Capstone Project-2

Bike Sharing Demand Prediction

(Supervised machine learning algorithm)

By

Nikhil Khushal Nimje

Introduction

AI

- Bike sharing is an increasingly popular part of urban transportation systems. Accurate demand prediction is the key to support timely re-balancing and ensure service efficiency. Most existing models of bike-sharing demand prediction are solely based on its own historical demand variation, essentially regarding bike sharing as a closed system and neglecting the interaction between different transport modes. This is particularly important because bike sharing is often used to complement travel through other modes (e.g., public transit). Despite some recent efforts, there is no existing method capable of leveraging spatiotemporal information from multiple modes with heterogeneous spatial units. To address this research gap, this study proposes a graph-based Machine learning approach for bike sharing demand prediction (B-MRGNN) with multimodal historical data as input. The spatial dependencies across modes are encoded with multiple intra- and inter-modal graphs. A multi-relational graph neural network (MRGNN) is introduced to capture correlations between spatial units across modes, such as bike sharing stations, subway stations, or ride-hailing zones.

❖ Points to be Discuss

- Business Understanding
- Data summary
- Data Collection
- Data Wrangling
- Feature Engineering
- EDA
- Model Building
- Conclusion on Bike Sharing Demand Prediction
- Impliction To Business

Business Understanding

- Bike rentals have became a popular service in recent years and it seems people are using it more often. With relatively cheaper rates and ease of pick up and drop at own convenience is what making this business thrive.
- Mostly used by people having no personal vehicles and also to avoid congested public transport which that’s why they prefer rental bikes.
- Therefore, the business to strive and profile more, it has to be always ready and supply no. of bikes at different locations to fulfil the demand
- Our projects goal is a pre planned set of bike count values that can be a handy solution to meet all demands.

Data summary

- We hasd a seoul Bike Data for our analysis and model building
- This dataset contains 8760 lines and 14 columns .
- This dataset contain weather information (Temperature, Humidity, Wind Speed, Visibility, Dew Points, Solar Radiation, Snowfall, Rainfall.) The no. of bikes rented per hour and date information.
- This Dataset contain Three categorical features ‘Seasons’, ‘Holiday’ & ‘Functioning Day’.

Data collections

- Date : year-month-day
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of he day
- Temperature-Temperature in Celsius
- Humidity - %
- Windspeed - m/s
- Visibility - 10m
- Dew point temperature - Celsius

Data Collection

- 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 wrangling

- **Categorical Features** : Seasons, Holiday & Functioning day.

- **Numerical columns** : Seasons, Humidity, Wind Speed, Visibility, Dew point, Temperature, Solar Radiation, Rainfall, Snowfall, Rental Bike Count.

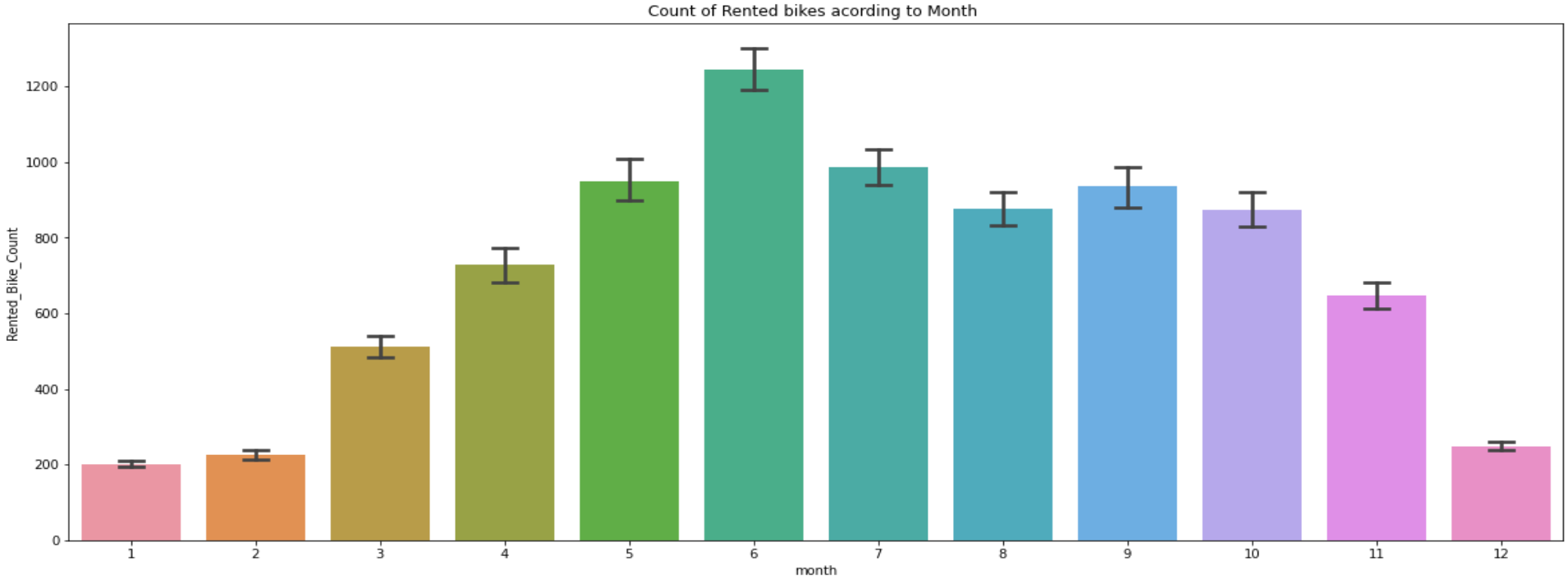
- **Rename Columns** : we renamed columns because they had units mentioned in brackets and was difficult to copy the feature name while working.

Feature Engineering

- There are no null values.
- The dataset show Hourly rental data for one year(1 December 2017 to 31 November 2018)(365 days). We consider this is a single year data.
- So now we convert the “date” column into 3 different column i.e “year”, “Month”, “Day”.
- There re no missing values present
- There are no duplicate values present
- And finally we have ‘rented bike count’ varible which we need to predict for new observations
- We change the name of some features for our convenience , they are as below ‘rented_bike_count’, ‘Hour’, ‘Temperature’, ‘Humidity’, ‘Wind speed’, ‘Visibility’, ‘Dew_point_temperature’, ‘solar_radiation’, ‘Rainfall’, ‘Snowfall’, ‘Seasons’, ‘Holiday’, ‘Functioning_day’, ‘Month’, ‘weekdays_weekend’

EDA

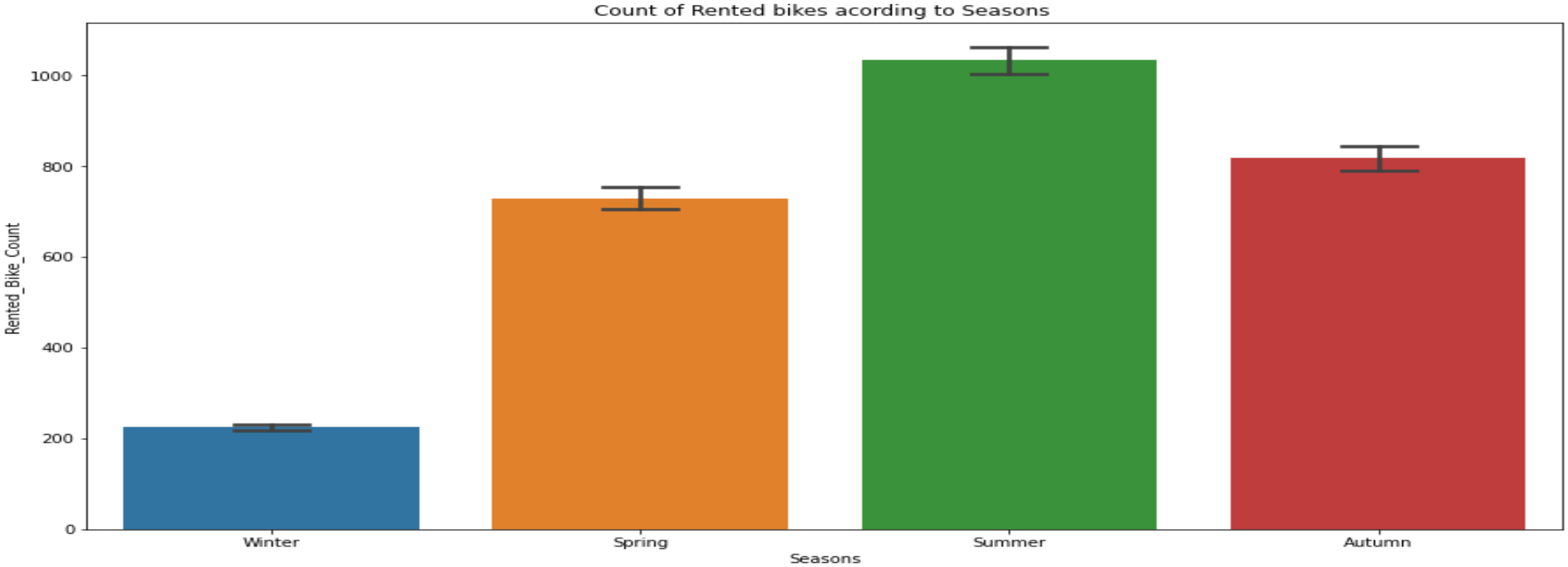
Analysis of Month Variable



- From the above bar plot we can clearly say that from the month 5 to 10 the demand of the rented bike is high as compared to other months. These months are comes inside the summer seasons

EDA

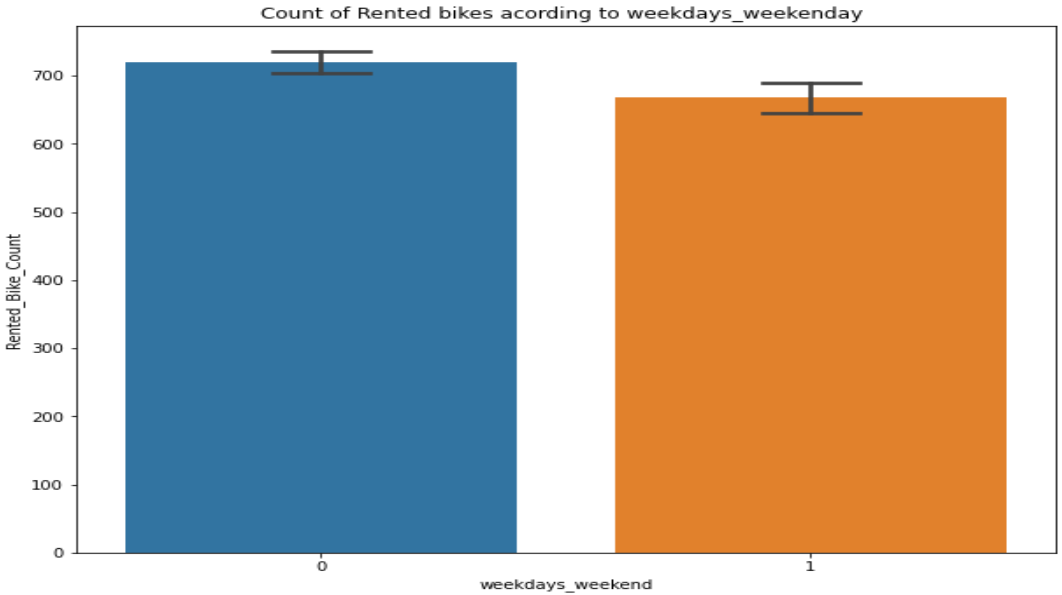
Analysis of Seasons variable



- summer seasons had the highest Bike Rent Count. People are more likely to take rented bikes in summer. Bike rentals in winter is very less compared to other seasons.

EDA

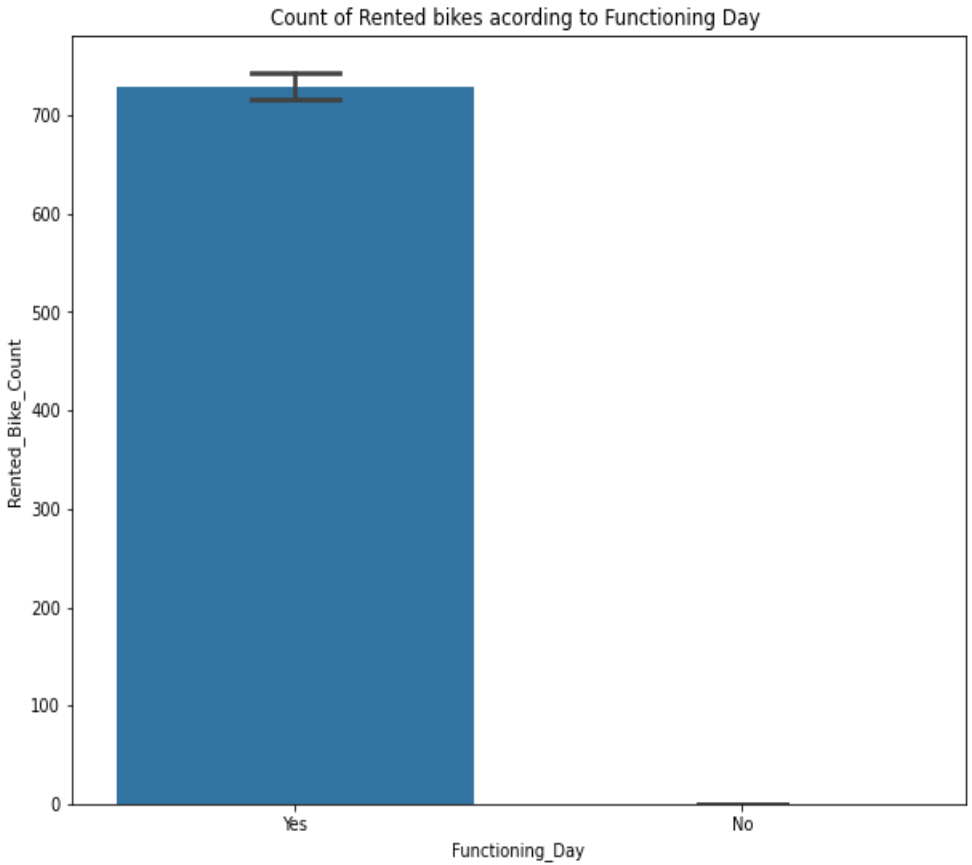
Analysis of weekdays_weekend variable



- From the above point plot and bar plot we can say that in the weekdays which represent in blue colour show that the demand of the bike higher because of the office.
- The orange colour represent the weekend days, and it show that the demand of rented bikes are very low especially in the morning .

EDA

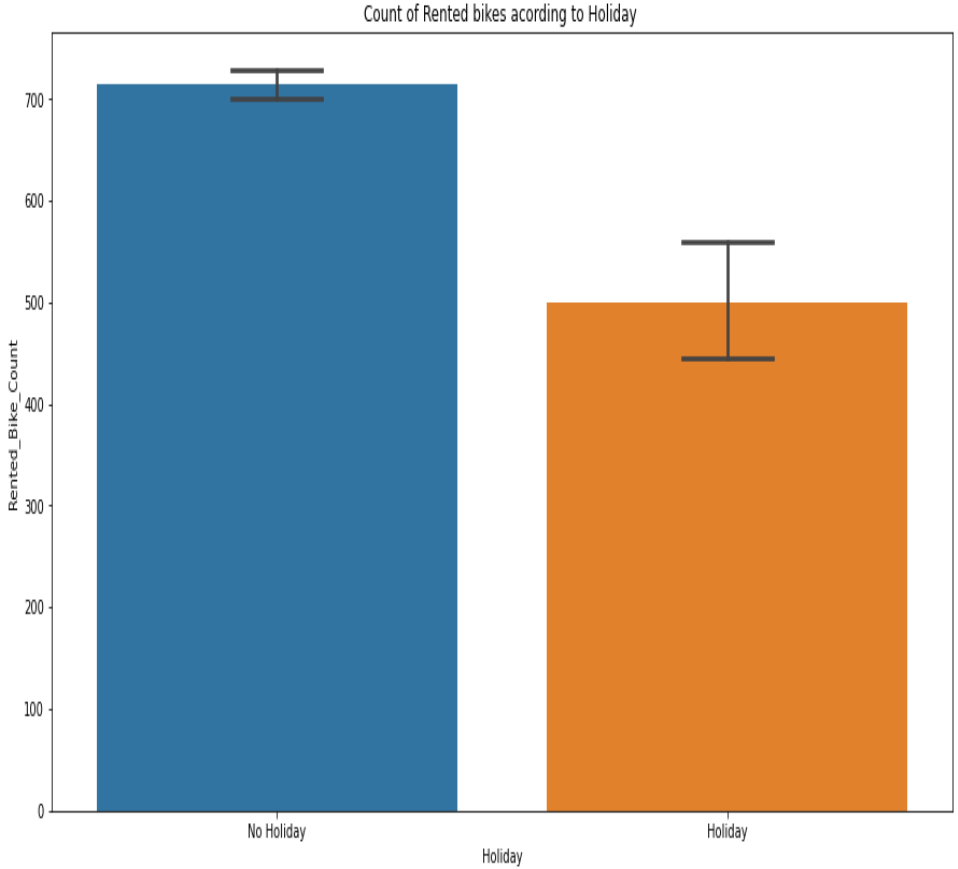
Analysis of Functioning Day variable



- In the above table point plot which shows the use of rented bike in functioning day or not, and it clearly shows that.
- People don’t use rented bikes in no functional day

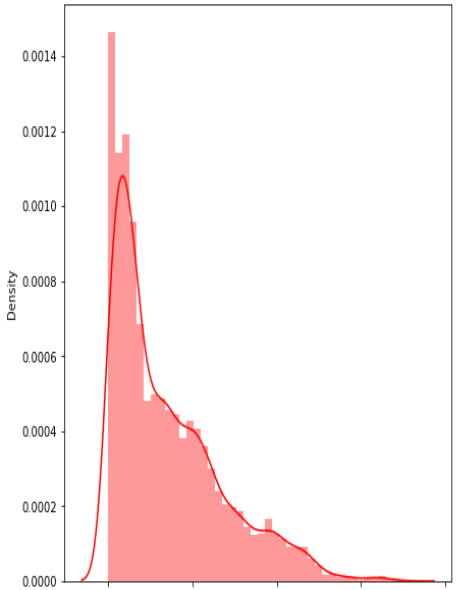
EDA

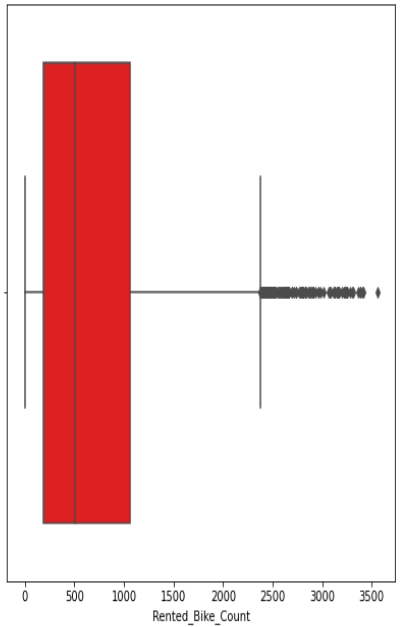
Analysis of Holiday Day variable

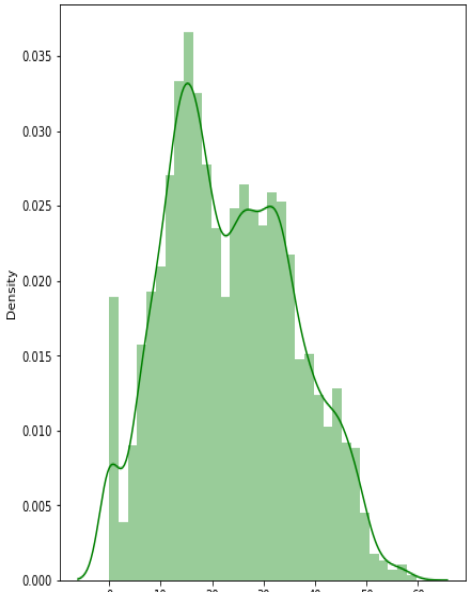


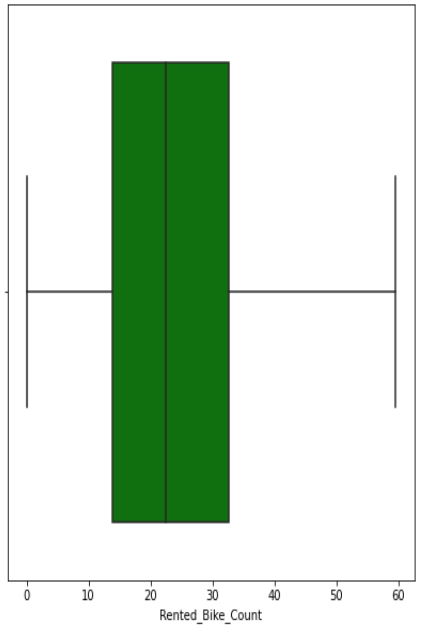
- High no of Bikes are rented in no holidays which is almost 700.
- Holiday there are approximately 500 bikes were rented.

Let's check distribution of target variable- "Bike Rented Count"









Distribution is rightly skewed and some outliers are observed

To normalise the distribution we applied square root method. After normalisation no outliers were founds.

Model Building

- Linear regression
- Lasso regression (regularized regression)
- Ridge regression (regularized regression)
- Elastic net
- Descision tree regression
- Random forest regression
- Gradient boosting regression

Linear Regression

Train set results

MSE : 59.97909536584496

RMSE : 7.744617186526715

MAE : 5.843650059667873

R2 : 0.6144125475019626

Adjusted R2 :0.6124949358407252

Test set results

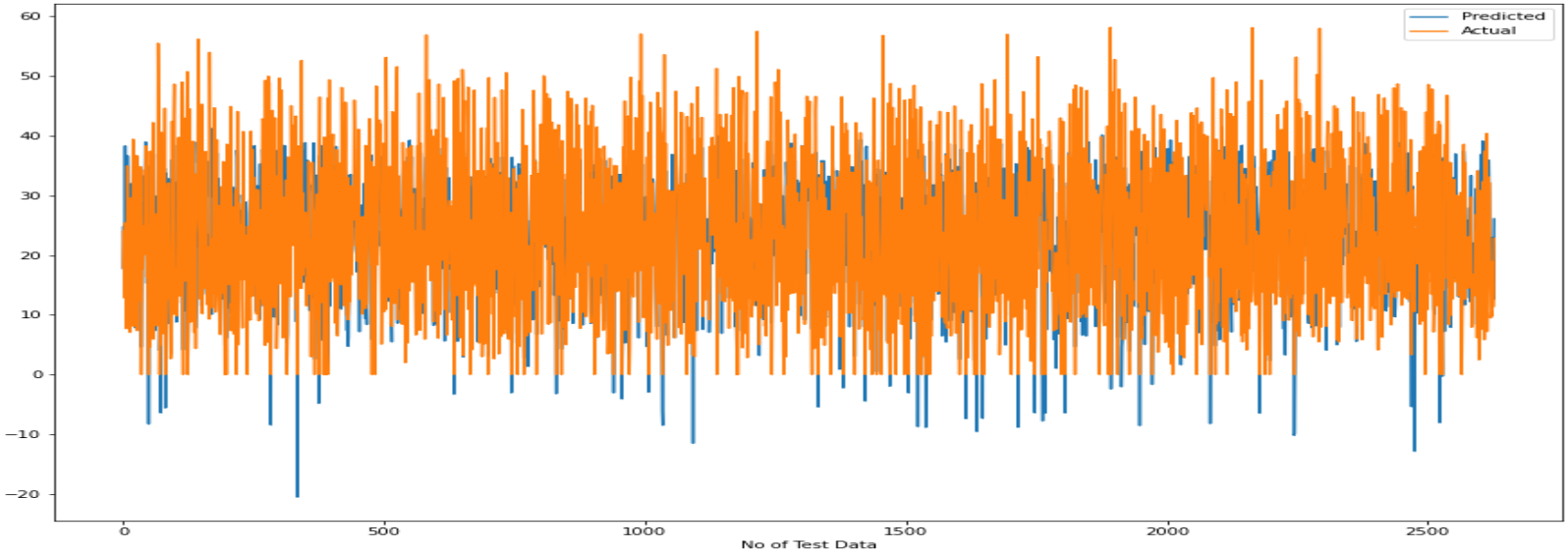
MSE : 58.0947481788344

RMSE : 7.621991090183352

MAE : 5.80569636155266

R2 : 0.621774269728069

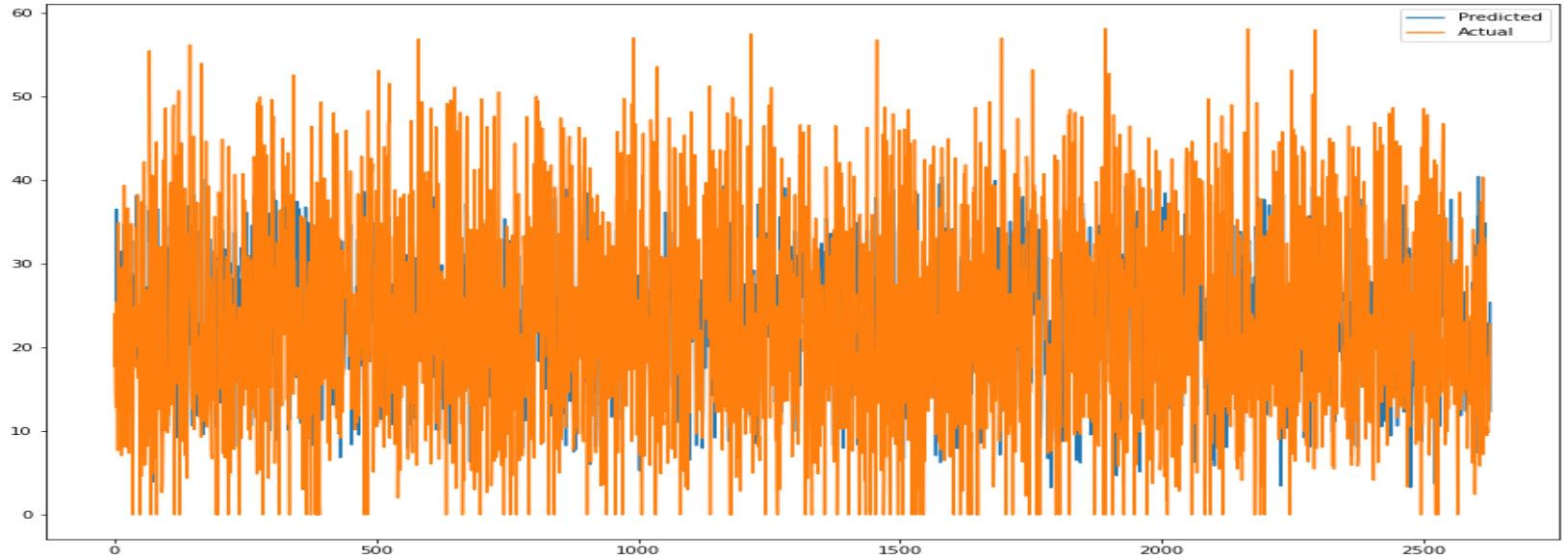
Adjusted R2 :0.6198932695392645



Lasso Regression

- Train set results
- MSE : 88.2234542654811
- RMSE : 9.392734120876684
- MAE : 7.040000965970058
- R2 : 0.4328381117902722
- Adjusted R2:0.43001749031103487

- Test set results
- MSE : 89.53739063019795
- RMSE : 9.462419914070498
- MAE : 7.112289785520384
- R2 : 0.41706701518868294
- Adjusted R2:0.41416796055878735



Ridge Regression

Train set results

MSE : 59.979104007440746

RMSE : 7.7446177444365025

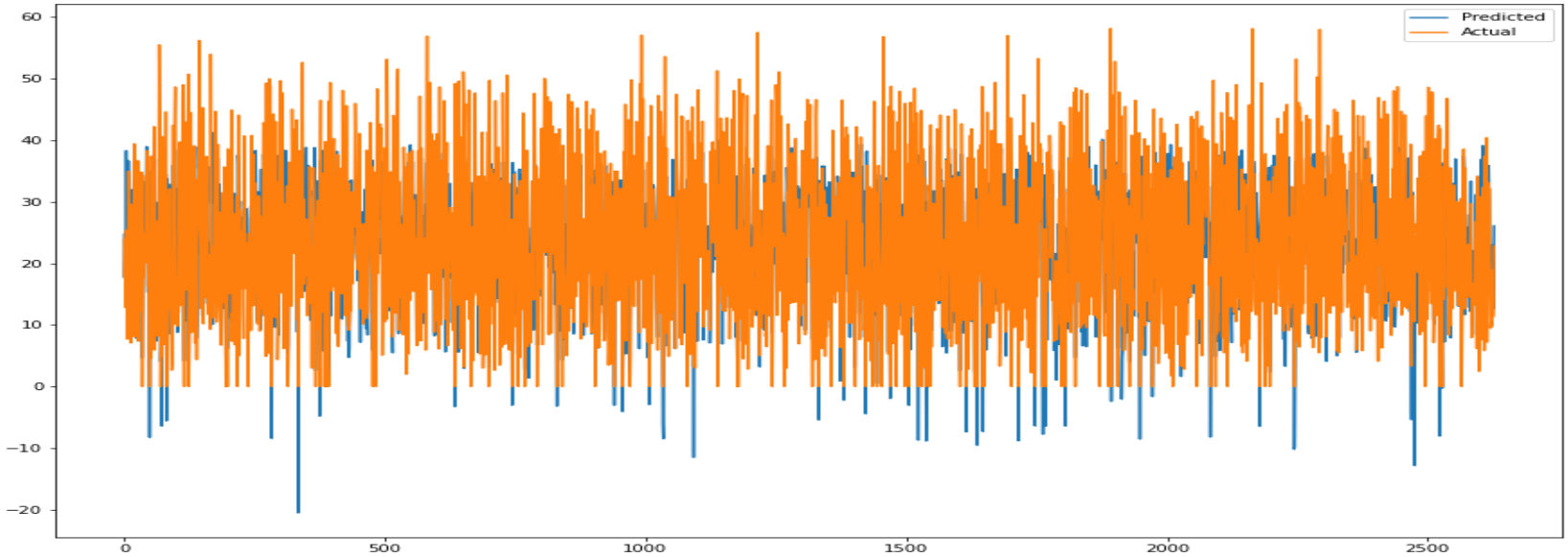
MAE : 5.843670594238321

R2 : 0.6144124919477586

Adjusted R2 :0.6124948800102379

Test set results

MSE : 58.09540824137624 RMSE : 7.622034389936603 MAE : 5.805807674500229 R2 : 0.6217699723923419 Adjusted R2 : 0.6198889508319365



Elastic Net

Train set results

MSE : 71.22370297072081

RMSE : 8.439413662732786

MAE : 6.382544656919715

R2 : 0.5421243681911899

Adjusted R2 : 0.5398472514300903

Test set results

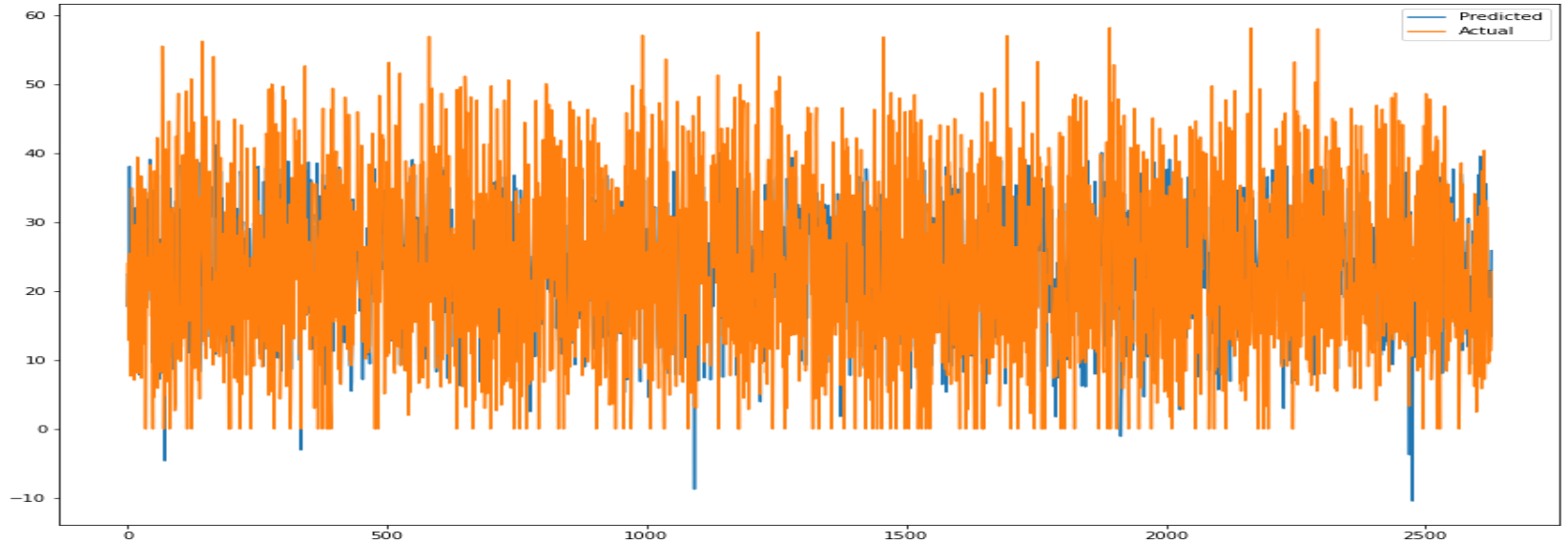
MSE : 71.04211892272907

RMSE : 8.428648700873056

MAE : 6.410681282705167

R2 : 0.5374804409703239

Adjusted R2 :0.5351802289323033



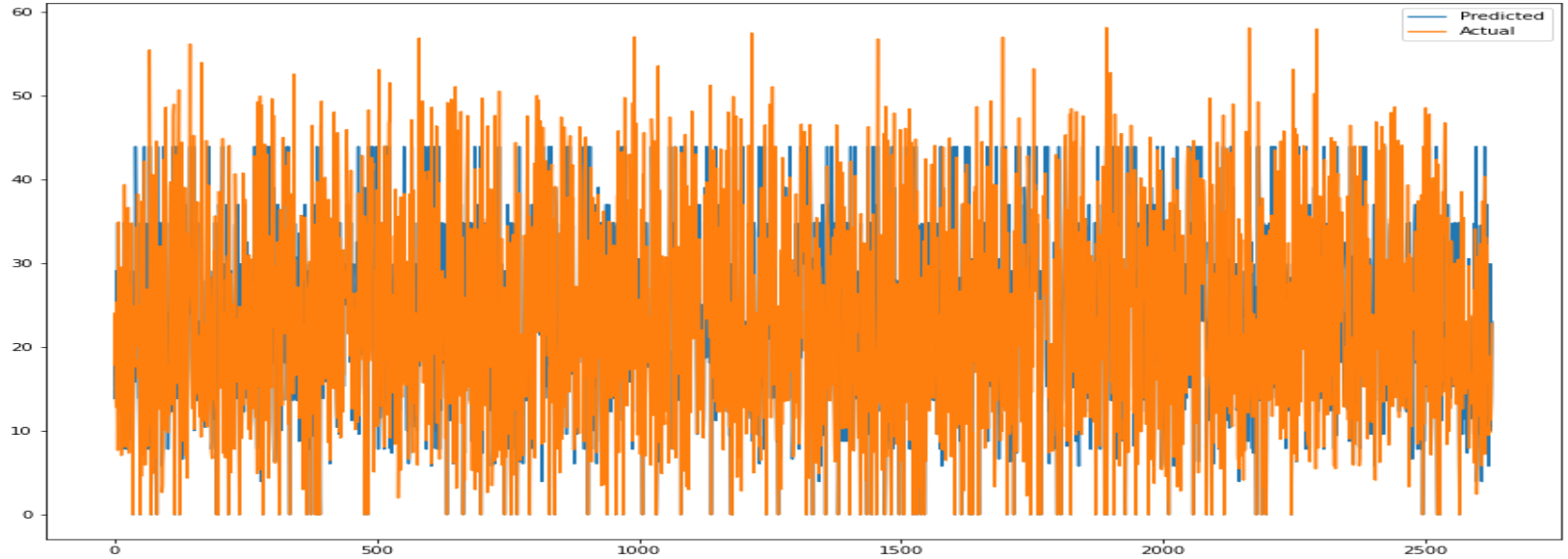
Decision Tree

Train set results

Model Score: 0.8575477039341736
MSE : 23.61568460086433
RMSE : 4.859597164463771
MAE : 3.285413559331172
R2 : 0.8481819105689653
Adjusted R2 : 0.8474268856406548

Test set results

MSE : 26.798968158663417
RMSE : 5.1767719824871
MAE : 3.484817714775623
R2 : 0.8255253767321165
Adjusted R2 : 0.8246576758512892



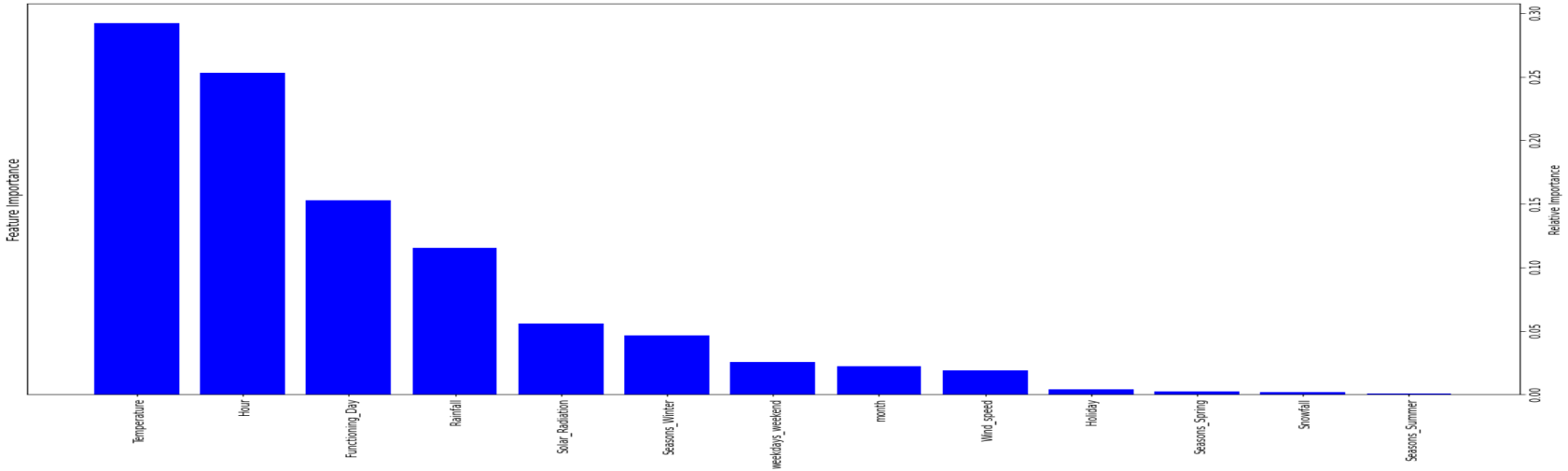
Random Forest

Train set results

Model Score: 0.9884067841901815
MSE : 1.803353797300936
RMSE : 1.3428900912959838
MAE : 0.806170423126537
R2 : 0.9884067841901815
Adjusted R2 : 0.9883491285645014

Test set results

MSE : 11.437769980681082
RMSE : 3.3819772294740664
MAE : 2.1429498810699403
R2 : 0.9255344236916483
Adjusted R2 : 0.9251640899150574



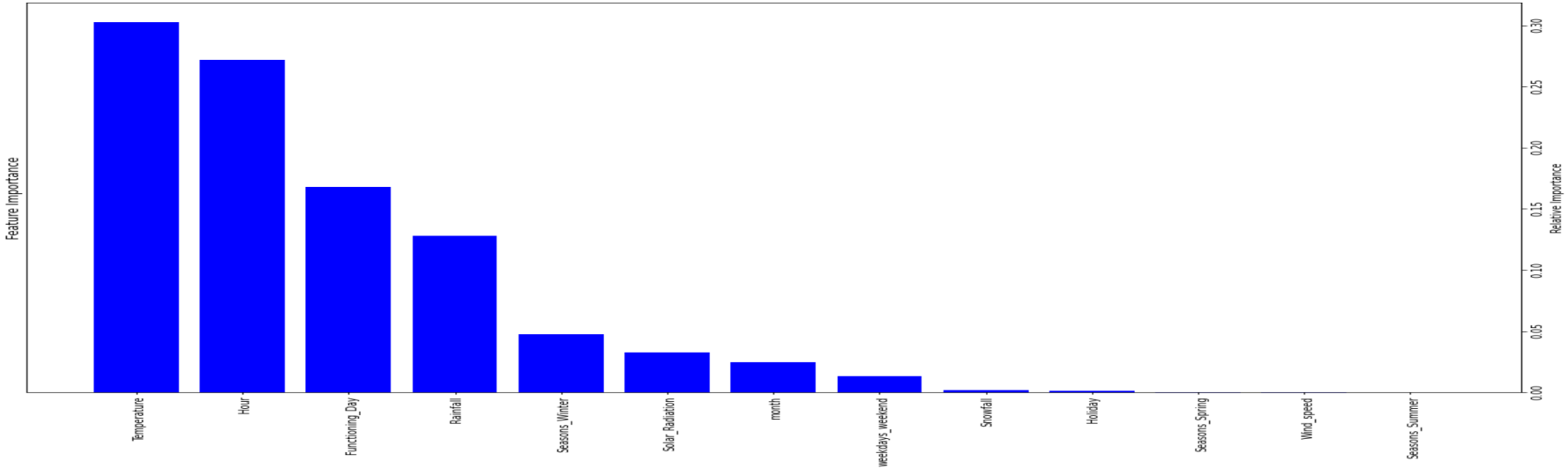
Gradient Boosting

Train set results

Model Score: 0.8781525188962888
MSE : 18.95368130332168
RMSE : 4.3535825825774435
MAE : 2.9899904076180146
R2 : 0.8781525188962888
Adjusted R2 : 0.877546544430203

Test set results

MSE : 20.095278074316163
RMSE : 4.482775710909053
MAE : 3.1099064703241317
R2 : 0.8691697362853041
Adjusted R2 : 0.8685190884550474



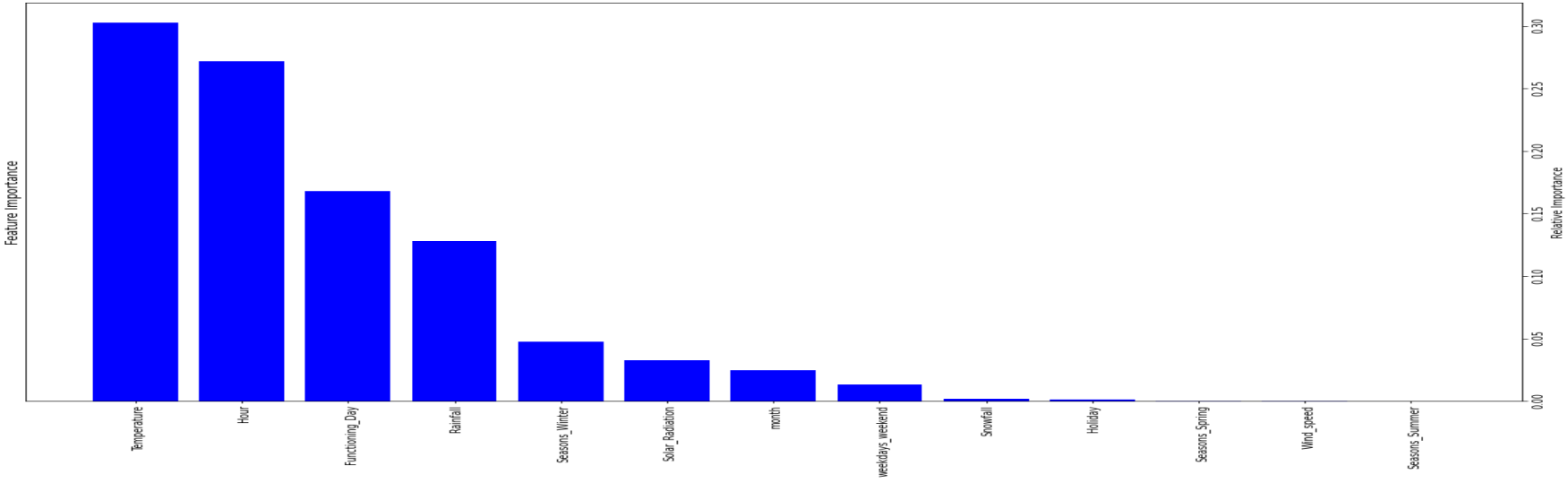
Gradient Boosting Regressor With Gridsearchcv

Train set results

Model Score: 0.9499181379354303
MSE : 7.790359259382813
RMSE : 2.7911215056644907
MAE : 1.7467932774564698
R2 : 0.9499181379354303
Adjusted R2 : 0.9496690697614291

Test set results

MSE : 11.60398815204139
RMSE : 3.4064627037502393
MAE : 2.159700033789734
R2 : 0.9244522606525093
Adjusted R2 : 0.92407654503984



Conclusion on Bike Sharing Demand Prediction AI

		Model	MAE	MSE	RMSE	R2_score	Adjusted R2
Training set	0	Linear regression	5.844	59.979	7.745	0.614	0.61
	1	Lasso regression	7.040	88.223	9.393	0.433	0.43
	2	Ridge regression	5.844	59.979	7.745	0.614	0.61
	3	Elastic net regression	6.383	71.224	8.439	0.542	0.54
	4	Dicision tree regression	3.285	23.616	4.860	0.848	0.85
	5	Random forest regression	0.806	1.803	1.343	0.988	0.99
	6	Gradient boosting regression	2.990	18.954	4.354	0.878	0.88
	7	Gradient Boosting gridsearchcv	1.747	7.790	2.791	0.950	0.95
Test set	0	Linear regression	5.806	58.095	7.622	0.622	0.62
	1	Lasso regression	7.112	89.537	9.462	0.417	0.41
	2	Ridge regression	5.806	58.095	7.622	0.622	0.62
	3	Elastic net regression Test	6.411	71.042	8.429	0.537	0.54
	4	Dicision tree regression	3.485	26.799	5.177	0.826	0.82
	5	Random forest regression	2.143	11.438	3.382	0.926	0.93
	6	Gradient boosting regression	3.110	20.095	4.483	0.869	0.87
	7	Gradient Boosting gridsearchcv	2.160	11.604	3.406	0.924	0.92

Conclusion on Bike Sharing Demand Prediction

- As we have calculated MAE, MSE, RMSE, & R2 score for each model. Based on R2 score will decide our model performance.

- **Our Assumptions :-** If the difference of R2 score between Train data and test is more than 5% we will consider it as overfitting

- **Linear, Ridge :**
- Linear, Ridge have almost similar R2 scores (61%) on both training and test data. (Even after using GridsearchCV we have got similar results as of base models).
- **Lasso :**
- In lasso model, we got R2 score as 43% on training data and on test data we got 41%.

Conclusion on Bike Sharing Demand Prediction

- **Elastic Net :**
- On Elastic Net model we got R2 score 54% on both training data and test data.
- **Descision Tree :**
- Descision Tree Regression model we got R2 score 85% on training data and 82% on test data.
- **Random Forest :**
- On Random Forest Regression model we got R2 score 99% on training data and 93% on test data.
- **Gradient Boosting :**
- On Gradient Boosting Regression model we got R2 score 87% on both training and test data

Implication To Business

- When we compare the root mean squared error and mean absolute error of all the models, Random forest Regression and gradient boosting gridsearchCV gives the highest R2 score of 99% and 95% respectively for Train set and 92% for Test set. So, finelly this model is best for predicting the bike rental count on daily basis.

- Prediction model decay with respect to time, or we can`t trust any model on long go. Frequent retraining is one way to address production model maintenance. Sudden changes in data could lead model to behave unwantedly. Monitoring and observability adds one more layer to assure model quality. Data centric avail better reliability. Being cautious about drifts is a good approach or it helps to early detect the change in model behavior.


THANK
YOU!