	-4.700000	0.000000	0.000000	0.000000
50%	504.500000	11.500000	13.700000	57.000000	1.500000	1698.000000	5.100000	0.010000	0.000000	0.000000
75%	1065.250000	17.250000	22.500000	74.000000	2.300000	2000.000000	14.800000	0.930000	0.000000	0.000000
max	3556.000000	23.000000	39.400000	98.000000	7.400000	2000.000000	27.200000	3.520000	35.000000	8.800000

Brief information regarding the dataset such as NaN value count and data type is given below,

```

RangeIndex: 8760 entries, 0 to 8759
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Date                                  8760 non-null   object
1   Rented Bike Count                    8760 non-null   int64
2   Hour                                 8760 non-null   int64
3   Temperature(°C)                     8760 non-null   float64
4   Humidity(%)                          8760 non-null   int64
5   Wind speed (m/s)                    8760 non-null   float64
6   Visibility (10m)                     8760 non-null   int64
7   Dew point temperature(°C)           8760 non-null   float64
8   Solar Radiation (MJ/m2)             8760 non-null   float64
9   Rainfall(mm)                        8760 non-null   float64
10  Snowfall (cm)                       8760 non-null   float64
11  Seasons                             8760 non-null   object
12  Holiday                             8760 non-null   object
13  Functioning Day                      8760 non-null   object
dtypes: float64(6), int64(4), object(4)

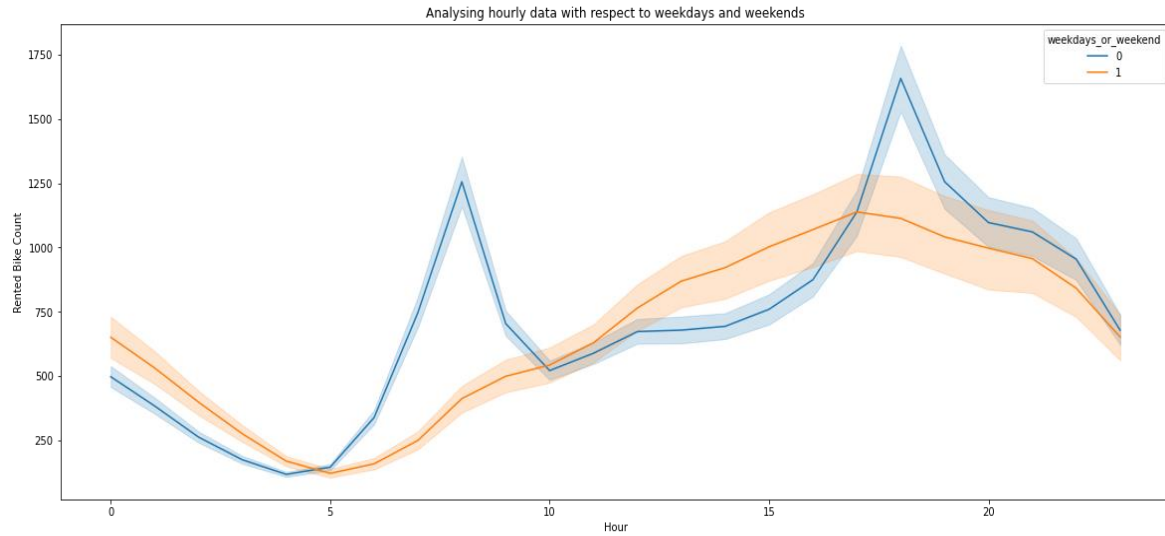
```

6. Initial preprocessing:

- Handling NaN values:
Since the dataset does not contain any NaN values, the dataset does not require any NaN value treatment.
- Feature alteration:
 - Here 'date' feature is converted into 3 different categorical features (day, month and year) for further analysis and using 'day' a new categorical feature 'weekday_or_weekend' is created which records whether the given day is a weekday or weekend.
 - Some of the numerical features i.e., 'Solar Radiation (MJ/m2)', 'Rainfall(mm)', 'Snowfall (cm)' is converted to categorical features.
 - In the case of 'Solar Radiation (MJ/m2)' if the value is zero then it is recorded as 'low radiation', if the value is greater than 0 but less than 0.93 then it is recorded as 'Moderate radiation' and if value greater than 0.93 it is recorded as 'high radiation'.
 - In the case of 'Rainfall(mm)' and 'Snowfall (cm)' if the value is zero then it is recorded as 'No rain' and 'No snowfall' else 'rain' and 'snow fall'.

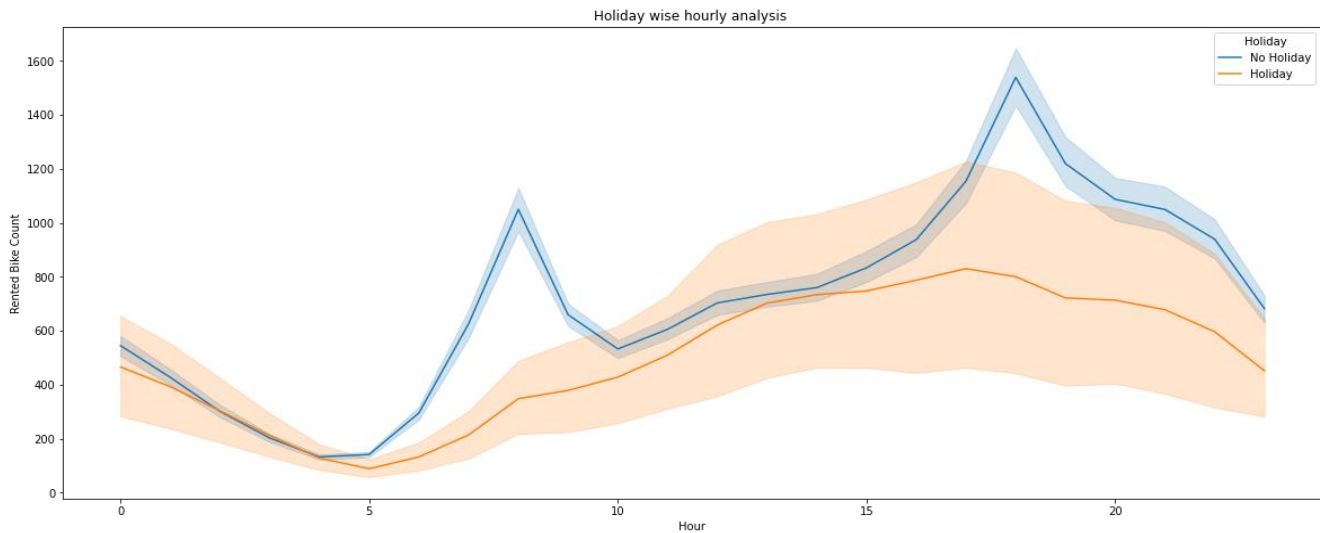
7. Exploratory Data Analysis:

After data cleaning and necessary feature changes, an exploratory data analysis is carried out to get meaningful insights, understand the underlying patterns, and get a bird's eye view of the data. Below mentioned are some of the key findings,



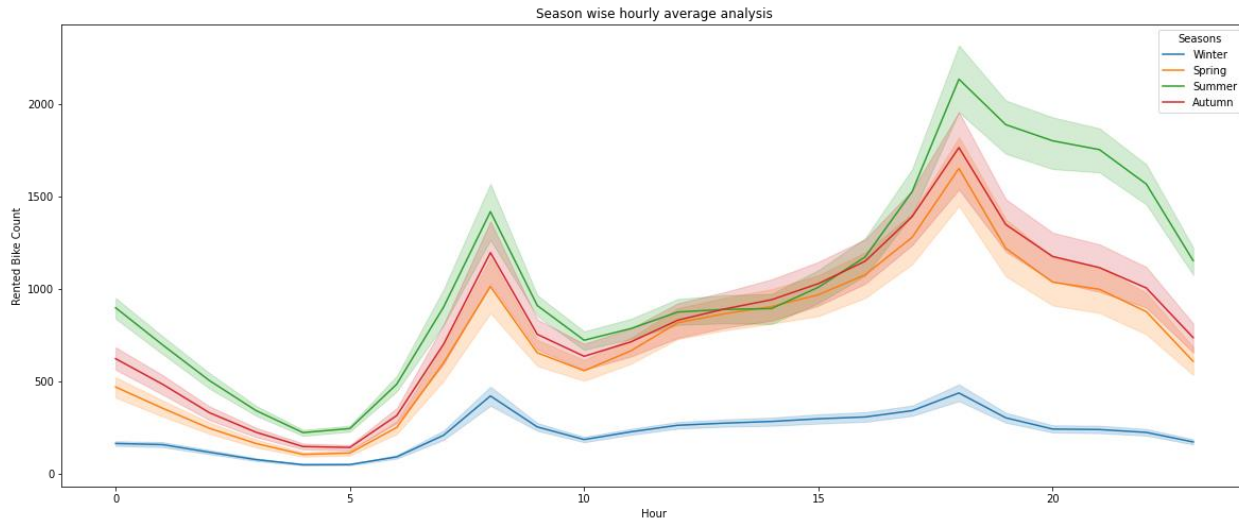
In case of weekdays and weekends,

- During weekdays two peak times are recorded at 5 am to 10 am and 4 pm to 8 pm
- No such peak times were found in the case of weekends
- without considering the peak times of weekdays, both the graph follows the same trajectory.
- Average use of the bike is usually high on weekdays compared to weekends.
- The average bike rented on weekdays is 719 and during weekends it is 667.



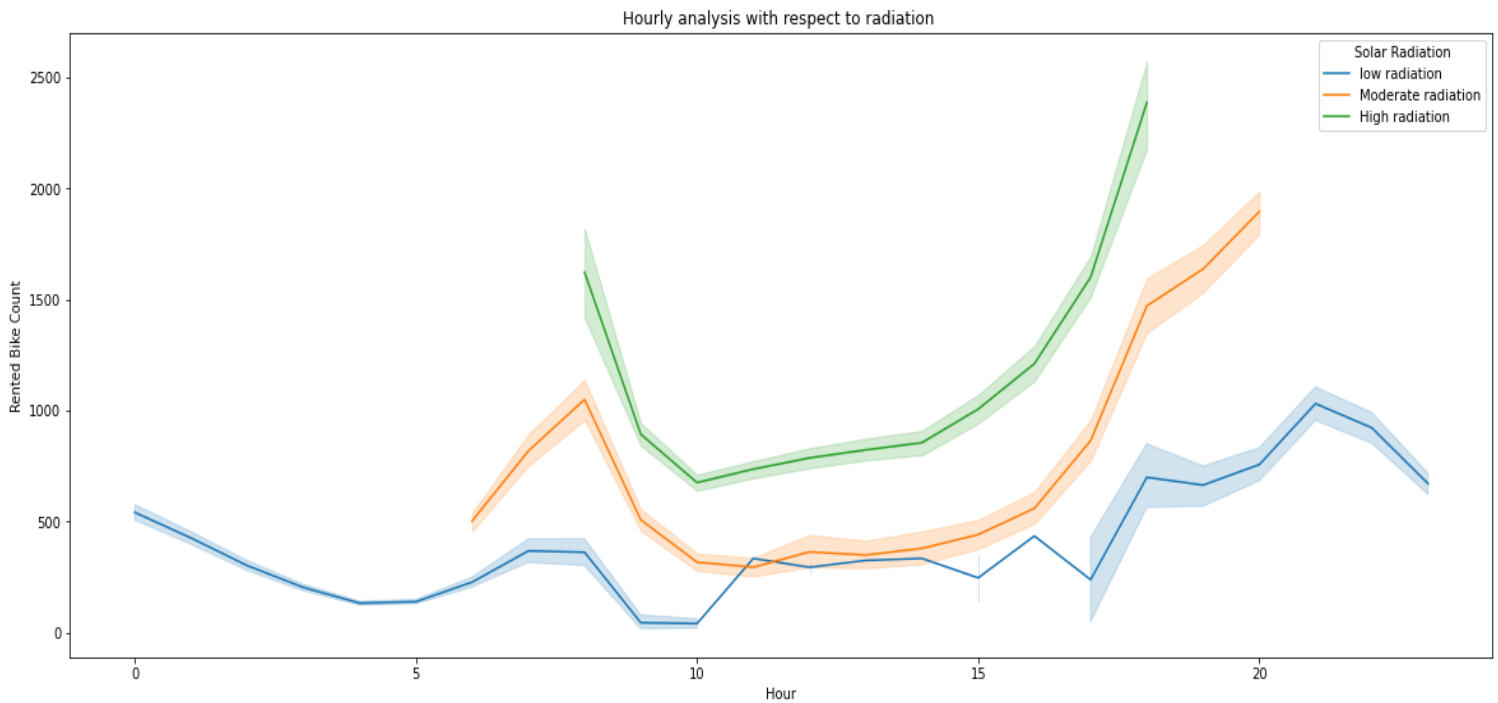
In the case of holidays and non-holidays,

- Patterns are in a way similar to that of Weekdays and weekends
- The average bike rented on holidays is 500 and during weekends it is 715.



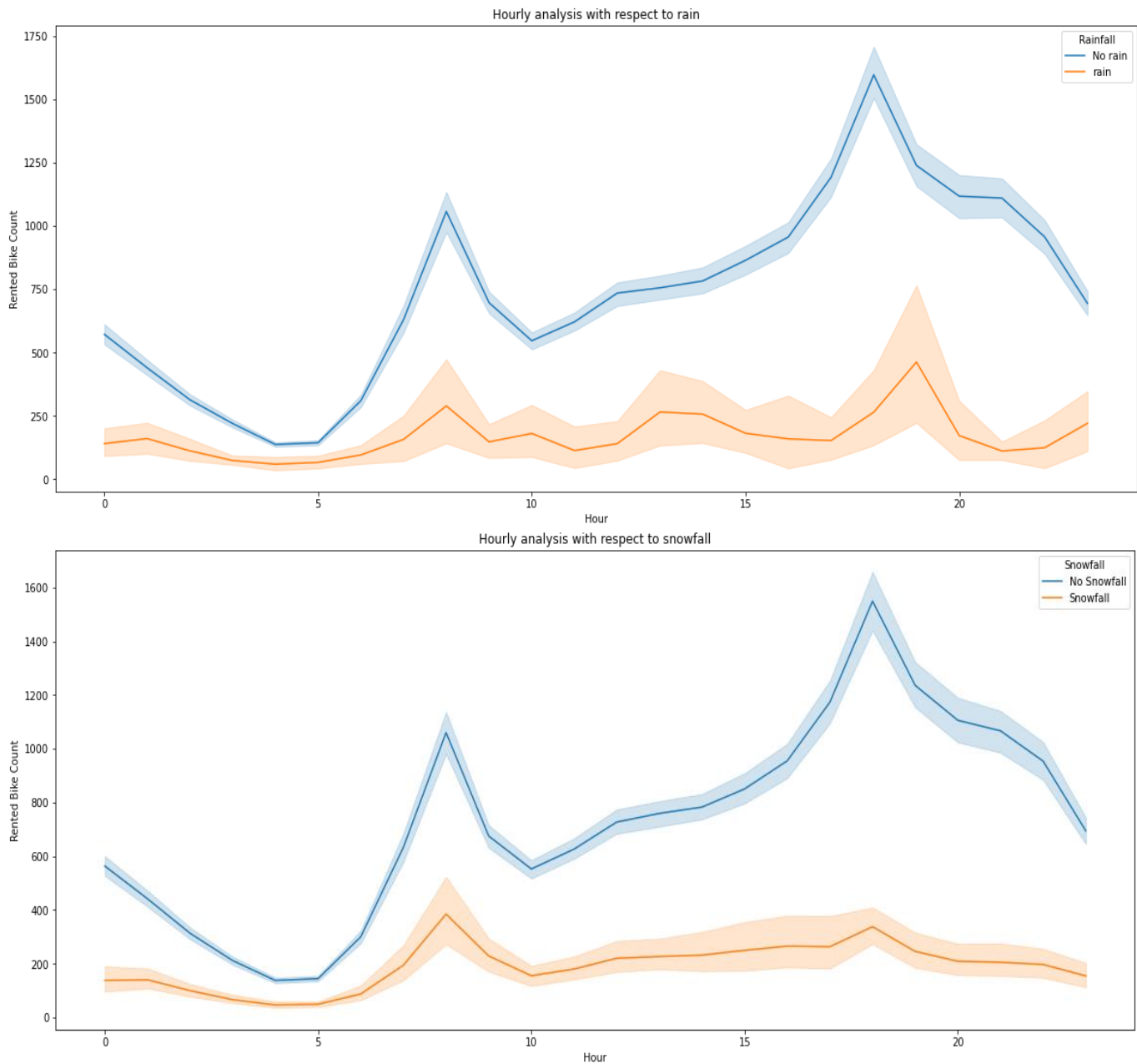
In the case of season-wise analysis,

- Use of bikes is high during summer followed by autumn, spring and then winter
- All the season except winter follows almost similar hourly patterns.
- The average use of bikes during summer is 1034, during autumn 819, during spring 730 and winter 225.



In the case of solar radiation-based analysis,

- Bike use is high during high radiation period followed by moderate and then low
- Average bike rented during high radiation is 967 followed by moderate with 863 and then low with 487.



In the case of rainfall and snowfall-based analysis,

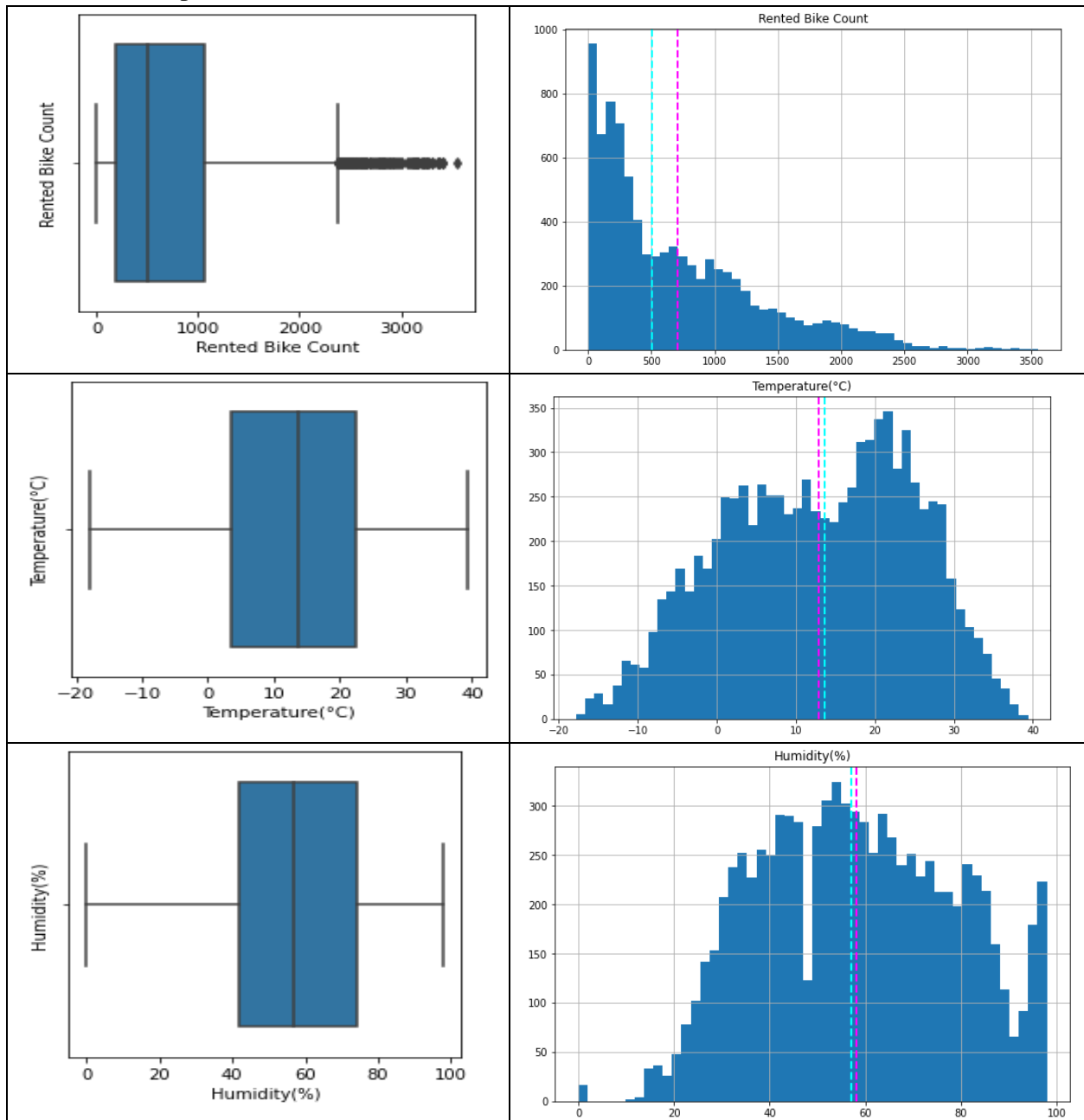
- People prefer bikes very less during rainfall and snowfall.
- The average bikes rented during rainfall and snowfall are 163 and 185

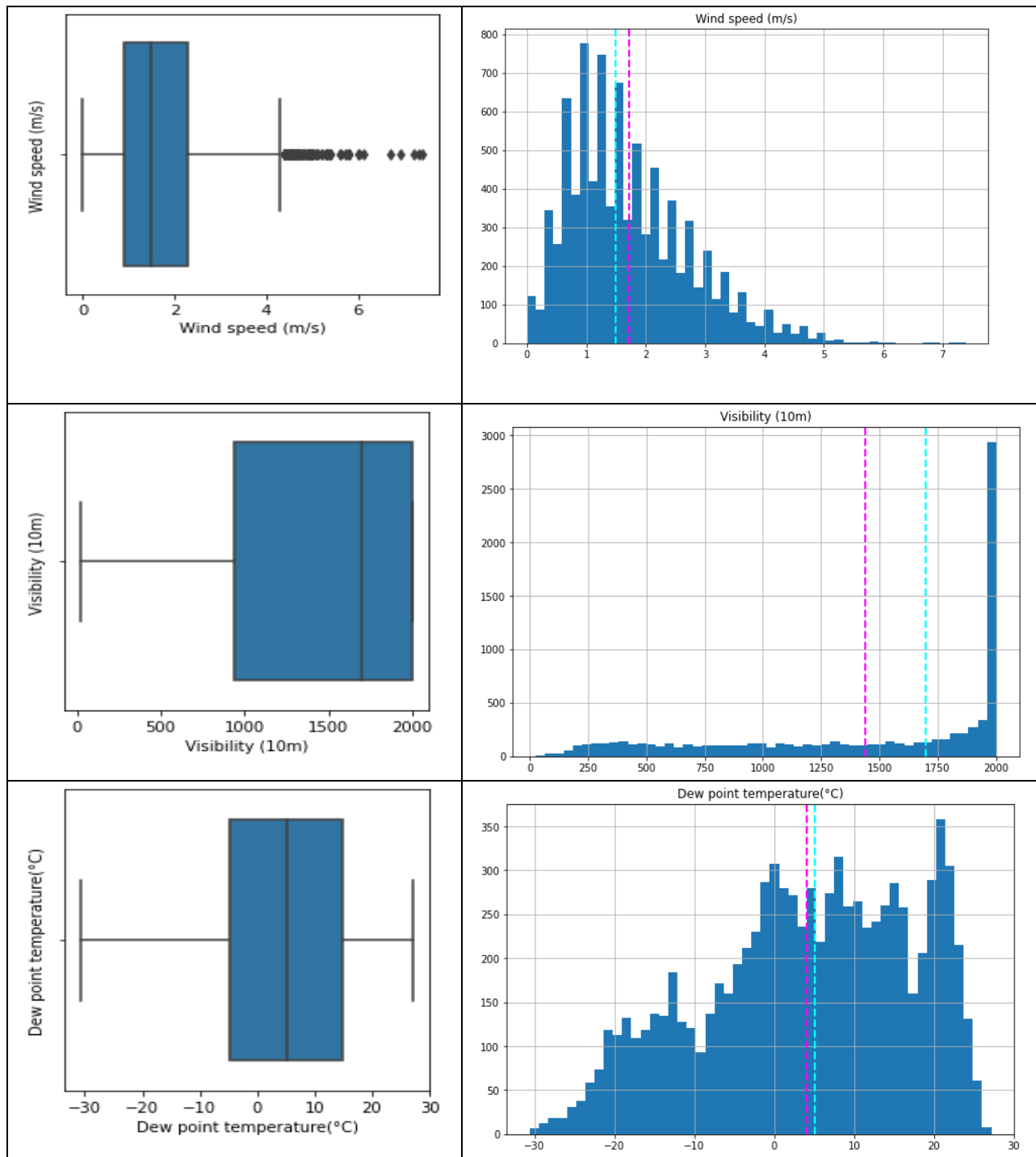
8. Final preprocessing:

Before implementing regression models the dataset requires some pre-processing such as feature conditioning, outlier handling, feature selection based on Correlation and multicollinearity analysis and One hot encoding on categorical features.

1. Feature conditioning & outlier handling:

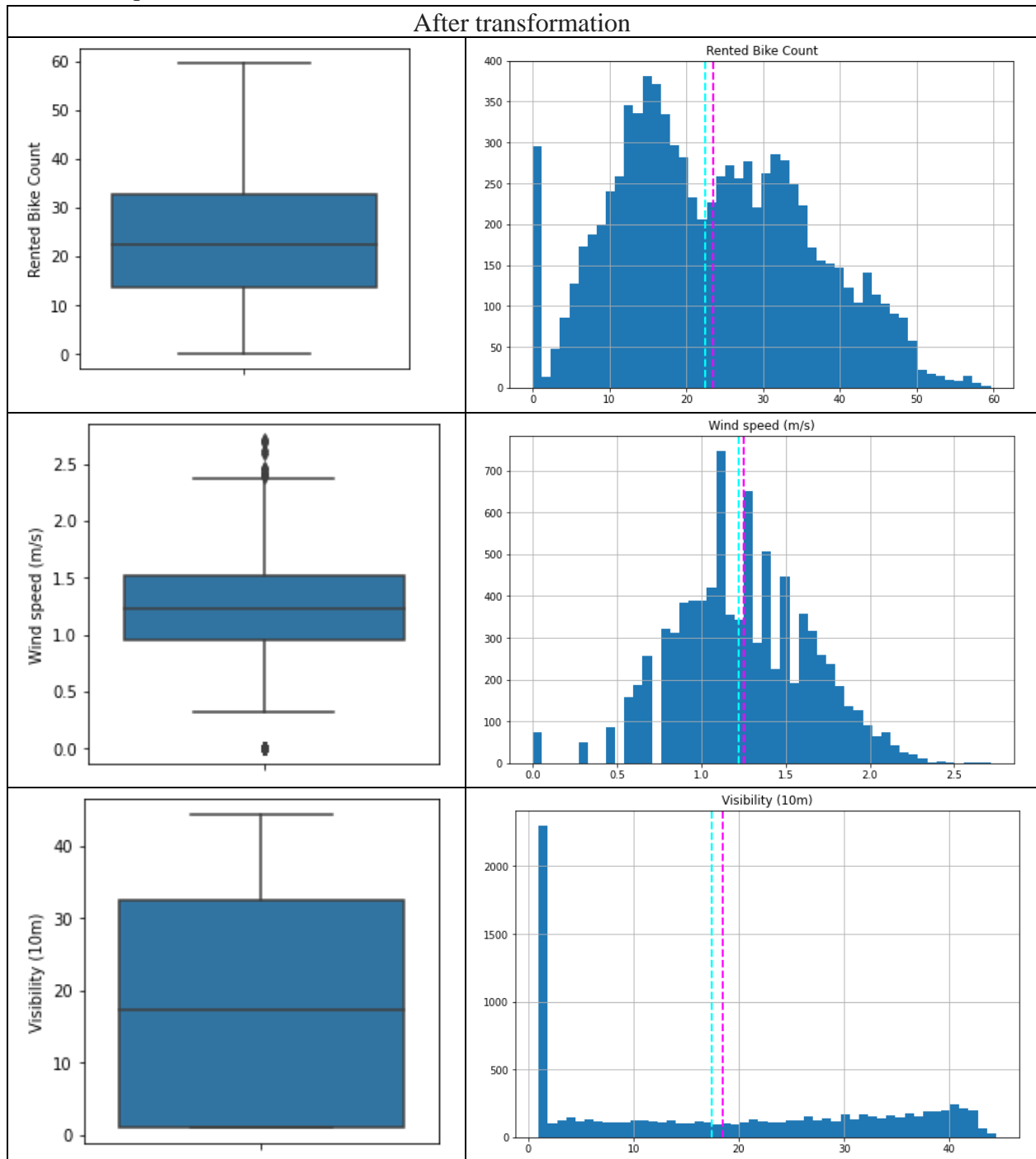
For analyzing feature data distribution and outliers, density plots and box plots are used,





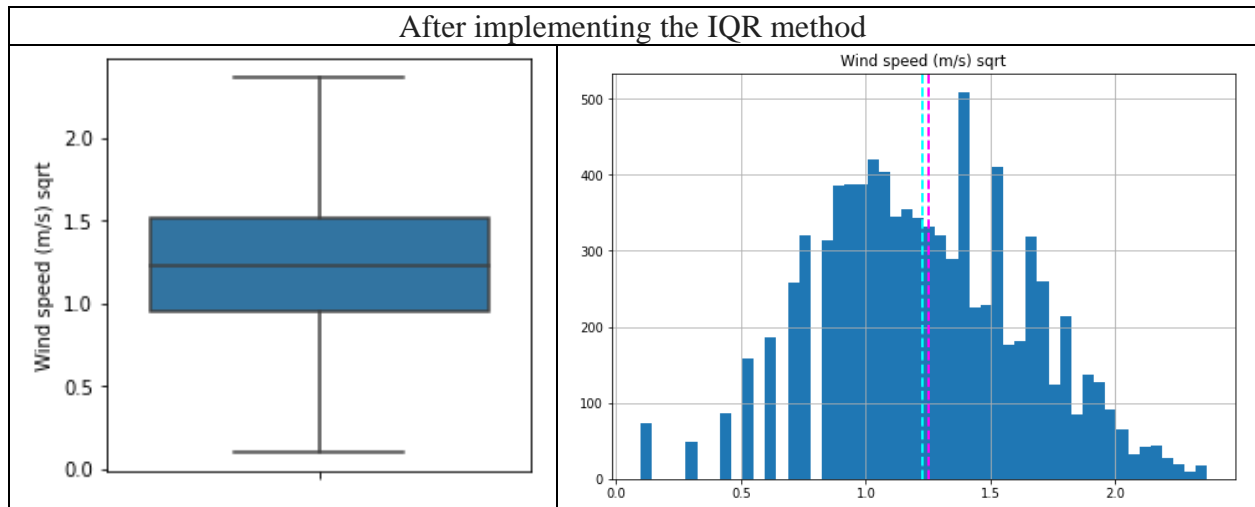
- Here positive skewness is observed in the 'Rented Bike Count' & 'Wind speed (m/s)' features and negative skewness is observed in the 'Visibility (10m)' feature.
- Outliers are observed in 'Rented Bike Count' & 'Wind speed (m/s)' features

To resolve positive skewness, root transformation is used and for negative skewness $\sqrt{\max(x+1) - x}$ transformation is used.



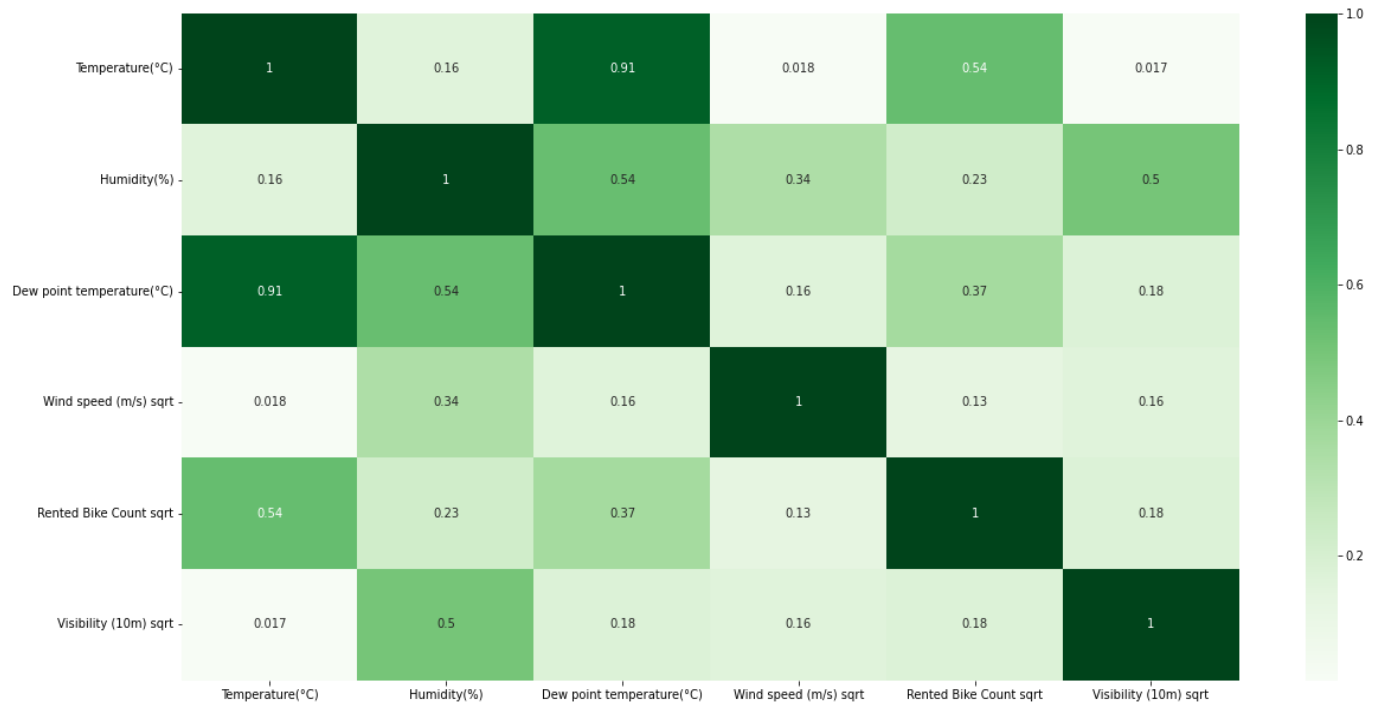
- Acceptable distribution obtained in all features after transformation.
- Outliers are still observed in the 'Wind speed (m/s)' feature.

To eliminate outliers in the 'Wind speed (m/s)' feature IQR (Interquartile range) method was used after square root transformation.



2. Feature selection using Correlation and multicollinearity analysis:

A linear regression model performs well when the independent features are carefully selected. The correlation between these features and the target feature should be good and free of multicollinearity. Multicollinearity occurs when independent features are correlated. Below is a heatmap that shows the correlation between features



Below is a table consisting of the VIF score of all features,

variables	VIF
Temperature(°C)	3.346457
Humidity(%)	7.514150
Wind speed (m/s) sqrt	6.009726
Rented Bike Count sqrt	6.079033
Visibility (10m) sqrt	3.440204

- Since 'Dew point temperature(°C)' and 'Temperature(°C)' has a correlation of 0.91, 'Dew point temperature(°C)' will be dropped from further analysis.
- VIF values of all the features are in the acceptable range.

3. One hot encoding on categorical features:

The categorical feature that underwent one hot encoding are ['Hour', 'Seasons', 'Holiday', 'Functioning Day', 'Solar Radiation', 'Rainfall', 'Snowfall', 'month', 'weekdays_or_weekend']

9. Model Implementation and prediction:

For model implementation the independent variables selected are ['Temperature(°C)', 'Humidity(%)', 'Wind speed (m/s) sqrt', 'Visibility (10m) sqrt', 'Hour_1', 'Hour_2', 'Hour_3', 'Hour_4', 'Hour_5', 'Hour_6', 'Hour_7', 'Hour_8', 'Hour_9', 'Hour_10', 'Hour_11', 'Hour_12', 'Hour_13', 'Hour_14', 'Hour_15', 'Hour_16', 'Hour_17', 'Hour_18', 'Hour_19', 'Hour_20', 'Hour_21', 'Hour_22', 'Hour_23', 'Seasons_Spring', 'Seasons_Summer', 'Seasons_Winter', 'Holiday_No Holiday', 'Functioning Day_Yes', 'Solar Radiation_Moderate radiation', 'Solar Radiation_low radiation', 'Rainfall_rain', 'Snowfall_Snowfall', 'month_2', 'month_3', 'month_4', 'month_5', 'month_6', 'month_7', 'month_8', 'month_9', 'month_10', 'month_11', 'month_12', 'weekdays_or_weekend_1'] and the dependent variable is ['Rented Bike Count sqrt'] .

The dataset is split into test and train with a test size of 25%.

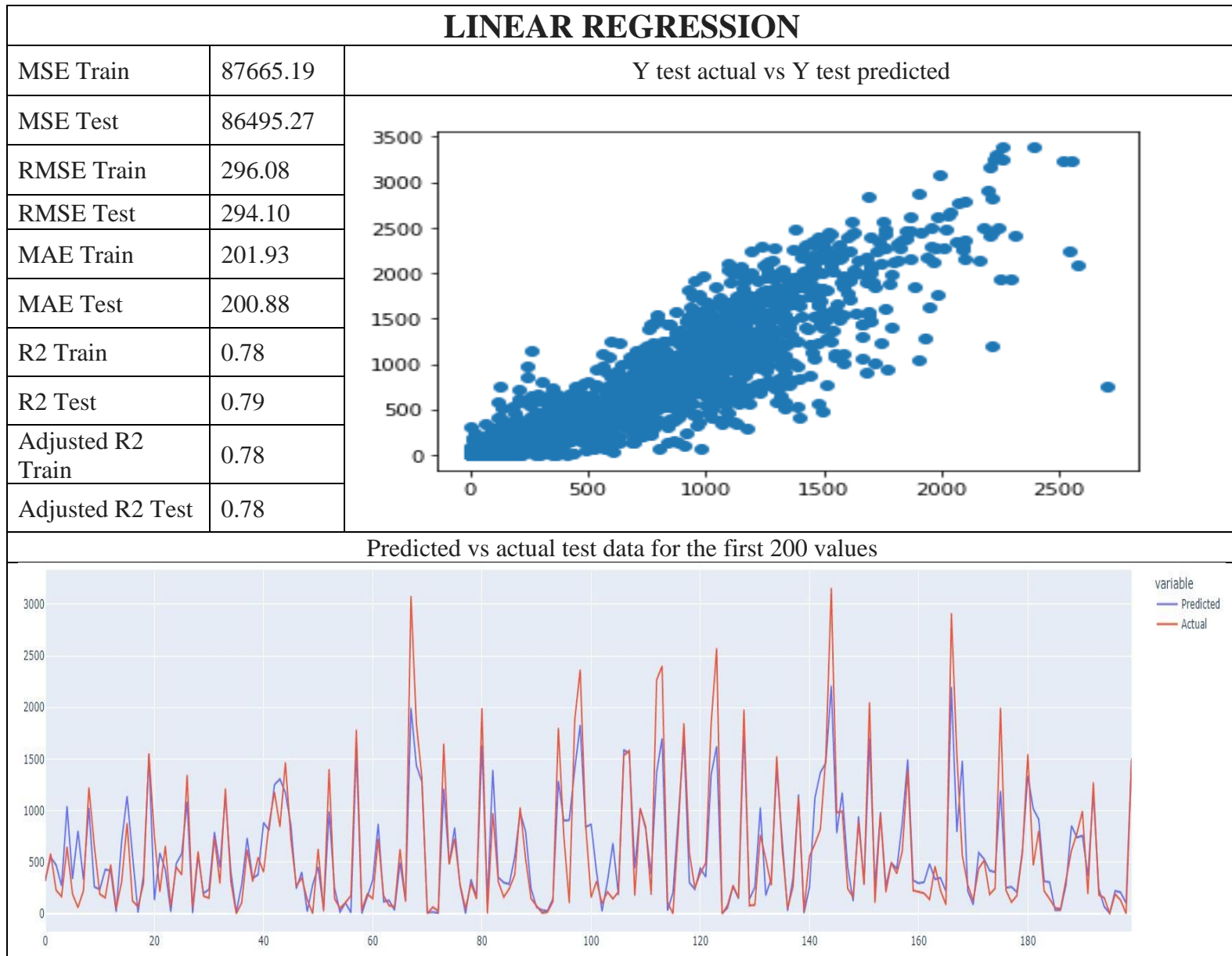
X train shape : (6570, 48)

Y train shape : (6570,1)

X test shape : (2190, 48)

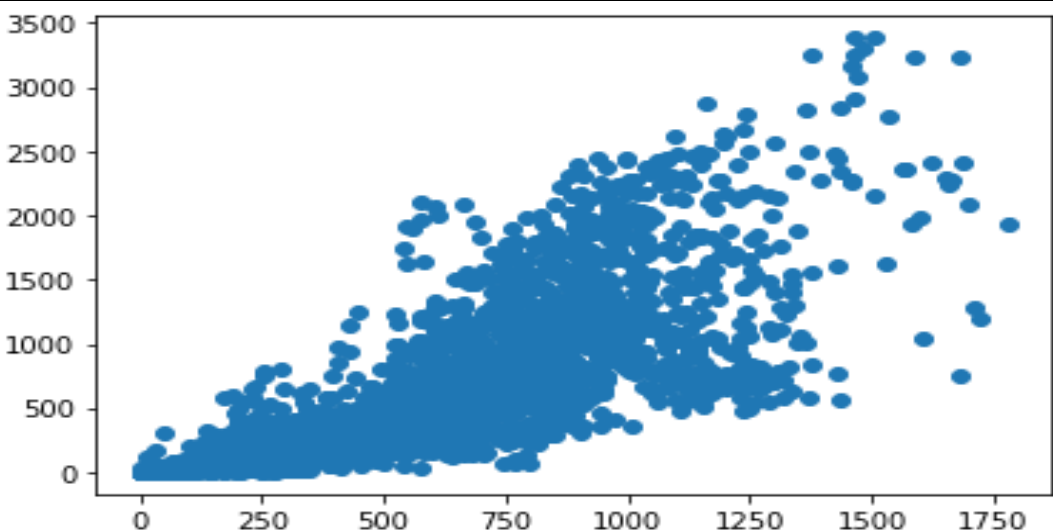
Y test shape : (2190,1)

Below mentioned are the visualizations and evaluation scores calculated for the models implemented,

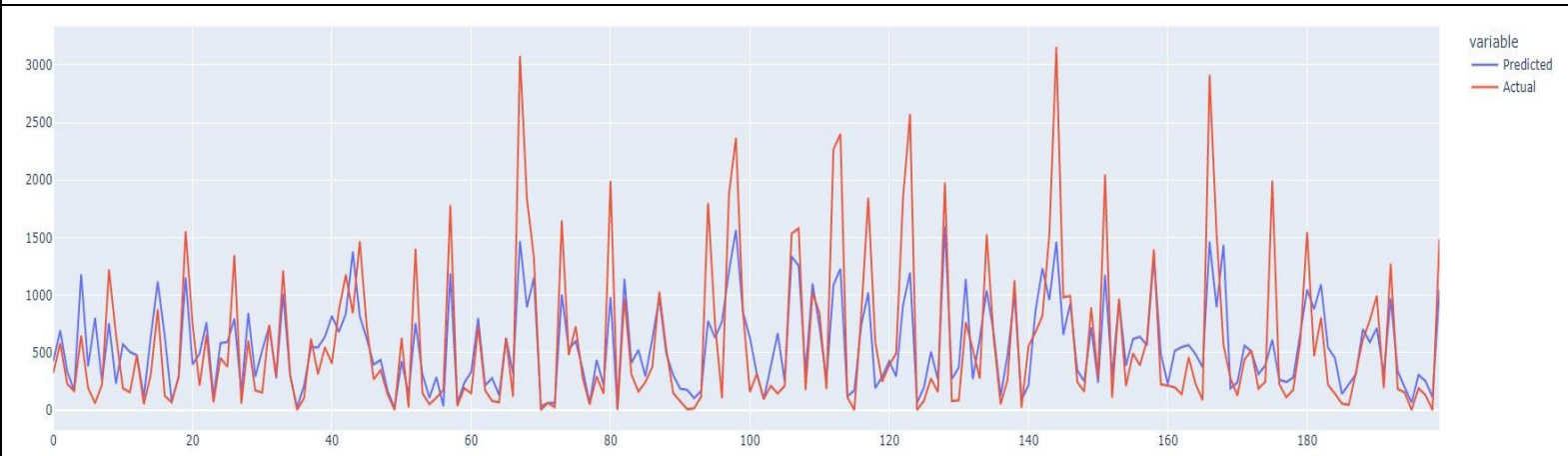


- The performance of linear regression is satisfactory and achieved an R2 score of 0.789 on the testing dataset

LASSO REGULARIZED REGRESSION

MSE Train	190750.15	<p>Y test actual vs Y test predicted</p> 
MSE Test	187901.42	
RMSE Train	436.75	
RMSE Test	433.48	
MAE Train	292.63	
MAE Test	292.48	
R2 Train	0.54	
R2 Test	0.55	
Adjusted R2 Train	0.53	
Adjusted R2 Test	0.54	

Predicted vs actual value of test data for the first 200 values

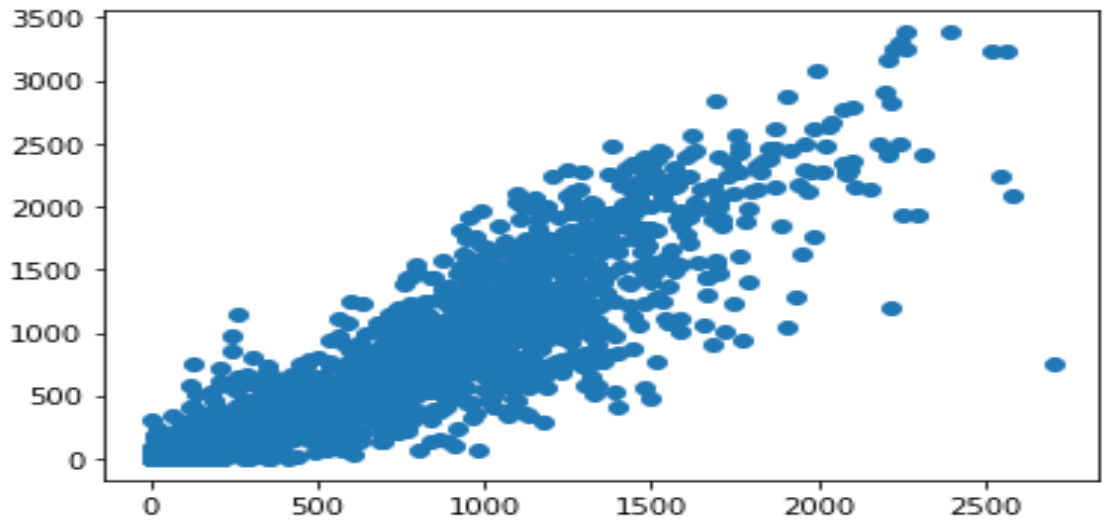


- The performance of linear regression is unsatisfactory and achieved an R2 score of 0.54 on the testing dataset

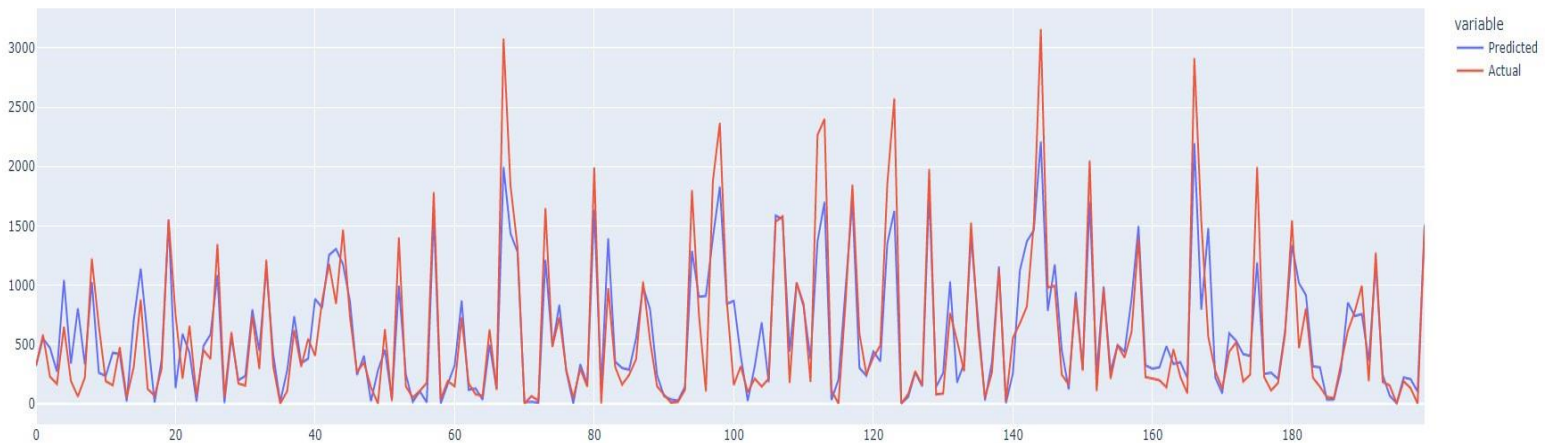
LASSO REGULARIZED REGRESSION AFTER CROSS-VALIDATION

MSE Train	87598.80
MSE Test	86440.61
RMSE Train	295.97
RMSE Test	294.01
MAE Train	201.88
MAE Test	200.85
R2 Train	0.79
R2 Test	0.79
Adjusted R2 Train	0.78
Adjusted R2 Test	0.79

Y test actual vs Y test predicted



Predicted vs actual value of test data for the first 200 values



- The performance of linear regression after cross-validation is satisfactory and achieved an R2 score of 0.79 on the testing dataset

RIDGE REGULARIZED REGRESSION

MSE Train 87603.63

MSE Test 86443.40

RMSE Train 295.98

RMSE Test 294.01

MAE Train 201.89

MAE Test 200.86

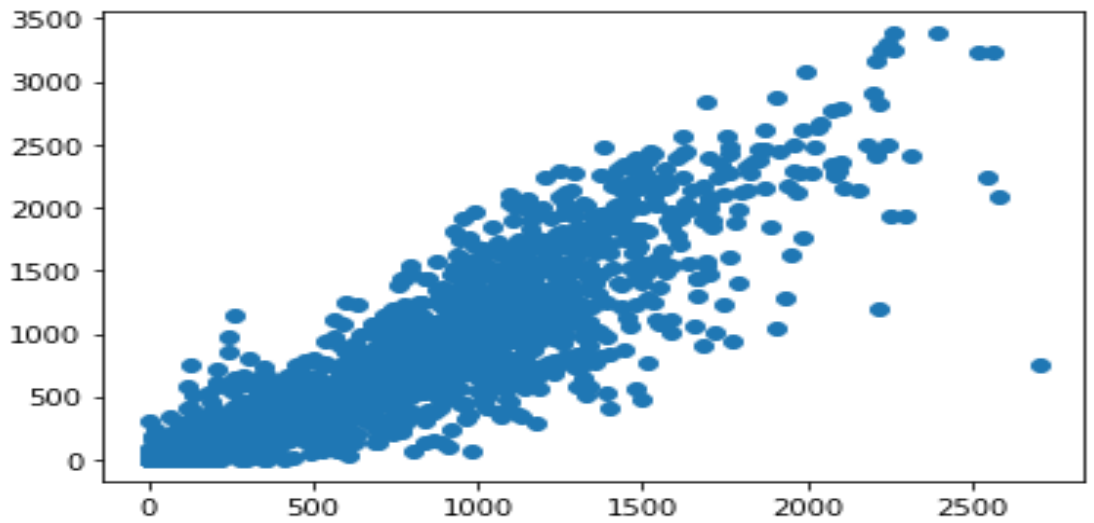
R2 Train 0.79

R2 Test 0.79

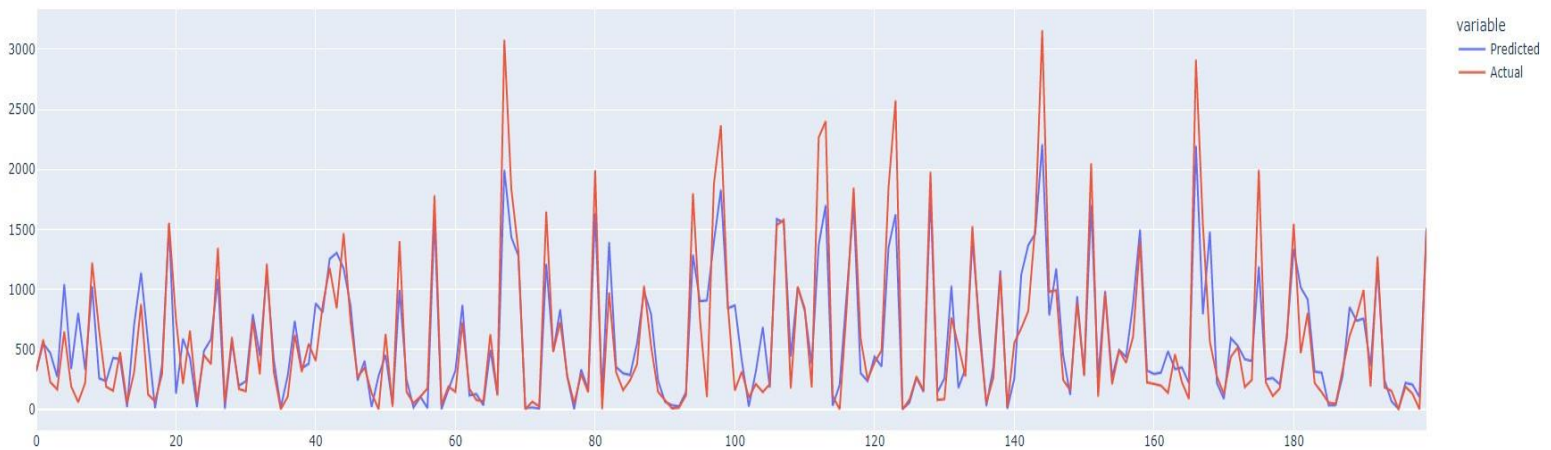
Adjusted R2 Train 0.78

Adjusted R2 Test 0.79

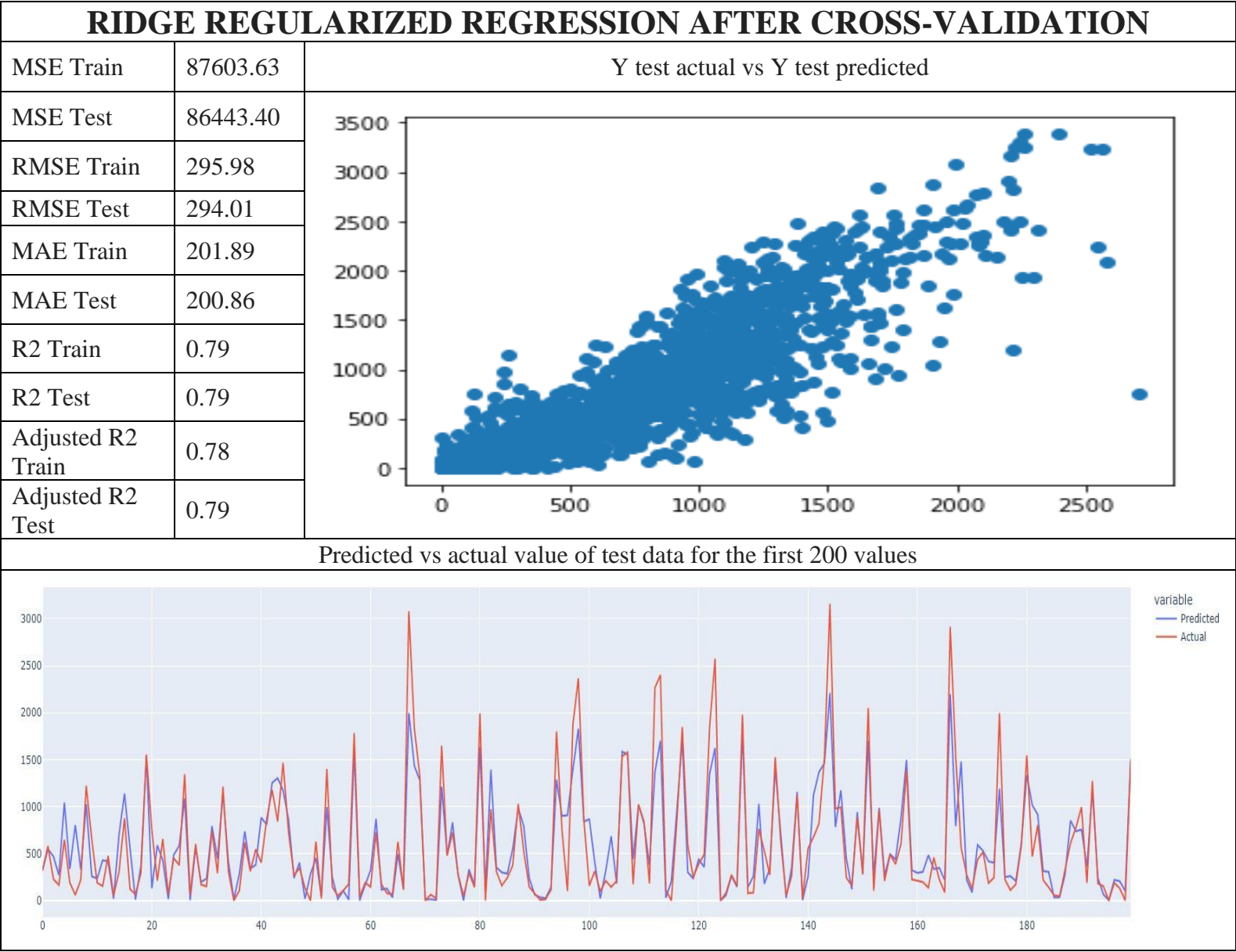
Y test actual vs Y test predicted



Predicted vs actual value of test data for the first 200 values



- The performance of ridge regularized regression is satisfactory and achieved an R2 score of 0.79 on the testing dataset

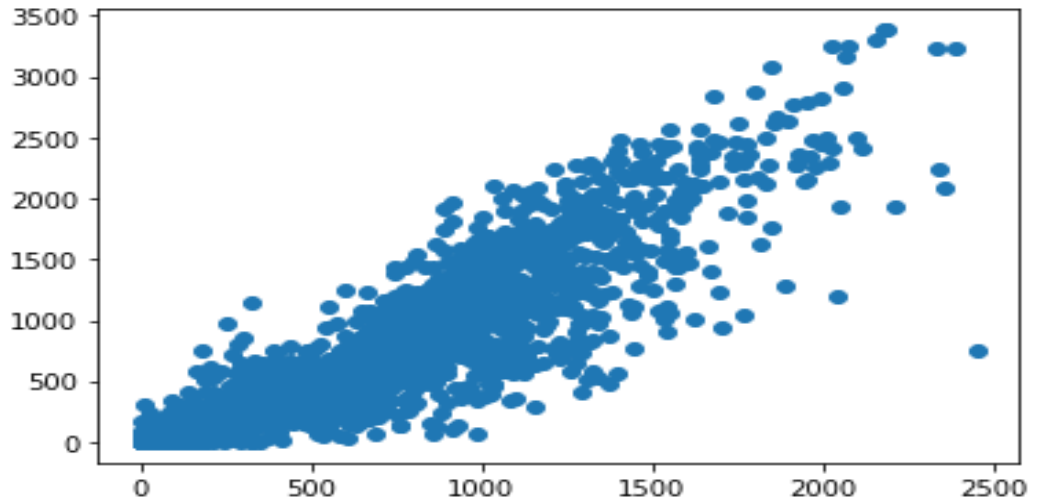


- The performance of ridge regularized regression after cross-validation is satisfactory and achieved an R2 score of 0.79 on the testing dataset

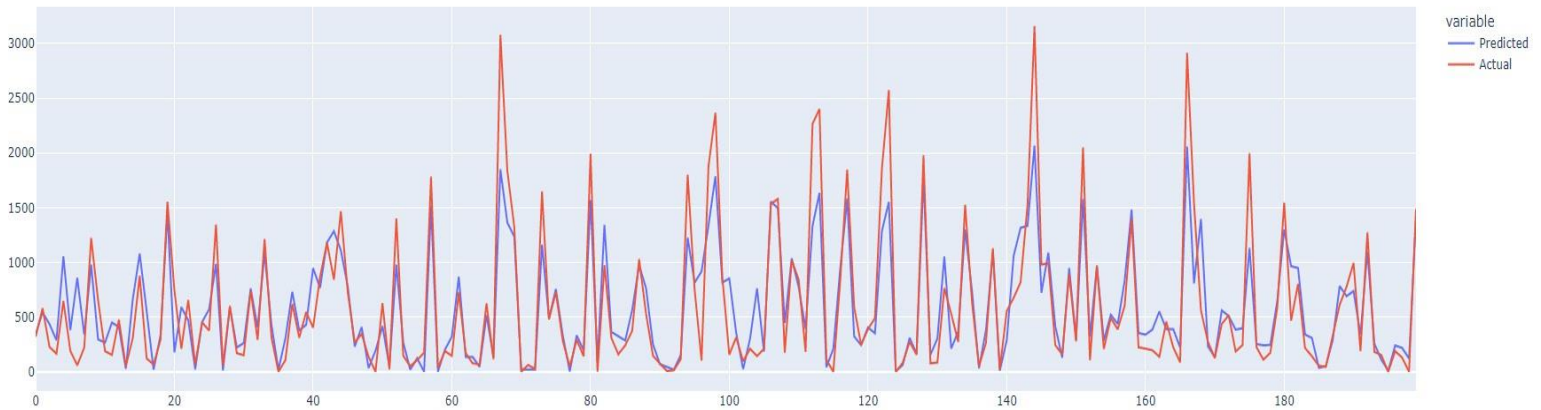
ELASTICNET REGULARIZED REGRESSION

MSE Train	95658.50
MSE Test	93621.75
RMSE Train	309.29
RMSE Test	305.98
MAE Train	209.94
MAE Test	208.68
R2 Train	0.77
R2 Test	0.78
Adjusted R2 Train	0.76
Adjusted R2 Test	0.77

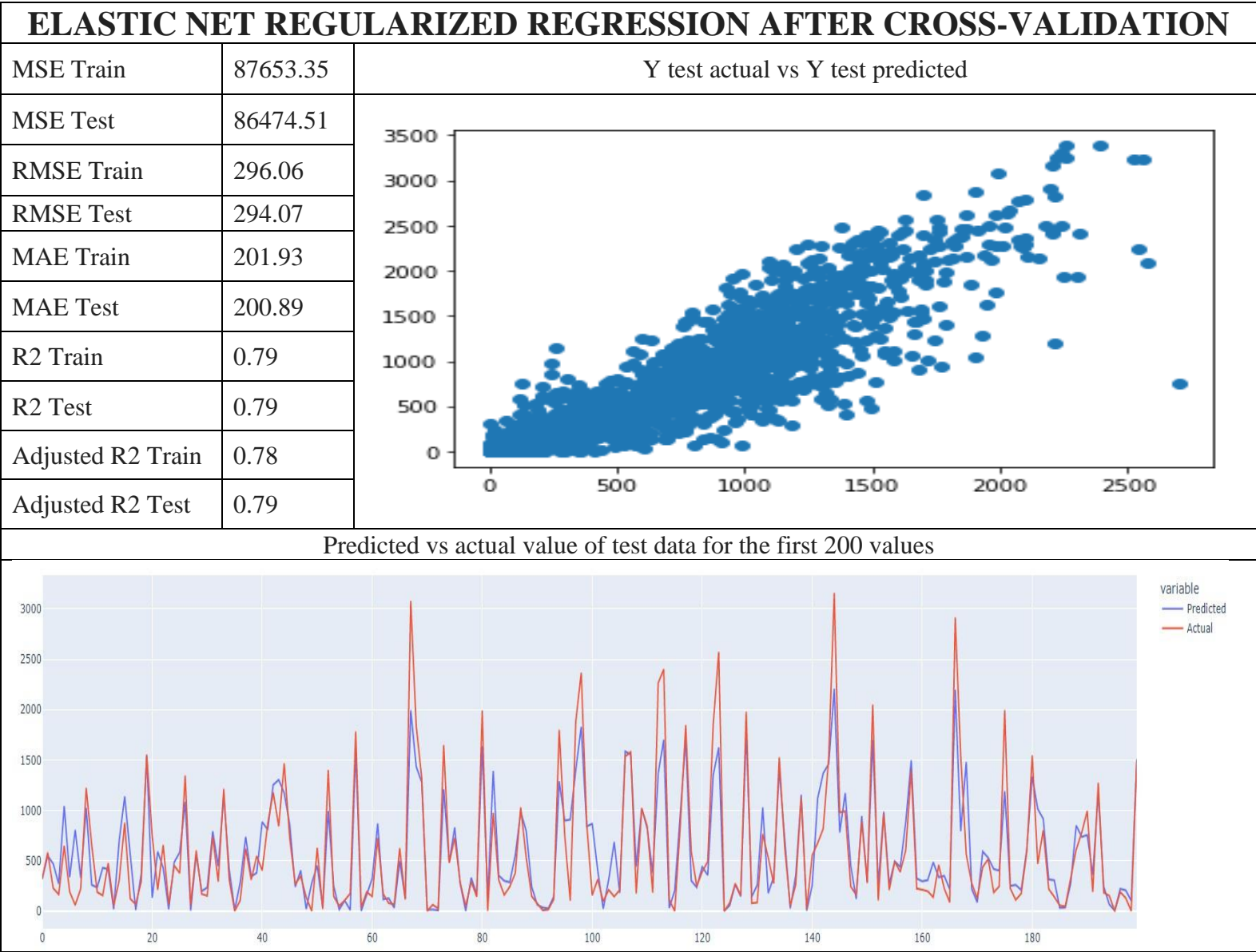
Y test actual vs Y test predicted



Predicted vs actual value of test data for the first 200 values

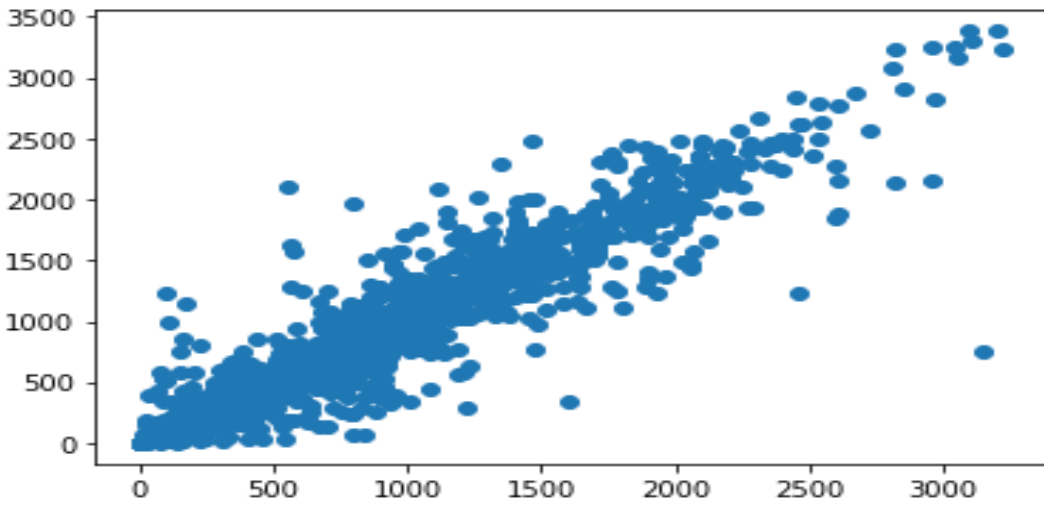


- The performance of elastic net regularized regression is satisfactory and achieved an R2 score of 0.78 on the testing dataset

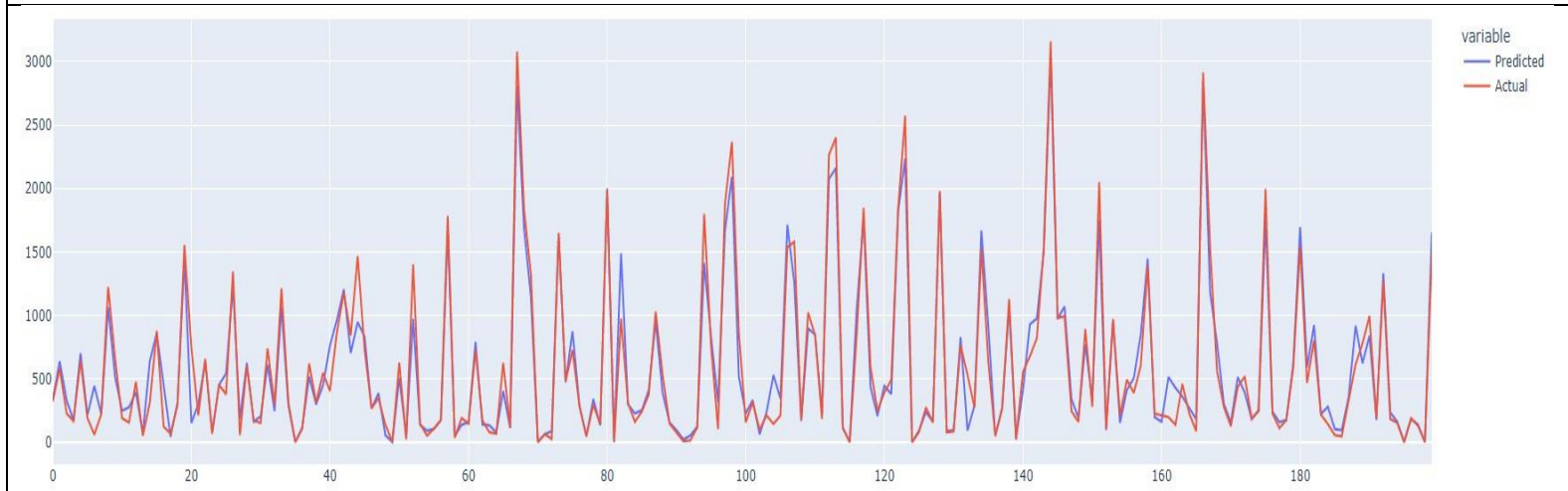


- The performance of elastic net regularized regression after cross-validation is satisfactory and achieved an R2 score of 0.79 on the testing dataset

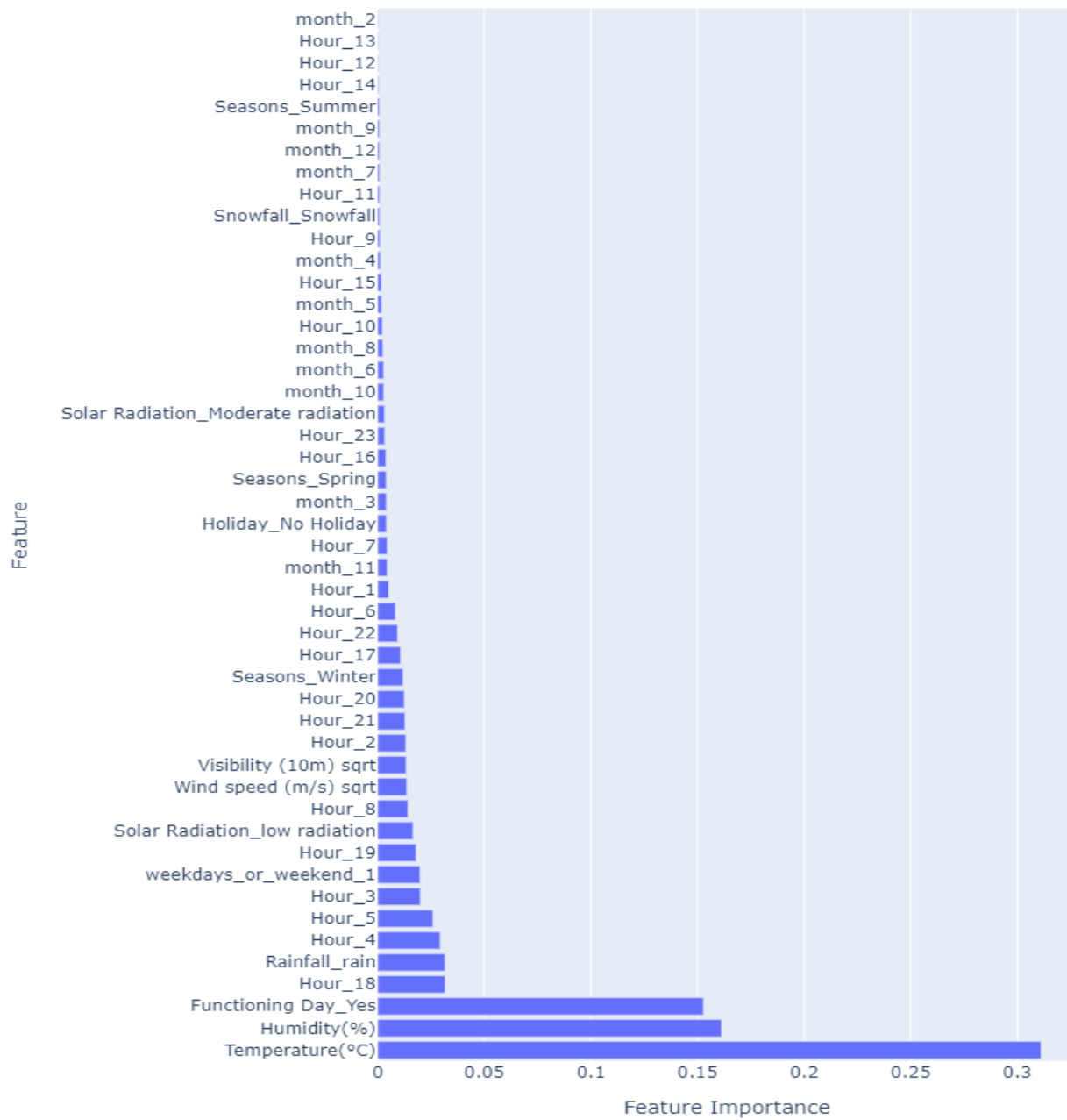
RANDOM FOREST REGRESSION

MSE Train	4600.54	<p style="text-align: center;">Y test actual vs Y test predicted</p> 
MSE Test	38668.66	
RMSE Train	67.83	
RMSE Test	196.64	
MAE Train	40.17	
MAE Test	113.14	
R2 Train	0.99	
R2 Test	0.91	
Adjusted R2 Train	0.99	
Adjusted R2 Test	0.91	

Predicted vs actual value of test data for the first 200 values



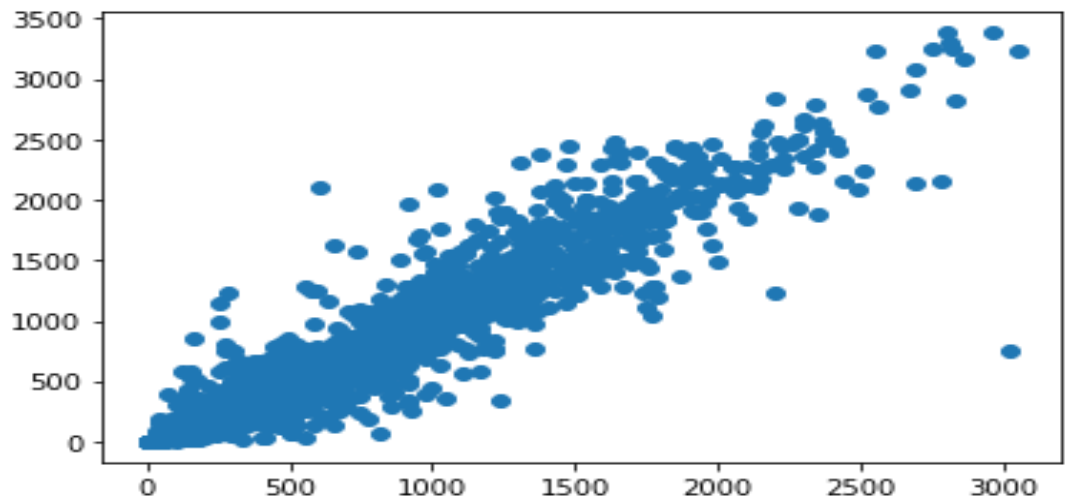
Feature importance plot



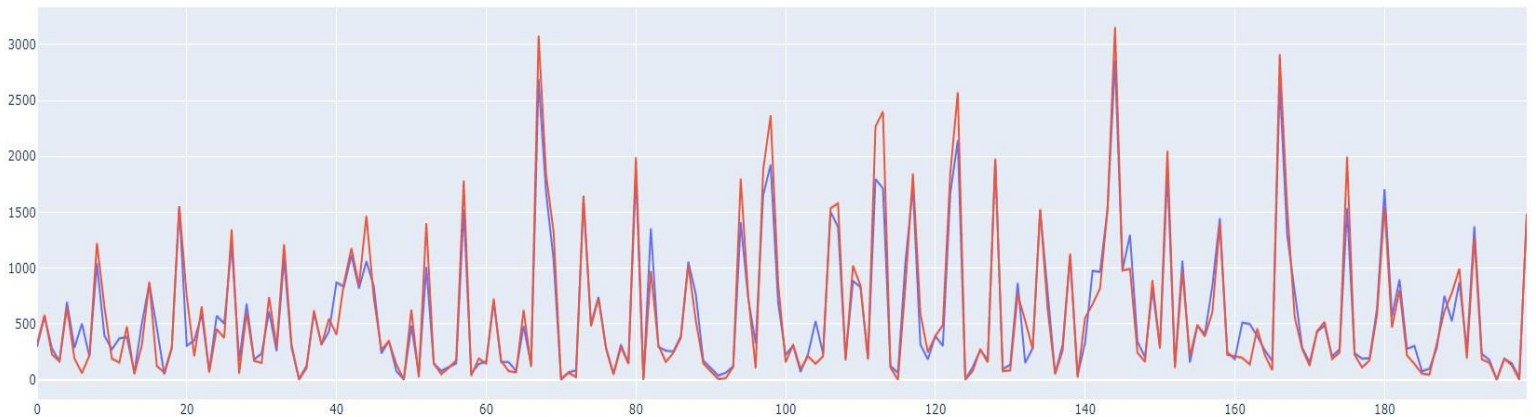
RANDOM FOREST REGRESSION AFTER CROSS-VALIDATION

MSE Train	12483.84
MSE Test	42541.30
RMSE Train	111.73
RMSE Test	206.26
MAE Train	68.49
MAE Test	124.99
R2 Train	0.97
R2 Test	0.90
Adjusted R2 Train	0.97
Adjusted R2 Test	0.90

Y test actual vs Y test predicted



Predicted vs actual value of test data for the first 200 values



- The performance of random forest regression is good and achieved an R2 score of 0.91 on the testing dataset

10. Conclusion:

Below mentioned is the summary of evaluation scores for all the models implemented,

Sl. No	Model	MAE train	MAE test	MSE train	MSE test	RMSE train	RMSE test	R2_score train	R2_score test	Adjusted R2 train	Adjusted R2 test
1	Linear regression	201.9	200.88	87665.19	86495.27	296.08	294.1	0.79	0.79	0.78	0.79
2	lasso regression	292.6	292.48	190750.2	187901.4	436.75	433.48	0.54	0.55	0.53	0.54
3	Lasso regression after cross-validation	201.9	200.85	87598.8	86440.61	295.97	294.01	0.79	0.79	0.78	0.79
4	Ridge regression	201.9	200.86	87603.63	86443.4	295.98	294.01	0.79	0.79	0.78	0.79
5	Ridge regression after cross-validation	201.9	200.86	87603.63	86443.4	295.98	294.01	0.79	0.79	0.78	0.79
6	Elastic net regression	209.9	208.68	95658.5	93621.75	309.29	305.98	0.77	0.78	0.76	0.77
7	Elastic net regression after cross-validation	201.9	200.89	87653.35	86474.51	296.06	294.07	0.79	0.79	0.78	0.79
8	Random forest regression	40.69	113.5	4634.89	38772.84	68.08	196.91	0.99	0.91	0.99	0.9
9	Random forest regression after cross-validation	52.82	123.78	7366.33	41590.43	85.83	203.94	0.98	0.9	0.98	0.9

With the data filtered for multicollinearity and features trimmed down, we ran several regression models on the features with test-train split of 0.25, of which Random Forest performed the best with an R2 of 0.91 on the test dataset.