Capstone Project-2

Seoul Bike Sharing Demand Prediction

(Supervised Machine Learning Regression)

By
Sweta Seal

Problem Statement

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern.

The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

The Final aim of our project is to predict the bike count on various affecting factors and build a model that helps for stable supply of bikes at required hours.

Workflow

Data collection and Description → Data Wrangling and feature description → Exploratory data analysis → Preparing the data for Modeling → Model selection, Execution and Evaluation → Conclusions

Data Description

AI

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

Attribute Information:

- **Date** : The date of the day, during 365 days from 01/12/2017 to 30/11/2018.
- **Rented Bike count** - Count of bikes rented at each hour.
- **Hour** - The hour of the day, starting from 0-23.
- **Temperature**-Temperature in Celsius.
- **Humidity** - Humidity in the air (%).
- **Wind Speed** - Speed of the wind (m/s).
- **Visibility** - Visibility in (m).
- **Dew point temperature** - Temperature at the beginning of the day (Celsius).
- **Solar radiation** - Sun contribution (MJ/m2).
- **Rainfall** - Amount of rain (mm).
- **Snowfall** - Amount of snowfall (cm).
- **Seasons** - Season of the year (Winter, Spring, Summer, Autumn).
- **Holiday** - If the day is a Holiday/No holiday.
- **Functional Day** - If the day is a Non Functioning day/Functional day.

Data Wrangling and feature description

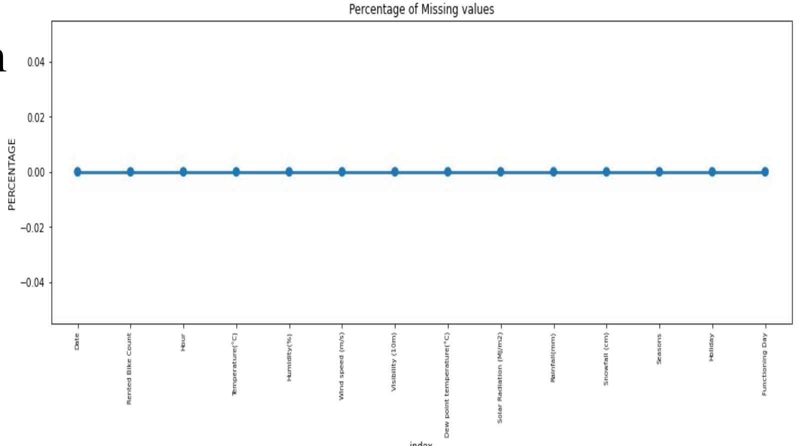
From the Seoul Bike data set given to us, we observed that we have **8760** observation with **14** feature attributes including the target variable.

Feature Classification: The feature attributes given to us are classified into 2 types

- **Categorical features:** Seasons, Holiday and Functioning day.
- **Numerical features:** Date, Hour, Rented bike count, Temperature, Humidity, Wind speed, Visibility, Dew point temperature, Solar radiation, Rainfall, Snowfall.

We tried to explore for any duplicate/missing values in the data set and we found that there are no duplicate/missing values present in the given data set.

The given plot explains that there are no missing values in the data set



Data Wrangling and feature description

AI

During our analysis we found that date attribute is given as ‘object’ type, so we changed the datatype of date attribute to datetime64.

```
# Changing The datatype of Date attribute to extract 'Month','Day',"Year". so that we can analyze the Bike rentals with respect to year, months and days.
bike_data['Date']=bike_data['Date'].astype('datetime64[ns]')
```

Later on we created two new attributes from the Date attribute namely ‘Month’ ‘Year’ & ‘Day’ which were used further for EDA. As our requirements were met by these newly formed attributes, we dropped the Date attribute from the dataset.

```
# Creating new attributes 'Month','Year','Day'.
bike_data['Month']=bike_data['Date'].dt.month
bike_data['Day']=bike_data['Date'].dt.day_name()
bike_data['year'] =bike_data['Date'].dt.year
```

```
# Droping 'Day', 'Date', 'Year' attributes.
bike_data.drop(['Date',"Day",'year'], axis=1 ,inplace=True)
```

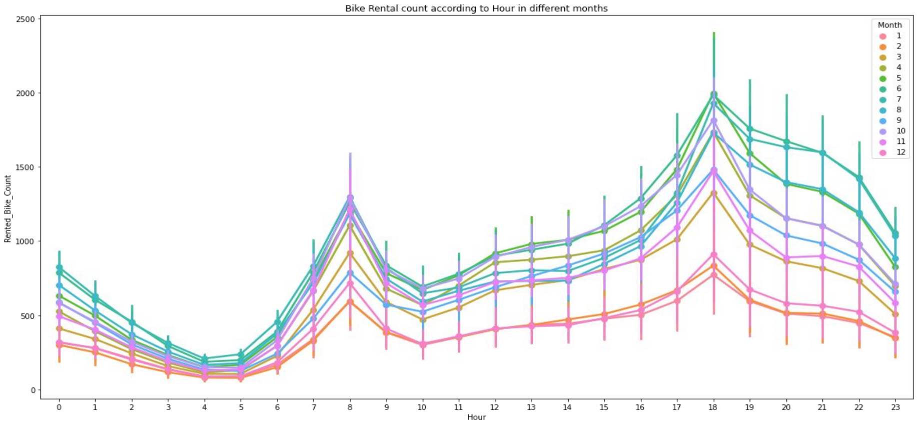
Exploratory Data Analysis(EDA)

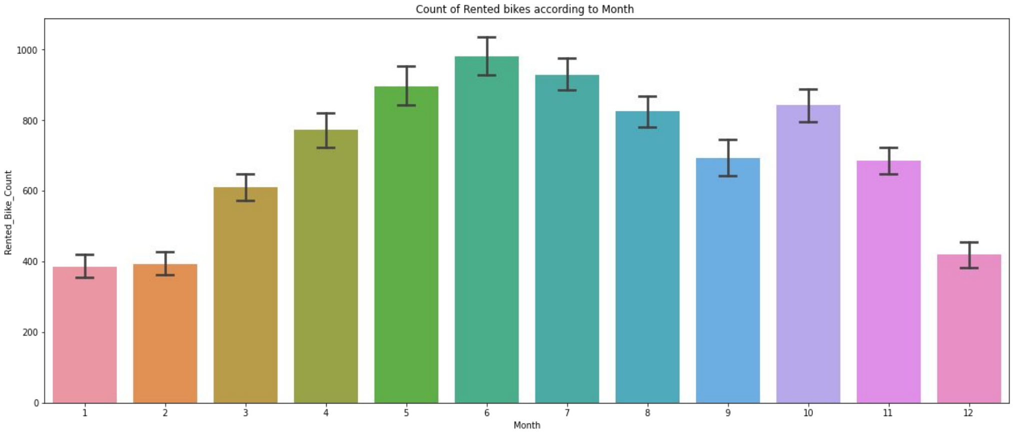
Analysis on Categorical Variables:

Bike Rental Count with respect to Hours on Months:

From the bar plot we can say that from the month 5 to 10 the demand of rented bike is high as compared to other months and these months comes under the summer season.

Rented bike count is higher in the 6th month from the below bar plot.

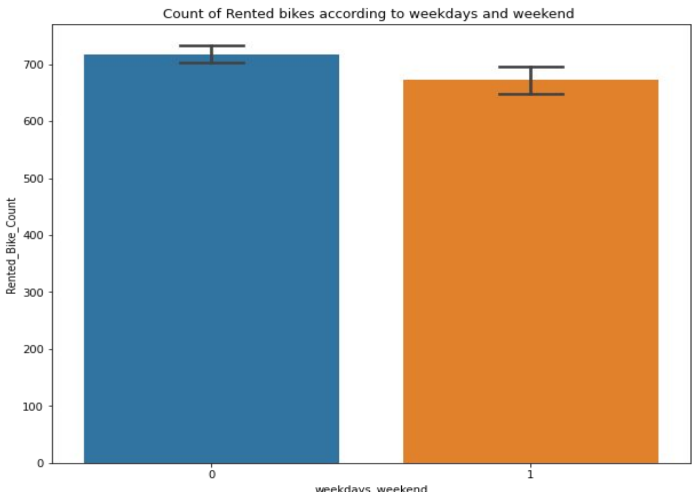




- From the point plot we can say that, there is sudden peak between 6/7 am to 10 am.
- Again we can see a peak between 5pm to 8pm. This may be due to office/college leaving time for the above people.
- We can also say that , from morning 7 AM to Evening 8 PM we have good Bike Rent Count. and from 7PM to 7AM Bike Rent count starts declining.

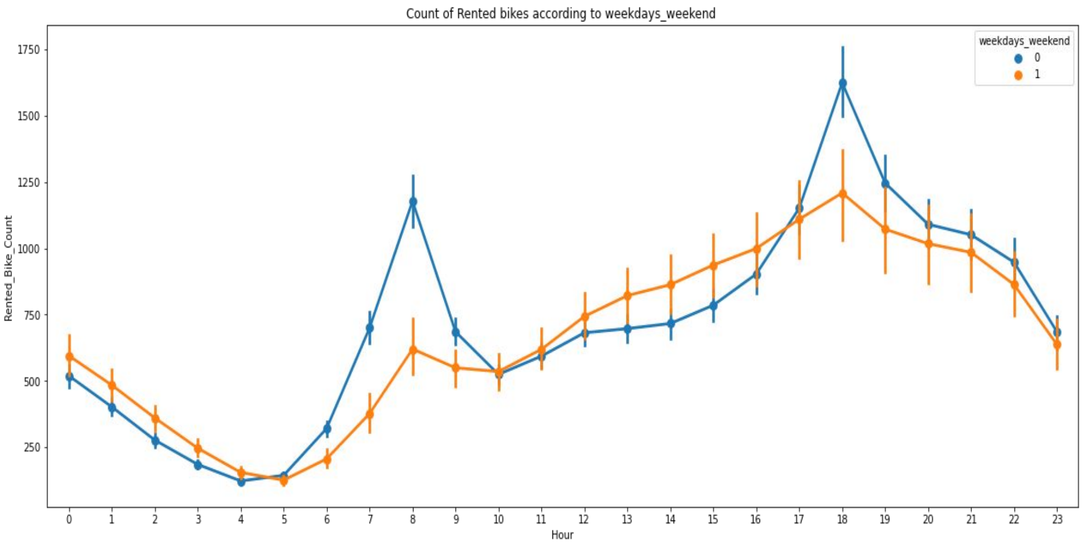
Exploratory Data Analysis(EDA)

Bike Rental count with respect to Hour on Weekdays_Weekends:



From bar plot we can say that the demand of the bikes is higher on weekdays than on the weekends.

This might be due to the office for employees or schools/colleges for children.



From the point plot we can say that Peak Times are between 7 am-9 am and 5 pm-7 pm.

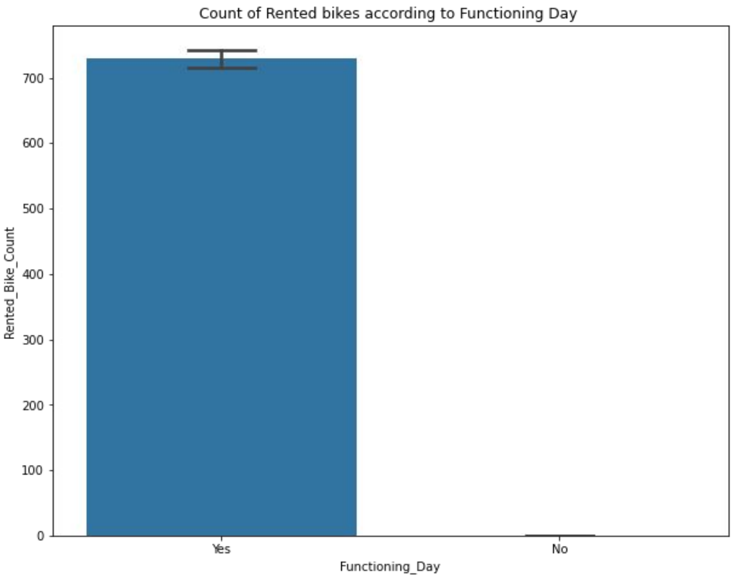
The orange line represents the weekend days, and it shows that the demand of rented bikes are very low in the mornings but in the evening from 4 pm to 8 pm the demand slightly increases.

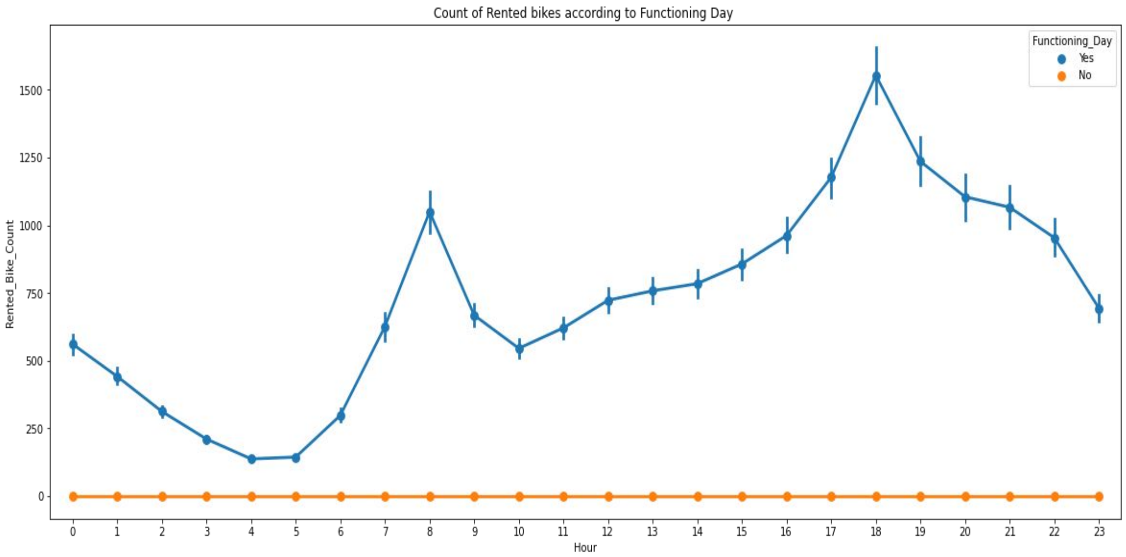
Exploratory Data Analysis(EDA)

Bike Rental count with respect to Hour on Functioning_Day:

From the bar plot and point plot which shows the use of rented bikes on functioning day and non functioning day, we can say that, Peoples dont use rented bikes on non functioning days.

We can also say from the above that count of rented bikes are high between 7am-9am and 5pm-7pm on a Functioning day.





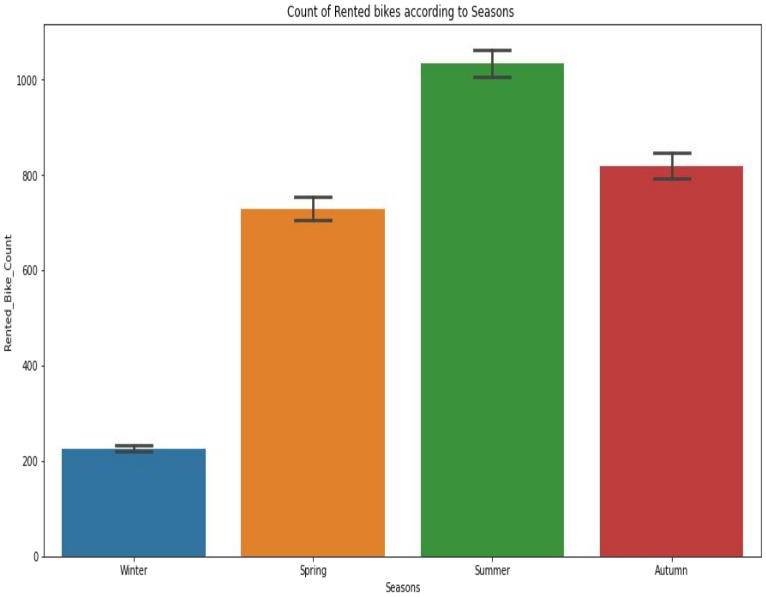
Exploratory Data Analysis(EDA)

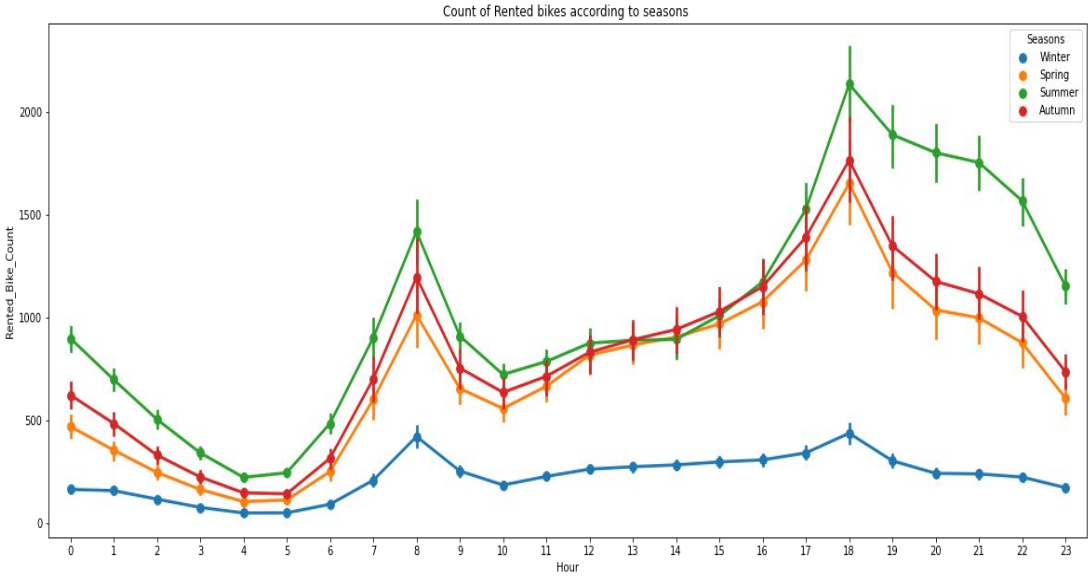
Bike Rental count with respect to Hour on different Seasons:

From the bar plot and point plot which shows the use of rented bikes in four different seasons, we can say that,

In the summer season, use of rented bikes is high and peak time is between 7am-9am and 5pm-7pm.

In the winter season, use of rented bikes is very low, which might be due to the rains and unfavourable weather conditions.





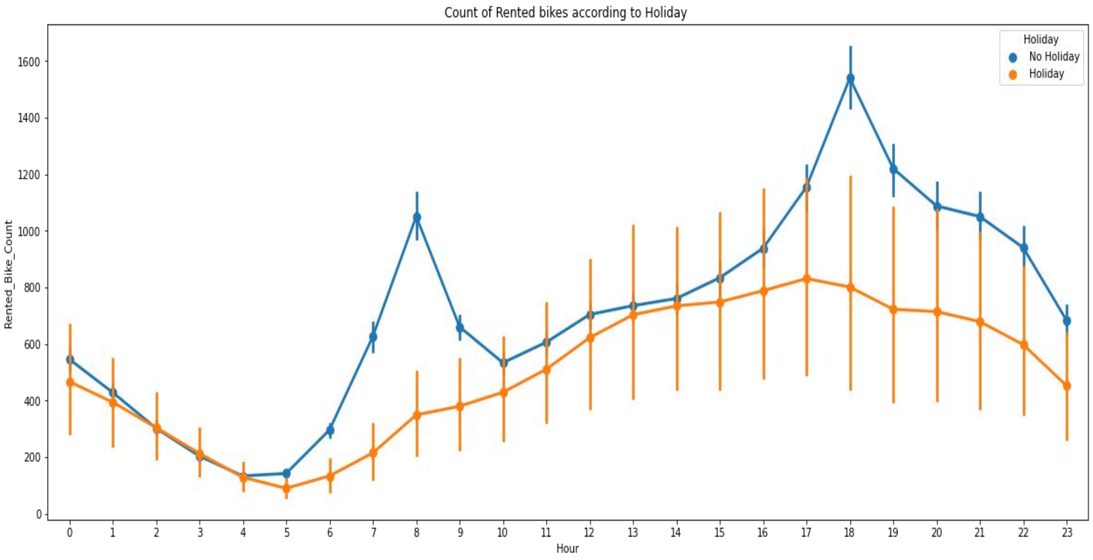
Exploratory Data Analysis(EDA)

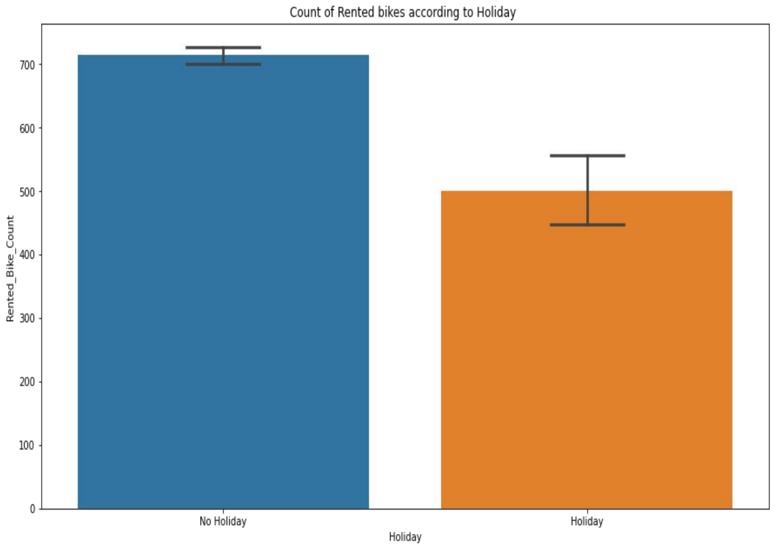
Bike Rental count with respect to Hour on Holiday:

From the bar plot and point plot which shows the use of rented bike in a holiday, we can say that,

Use of Rented bikes is high on No Holidays than on Holidays and the peak timings are between 7am-9am and 5pm-9pm.

We can also say that on Holiday people uses the rented bike from 2pm-7pm.

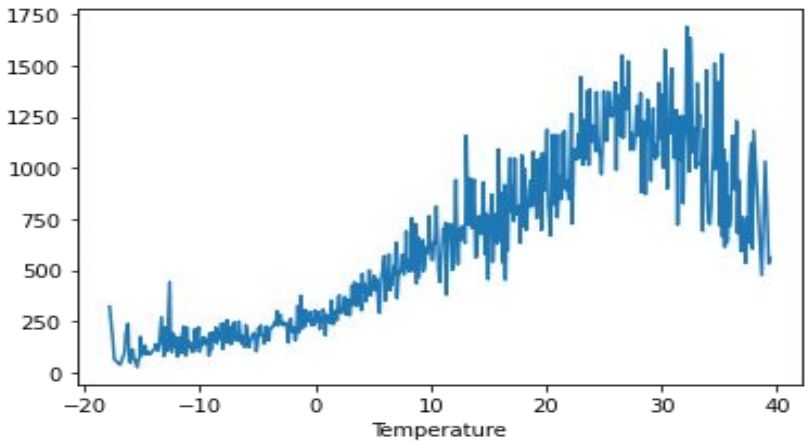




Exploratory Data Analysis(EDA)

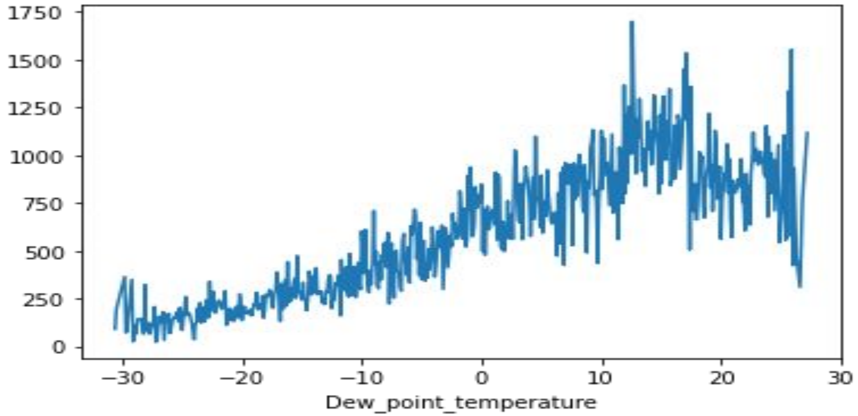
Analysis on Numerical Variables:

Rented_Bike_Count with respect to Temperature:



From the above plot we can say that people like to ride bikes when the temperature is above 20 and are less probable to ride bikes in lower temperatures

Rented_Bike_Count and Dew_point_temperature:

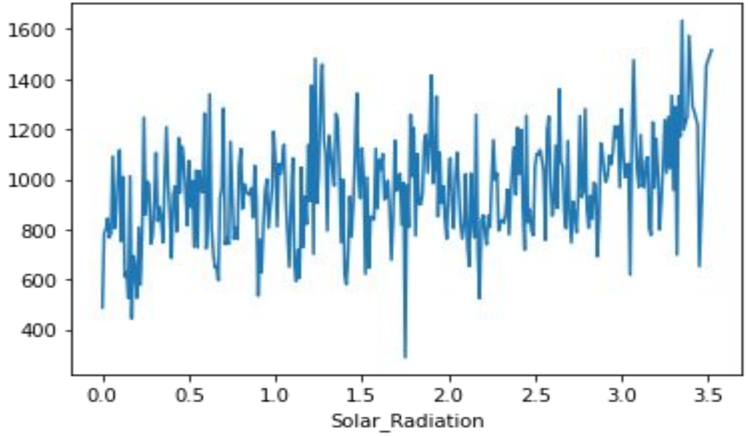


From the above plot, we can observe from the results that, "Dew_point_temperature' is almost same as the 'temperature',

From this we can assume that there must be some correlation present between them.

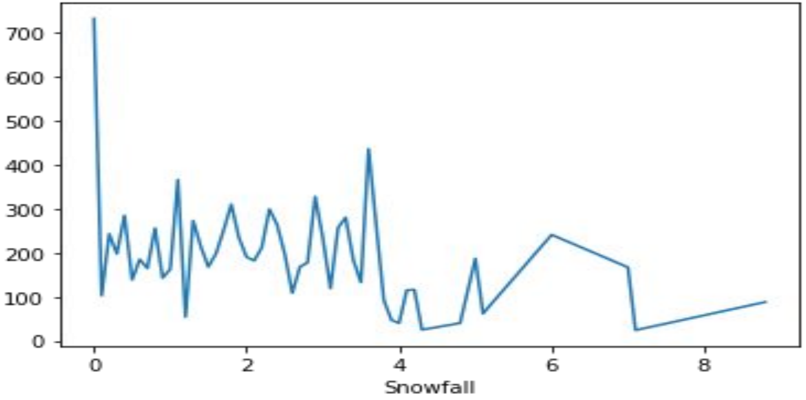
Exploratory Data Analysis(EDA)

Rented_Bike_Count with respect to Solar_Radiation:



From the above plot we can say that, the amount of rented bikes is huge, when there is solar radiation, and the average count of bikes rented is around 1000

Rented_Bike_Count with respect to Snowfall:

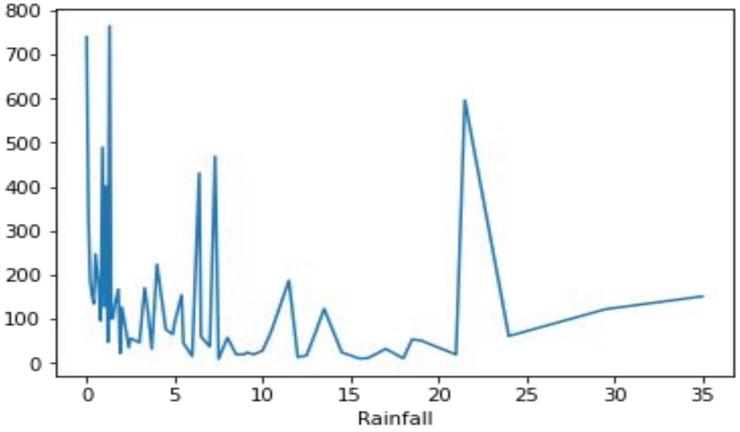


From the plot we can see that, the amount of rented bike is very low and when we have a snowfall of more than 4 cm, the bike rents count is much lower.

Exploratory Data Analysis(EDA)

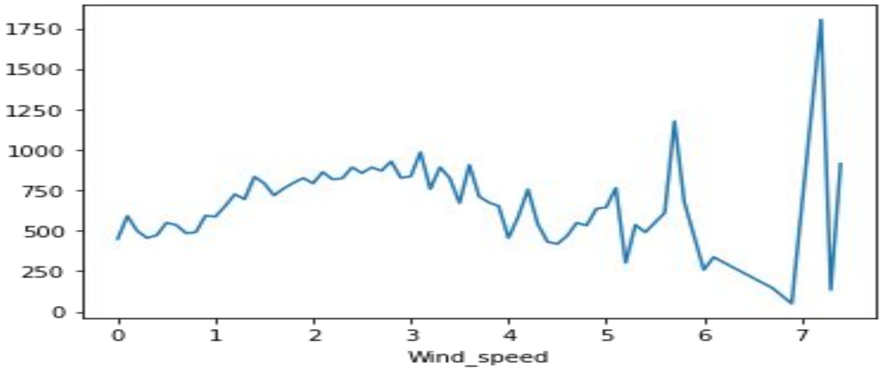
AI

Rented_Bike_Count with respect to Rainfall:



From the above plot we can say that even with rainfalls, demand for rental bikes is not decreasing, we can see from above that, even having a rainfall of 20 mm, there is a big peak of rented bikes

Rented_Bike_Count with respect to Wind_Speed:

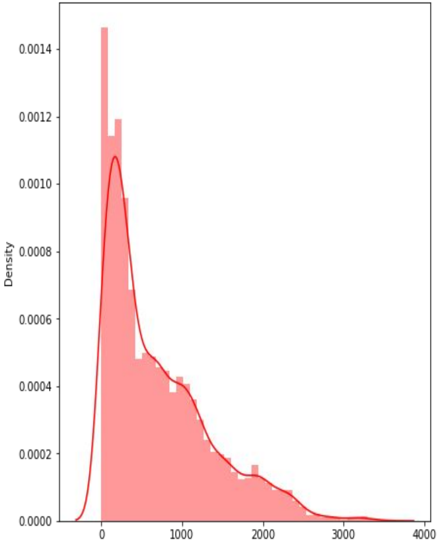


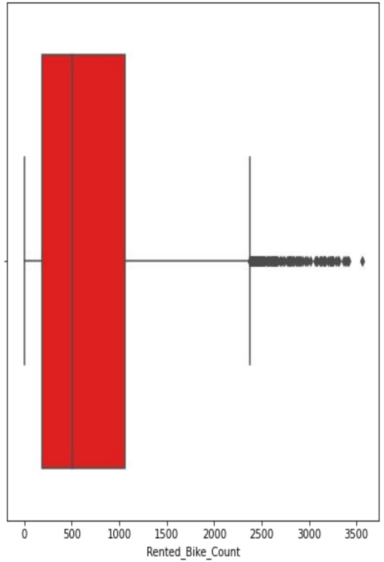
From the above plot we can say that the demand of rented bike is uniformly distributed despite of wind speed.

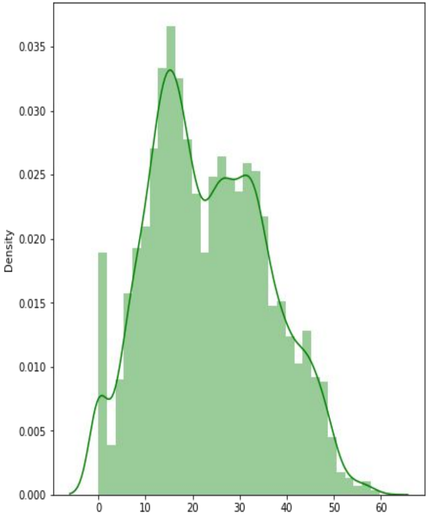
We can also see that, when the speed of wind is 7 m/s, the demand for bike rentals increased rapidly, from which we can say that peoples love to ride bikes when its little windy.

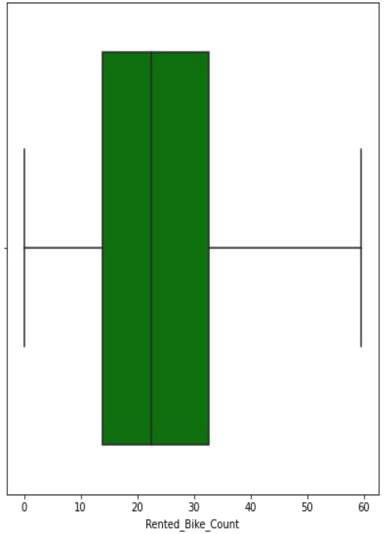
Exploratory Data Analysis(EDA)

Distribution of target variable- "Bike Rented Count”









The above graph shows that, Distribution of Rented Bike Count is slightly right skewed.

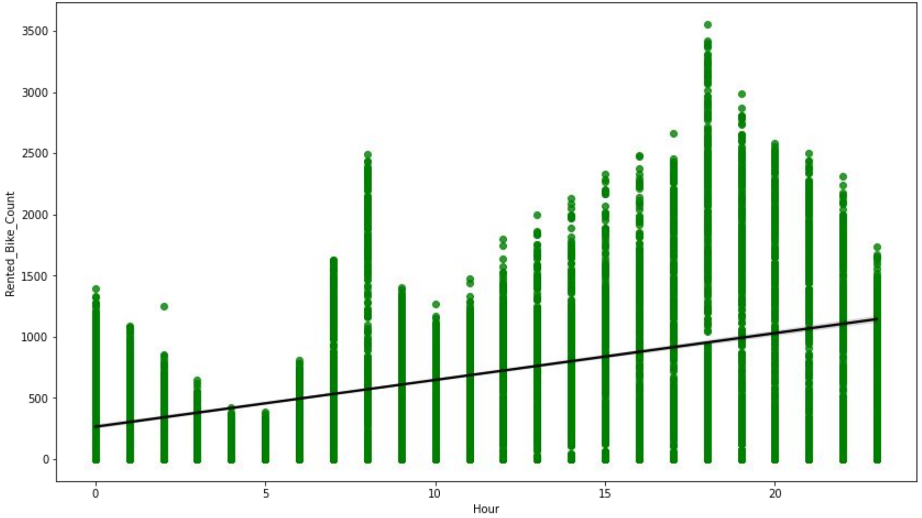
From the boxplot we can see that we have outliers in Rented Bike Count attribute.

Thus we normalized our dependent variable by square root method and also in boxplot above we can see that there are no outliers present after normalization.

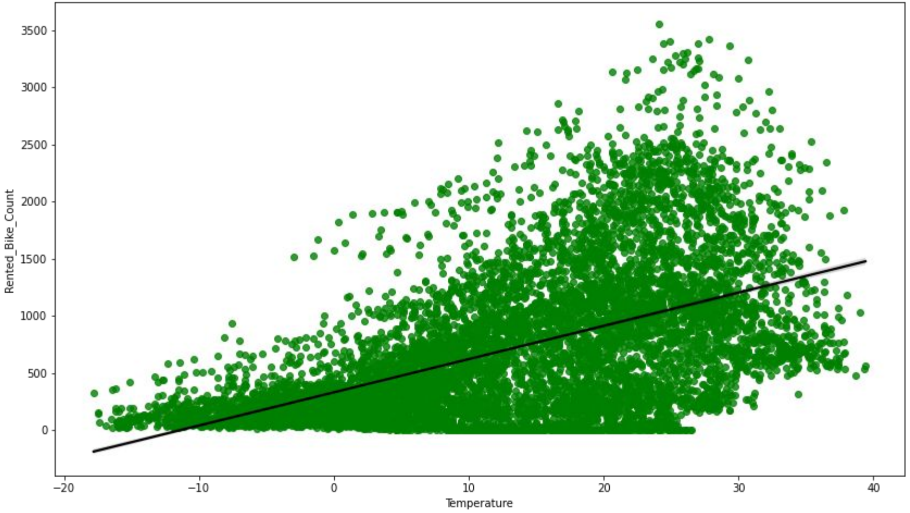
Exploratory Data Analysis(EDA)

Checking the relationship between the dependent variable-"Rented Bike Count' and independent variables through Regression Plot

Regression plot for Hour:

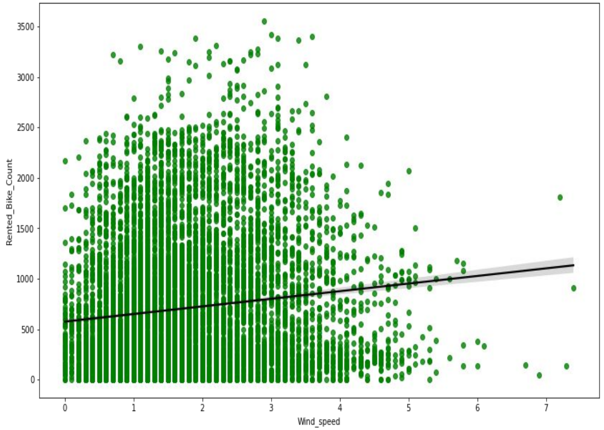


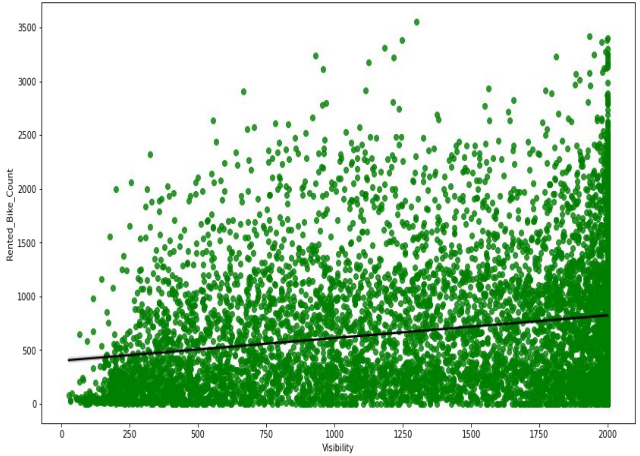
Regression plot for Temperature:



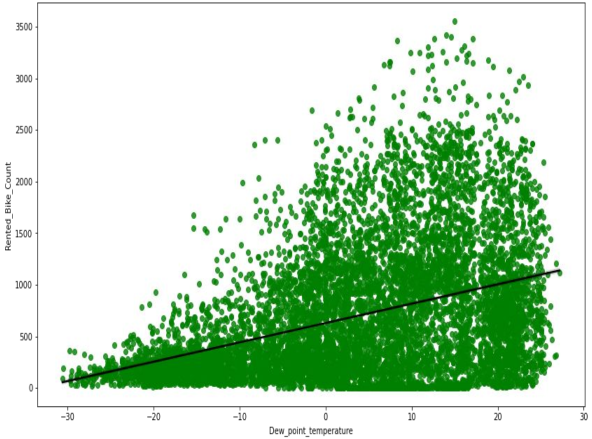
From the above regression plots we can say that Hour and Temperature are Positively related with the dependent variable.

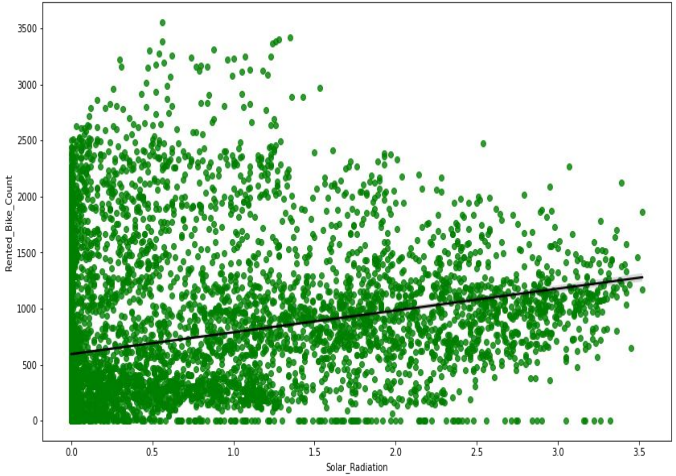
Exploratory Data Analysis(EDA)



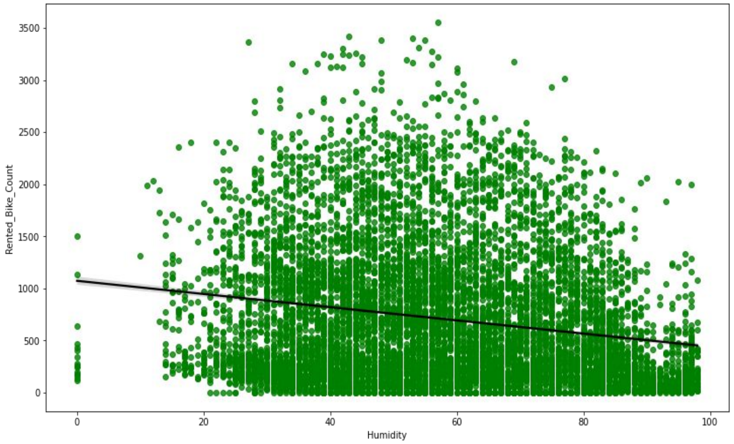


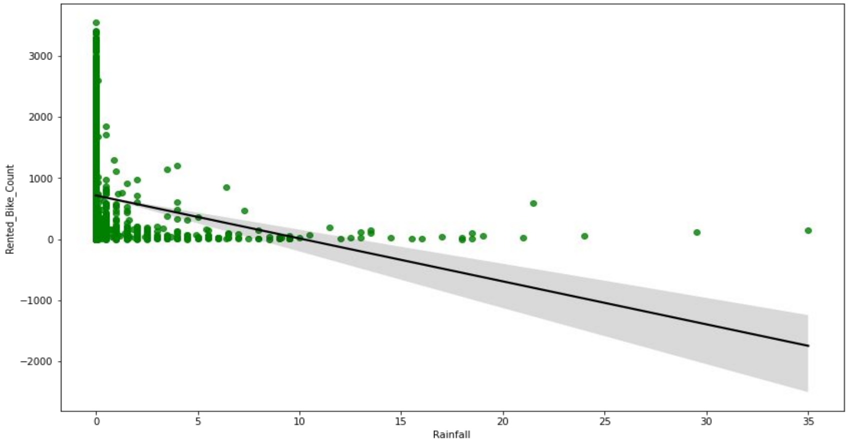
From the plots we can observed that,
'Wind_speed', 'Visibility', 'Dew_point_temperature', 'Solar_Radiation' are positively related to the target variable, which means the rented bike count increases with increase of these features.



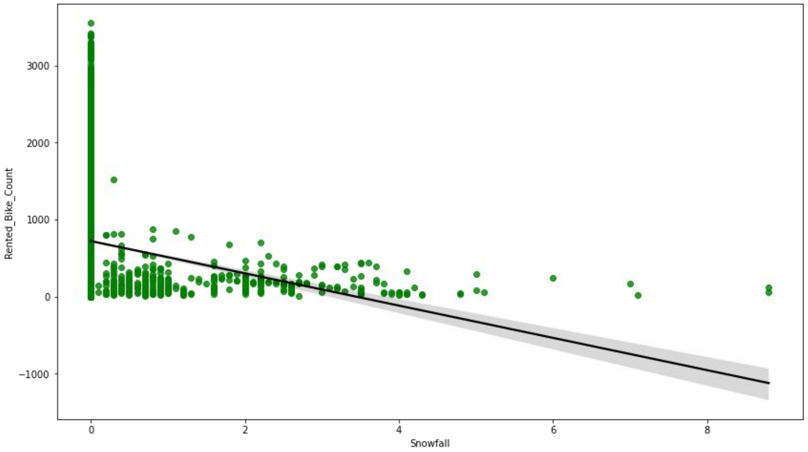


Exploratory Data Analysis(EDA)

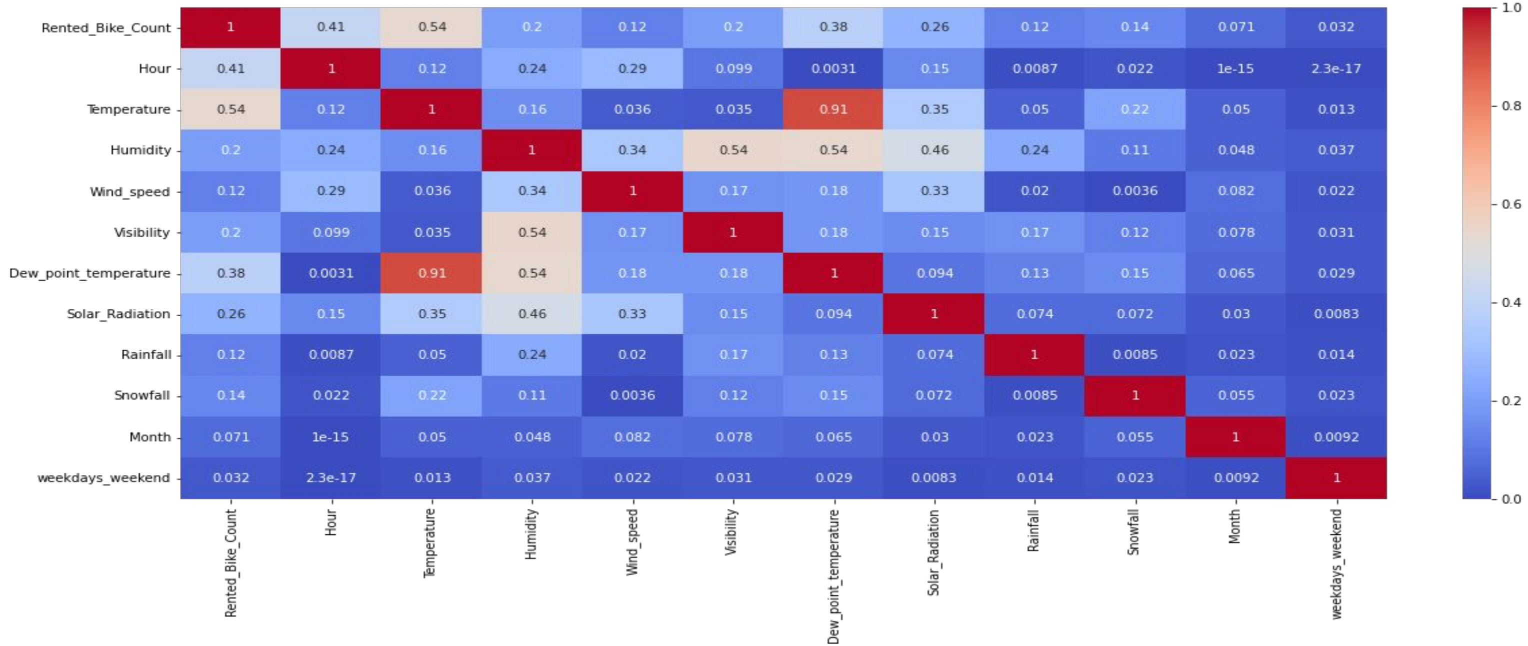




From the above Regression plots we observed that,
'Rainfall', 'Snowfall', 'Humidity' features are negatively related with the target variable which means the rented bike count decreases when these features increase.



Preparing the data for Modeling



We plotted an heatmap to check the correlation between the features, and dropped those with high correlation.
We can observe that "**Temperature**" and "**Dew point Temperature**" are highly correlated(91%).
As per our regression assumption, there should not be collinearity between independent variables. So we can drop one of them

As the correlation between temperature and our dependent variable "Bike Rented Count" is high, we will Keep the Temperature column and drop the "Dev
Point Temperature" column.

Preparing the data for Modeling

AI

```
#Checking the VIF
Calculate_vif(df[[i for i in df.describe().columns if i not in ['Rented_Bike_Count']]])
```

	variables	VIF
0	Hour	3.961874
1	Temperature	3.236809
2	Humidity	6.114153
3	Wind_speed	4.616621
4	Visibility	5.404128
5	Solar_Radiation	2.272068
6	Rainfall	1.081252
7	Snowfall	1.125304
8	Month	4.580307
9	weekdays_weekend	1.399708

On checking for VIF(Variance Inflation Factor), we observed that ‘Visibility’ & ‘Humidity’ have VIF value greater than 5, so we dropped those attributes

Later on we created dummy variables for categorical **Season** attribute and had done labelling for **Holiday** and **Functioning day** attributes as 0 and 1 as a part of data preparation for model building

```
[ ] # Createing dummy variables for seasons
    df=pd.get_dummies(df,columns=['Seasons'],prefix='Seasons',drop_first=True)
```

```
[ ] # Labeling for holiday=1 and no holiday=0
    df['Holiday']=df['Holiday'].map({'No Holiday':0, 'Holiday':1})
```

```
[ ] # # Labeling for Yes=1 and No=0
    df['Functioning_Day']=df['Functioning_Day'].map({'Yes':1, 'No':0})
```

By this the data was good and well prepared for modeling

Model selection, Execution and Evaluation

As this is a Regression problem, we used the following Regression Models for Evaluation

- Linear Regression
- Lasso Regression
- Ridge Regression
- Elastic Net Regression
- Decision Tree Regression
- Random Forest Regression
- Gradient Boosting Regression

There are some basic assumptions that must be fulfilled before implementing Linear Regression algorithm. They are:

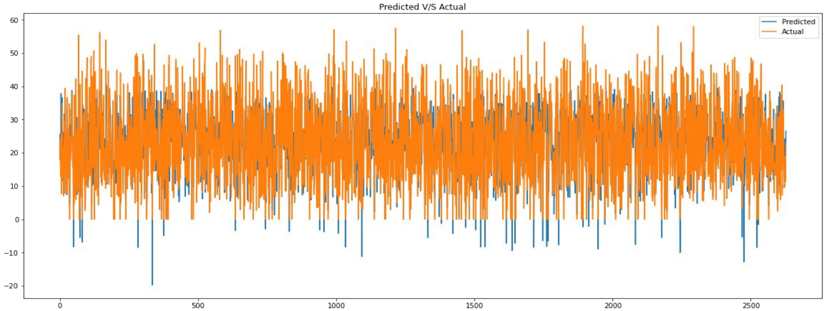
1. No multicollinearity in the dataset.
2. Independent variables should show linear relationship with dependent variable.
3. Residual mean should be 0 or close to 0.
4. There should be no heteroscedasticity i.e., variance should be constant along the line of best fit.

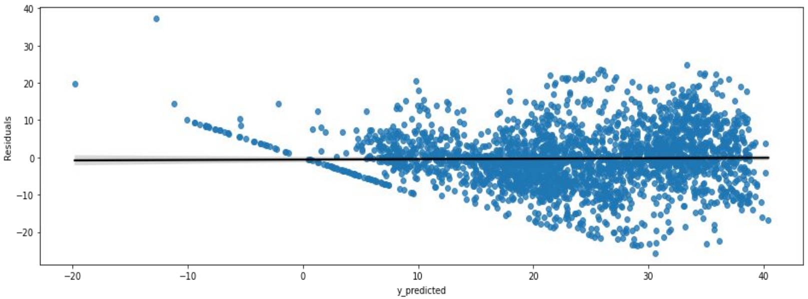
Model selection, Execution and Evaluation

Linear Regression

Train Set Results:

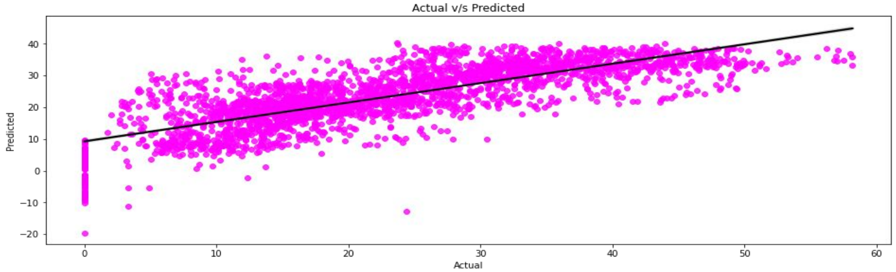
The Mean Absolute Error (MAE) is 5.855397241788345.
The Mean Squared Error(MSE) is 60.2994929244455.
The Root Mean Squared Error(RMSE) is 7.765274813195316.
The R2 Score is 0.6123528085603556.

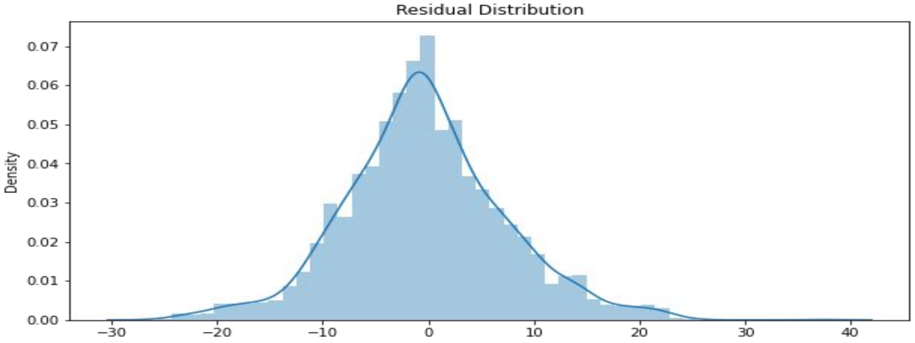




Test Set Results:

The Mean Absolute Error (MAE) is 5.834169822951748.
The Mean Squared Error(MSE) is 58.624247223024895.
The Root Mean Squared Error(RMSE) is 7.656647257319936.
The R2 Score is 0.618326967365199.



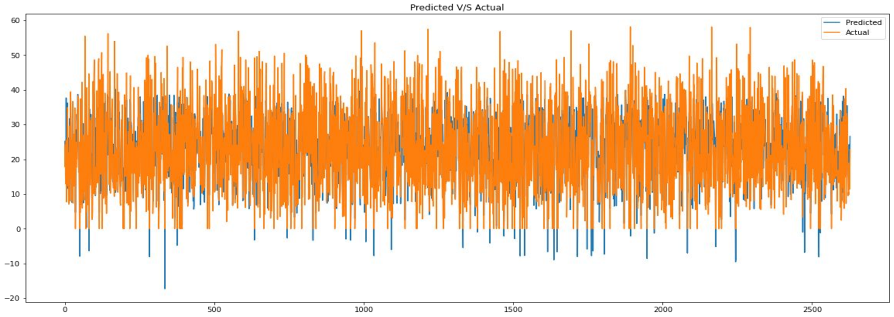


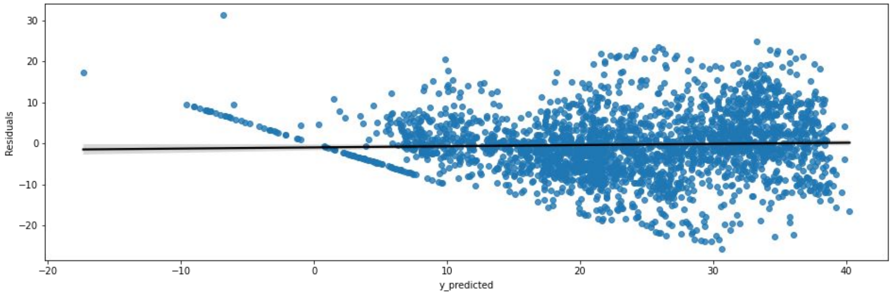
Model selection, Execution and Evaluation

Lasso Regression

Train Set Results:

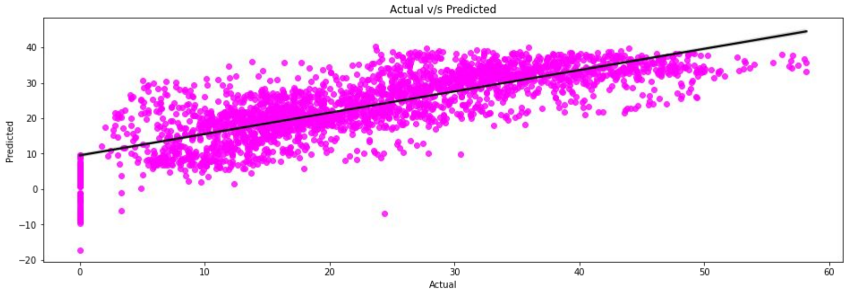
The Mean Absolute Error (MAE) is 5.869103531726283.
The Mean Squared Error(MSE) is 60.4640243649349.
The Root Mean Squared Error(RMSE) is 7.775861647749624.
The R2 Score is 0.6112950857219155.

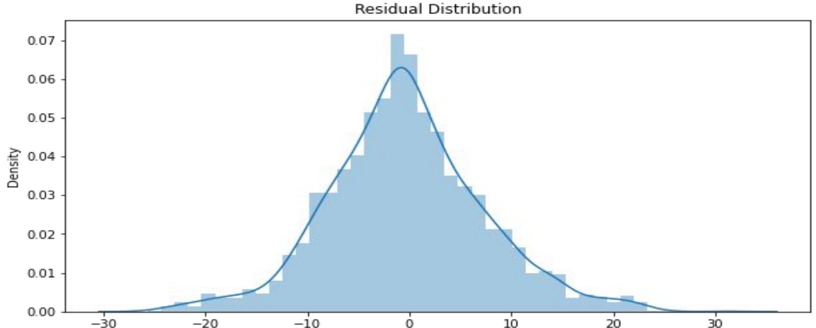




Test Set Results:

The Mean Absolute Error (MAE) is 5.85056642263689.
The Mean Squared Error(MSE) is 58.79268408749925.
The Root Mean Squared Error(RMSE) is 7.667638755673042.
The R2 Score is 0.61723035952942.





Model selection, Execution and Evaluation

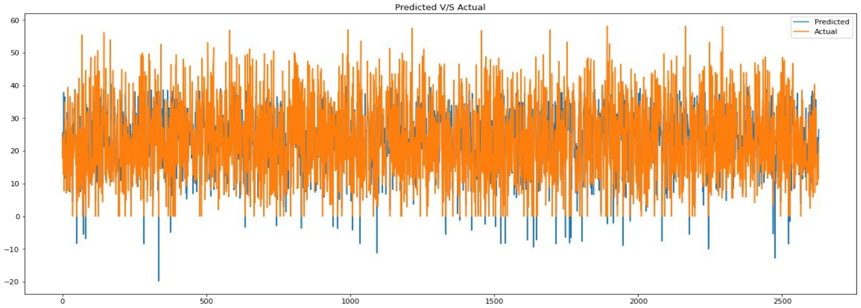
Ridge Regression

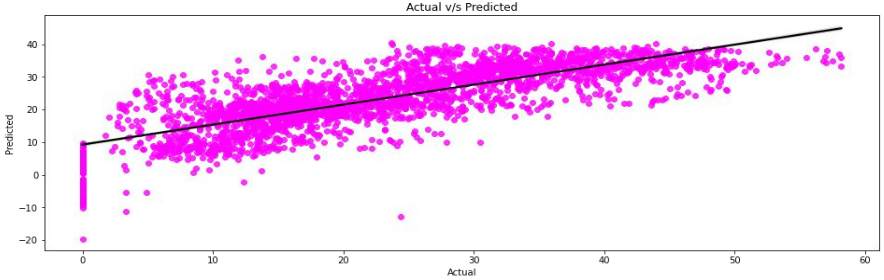
Train Set Results:

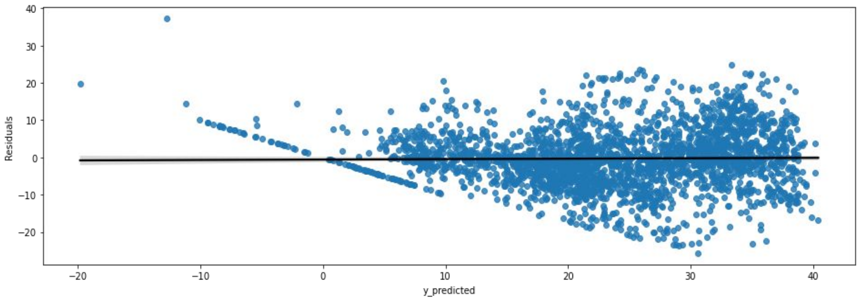
The Mean Absolute Error (MAE) is 5.869103531726283.
The Mean Squared Error(MSE) is 60.4640243649349.
The Root Mean Squared Error(RMSE) is 7.775861647749624.
The R2 Score is 0.6112950857219155.

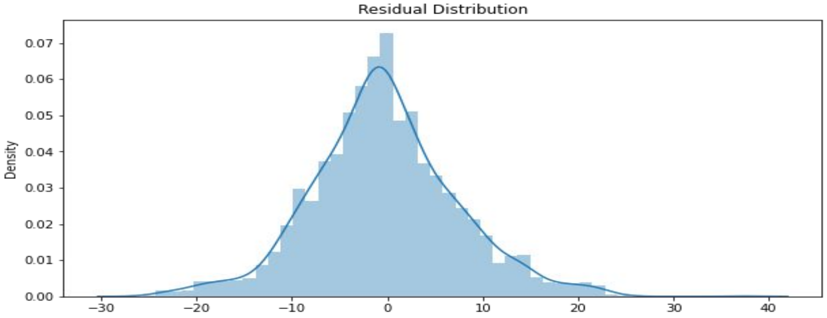
Test Set Results:

The Mean Absolute Error (MAE) is 5.850566426263689.
The Mean Squared Error(MSE) is 58.79268408749225.
The Root Mean Squared Error(RMSE) is 7.667638755673042.
The R2 Score is 0.61723035952942.









Model selection, Execution and Evaluation

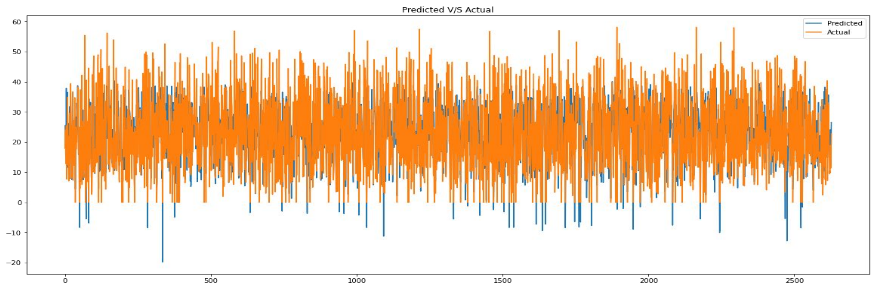
Elastic Net Regression

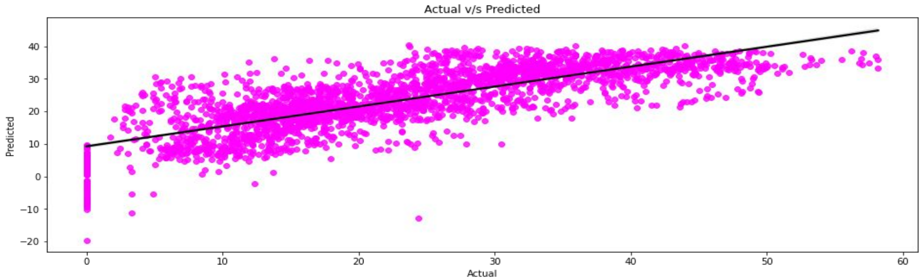
Train Set Results:

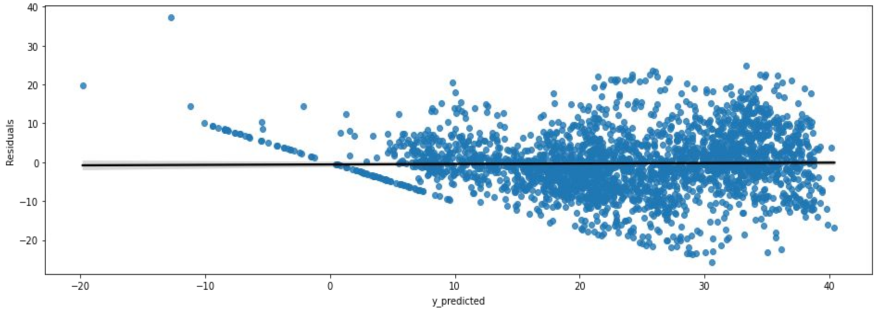
The Mean Absolute Error (MAE) is 5.8932275545714745.
The Mean Squared Error(MSE) is 60.90273656811195.
The Root Mean Squared Error(RMSE) is 7.804020538678249.
The R2 Score is 0.6084747377362095.

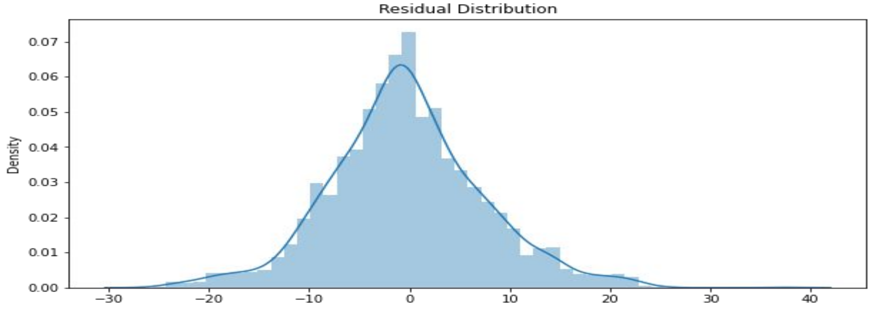
Test Set Results:

The Mean Absolute Error (MAE) is 5.834169822951748.
The Mean Squared Error(MSE) is 58.624247223024895.
The Root Mean Squared Error(RMSE) is 7.656647257319936.
The R2 Score is 0.618326967365199.









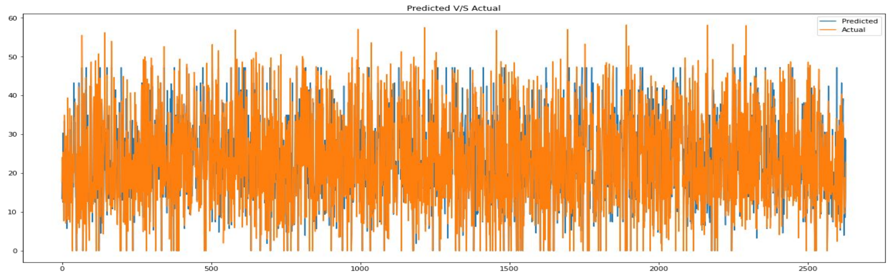
Model selection, Execution and Evaluation

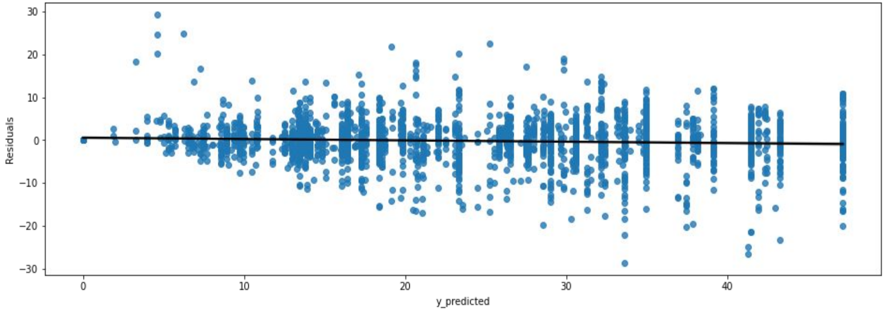
AI

Decision Tree Regression: (Hyper parameter tuned: max_depth=9, max_features= 'auto')

Train Set Results:

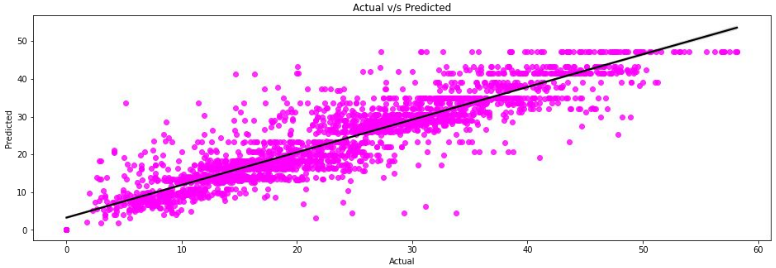
The Mean Absolute Error (MAE) is 2.885516521569070б.
The Mean Squared Error(MSE) is 18.44462508772692.
The Root Mean Squared Error(RMSE) is 4.294720606480347.
The R2 Score is 0.8814250872495163.

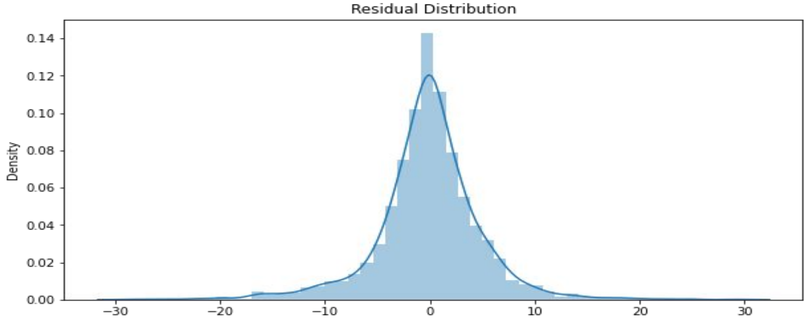




Test Set Results:

The Mean Absolute Error (MAE) is 3.405322370033263.
The Mean Squared Error(MSE) is 25.08982578263722.
The Root Mean Squared Error(RMSE) is 5.00897452405552.
The R2 Score is 0.8366527444129481.





Model selection, Execution and Evaluation

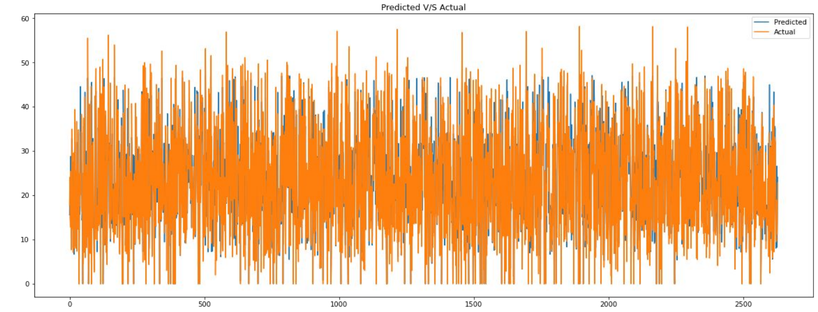
Random Forest Regression:(Hyper parameter tuned: max_depth=9, ‘n_estimators= ‘100’)

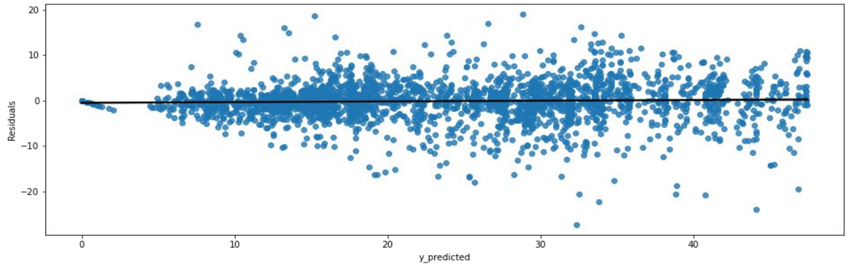
Train Set Results:

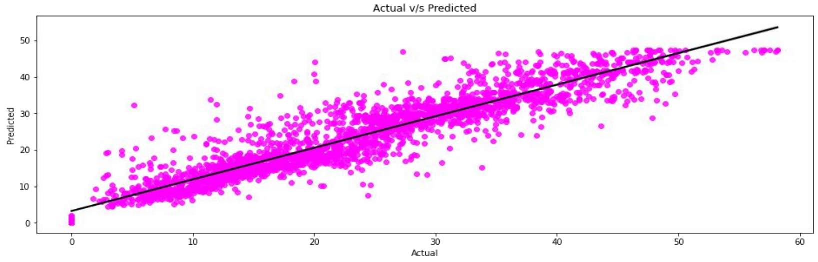
The Mean Absolute Error (MAE) is 2.625998855026434.
The Mean Squared Error(MSE) is 14.875654965147472.
The Root Mean Squared Error(RMSE) is 3.8568970643702007.
The R2 Score is 0.9043689160820976.

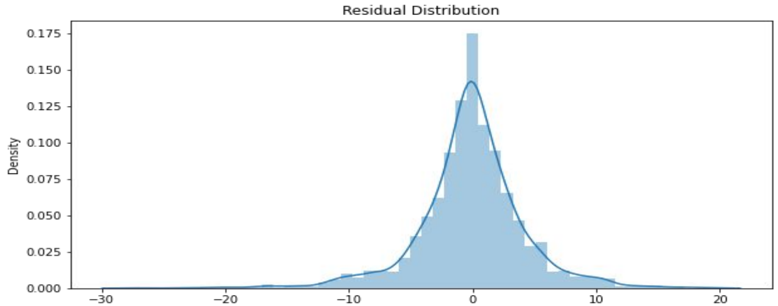
Test Set Results:

The Mean Absolute Error (MAE) is 2.952489659032659.
The Mean Squared Error(MSE) is 18.700684900458626.
The Root Mean Squared Error(RMSE) is 4.324428852514355.
The R2 Score is 0.8782492320770888.









Model selection, Execution and Evaluation

AI

Gradient Boosting Regression:

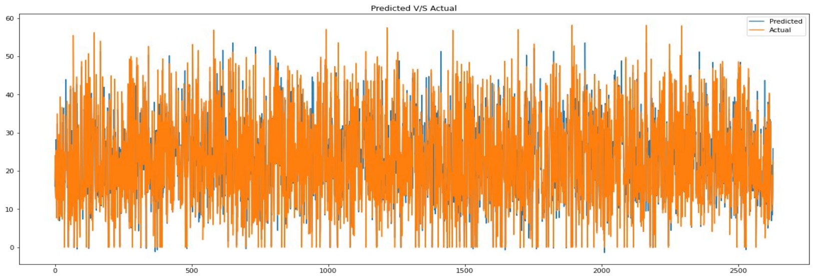
(Hyper parameter tuned: ‘Learning_rate=0.04-max_depth=8, ‘n_estimators= ‘150’- ‘subsample’=0.9)

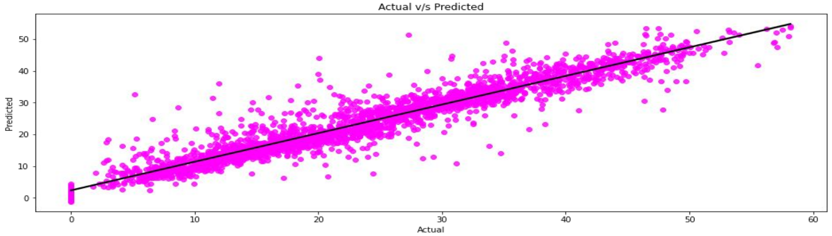
Train Set Results:

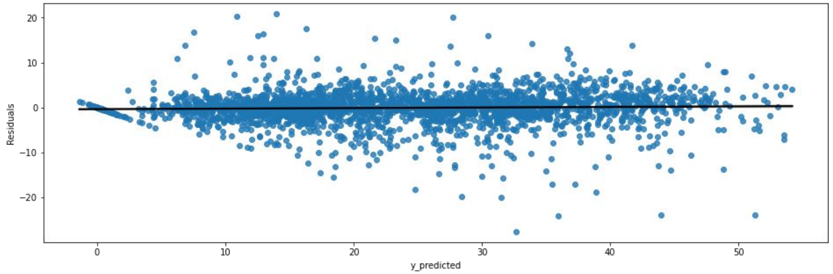
The Mean Absolute Error (MAE) is 1.5034847587722098.
The Mean Squared Error(MSE) is 4.749558591419942.
The Root Mean Squared Error(RMSE) is 2.17934820334428.
The R2 Score is 0.9694665251853957.

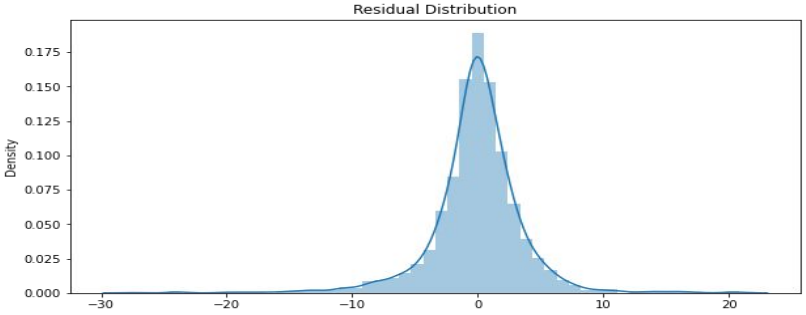
Test Set Results:

The Mean Absolute Error (MAE) is 2.387989114802619.
The Mean Squared Error(MSE) is 13.220271451193188.
The Root Mean Squared Error(RMSE) is 3.63596912132009.
The R2 Score is 0.9139294517874778.

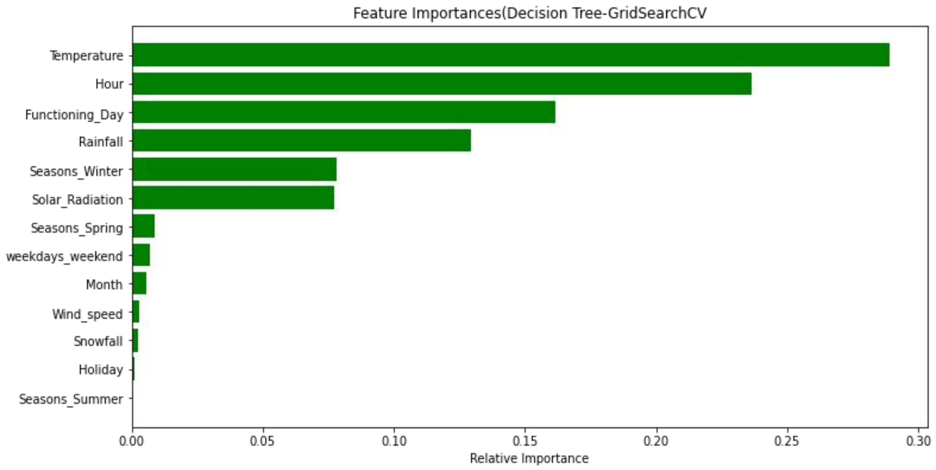


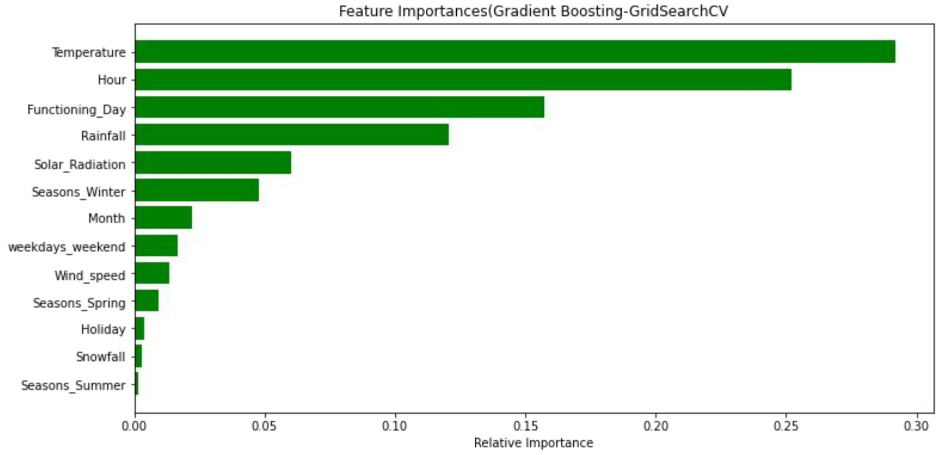


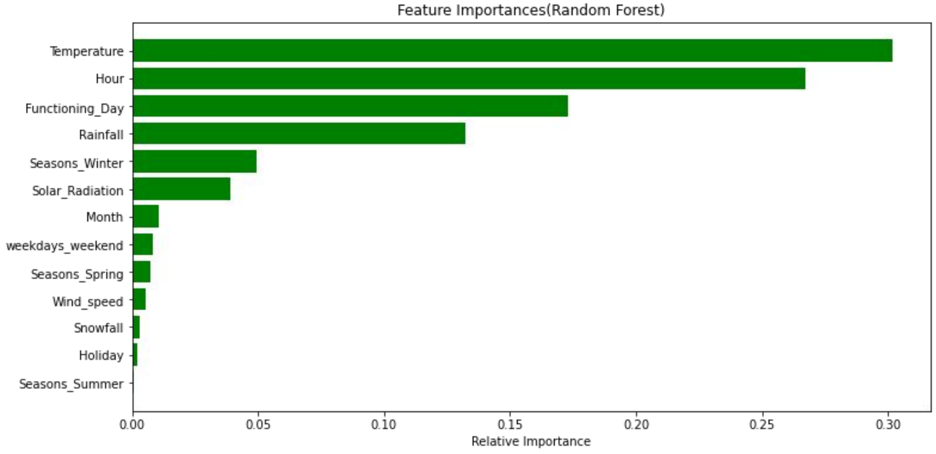




Feature Importance







From all the three models we can say that ‘**Temperature**’, ‘**Hour**’, ‘**Functioning_Day**’ are playing a very important role in bike rentals.

Conclusions

AI

		Model	MAE	MSE	RMSE	R2_score
Training set	**0**	Linear Regression	5.8555	60.2995	7.7653	0.6124
	1	Lasoo	5.8691	60.4640	7.7759	0.6113
	2	RidgeGridSearchCV	5.8691	60.4640	7.7759	0.6113
	3	ElasticNet(GridSearchCV-Tunned)	5.8932	60.9027	7.8040	0.6085
	4	Decision Tree Regressor-GridSearchCV	2.8855	18.4446	4.2947	0.8814
	5	Random Forest	0.9458	2.2026	1.4841	0.9858
	6	Random Forest-GridSearchCv	2.6214	14.8353	3.8517	0.9046
	7	Gardient boosting Regression	3.1772	20.5277	4.5308	0.8680
	8	Gradient Boosting Regression(GridSearchCV)	1.5162	4.8189	2.1952	0.9690
Test set	**0**	Linear Regression	5.8342	58.6242	7.6566	0.6183
	1	Lasso	5.8506	58.7927	7.6676	0.6172
	2	Ridge(GridsearchCv Tunned)	5.8342	58.6242	7.6566	0.6183
	3	ElasticNet(GridSearchCV-Tunned)	5.8342	58.6242	7.6566	0.6183
	4	Decision Tree Regressor(GridsearchCV)	3.4004	24.9284	4.9928	0.8377
	5	Radom forest	2.5205	14.6099	3.8223	0.9049
	6	Random Forest-GridSearchCv	2.9373	18.5110	4.3024	0.8795
	7	Gradient Boosting Regression	3.2825	21.6738	4.6555	0.8589
	8	Gradient Boosting Regression(GridSearchCV)	2.3747	13.1347	3.6242	0.9145

From the Final Dataframe, we can see that **Linear**, **Lasso**, **Ridge** and **Elastic Net Regression** models have almost similar R2 scores(61%) on both training and test data. Even after using GridsearchCV we have obtained similar results.

In case of **Decision Tree Regression** model, we got a R2 score of 88% on training data and 83% on test data after hyperparameter tuning, which is quite good for us.

For **Random Forest Regression** model, without hyperparameter tuning we got a R2 score of 98% on training data and 90% on test data.

Thus our model memorised the data and we can assume that it was an overfitted model.

Conclusions

		Model	MAE	MSE	RMSE	R2_score
Training set	0	Linear Regression	5.8555	60.2995	7.7653	0.6124
	1	Lasoo	5.8691	60.4640	7.7759	0.6113
	2	RidgeGridSearchCV	5.8691	60.4640	7.7759	0.6113
	3	ElasticNet(GridSearchCV-Tunned)	5.8932	60.9027	7.8040	0.6085
	4	Decision Tree Regressor-GridSearchCV	2.8855	18.4446	4.2947	0.8814
	5	Random Forest	0.9458	2.2026	1.4841	0.9858
	6	Random Forest-GridSearchCv	2.6214	14.8353	3.8517	0.9046
	7	Gardient boosting Regression	3.1772	20.5277	4.5308	0.8680
	8	Gradient Boosting Regression(GridSearchCV)	1.5162	4.8189	2.1952	0.9690
Test set	0	Linear Regression	5.8342	58.6242	7.6566	0.6183
	1	Lasso	5.8506	58.7927	7.6676	0.6172
	2	Ridge(GridsearchCv Tunned)	5.8342	58.6242	7.6566	0.6183
	3	ElasticNet(GridSearchCV-Tunned)	5.8342	58.6242	7.6566	0.6183
	4	Decision Tree Regressor(GridsearchCV)	3.4004	24.9284	4.9928	0.8377
	5	Radom forest	2.5205	14.6099	3.8223	0.9049
	6	Random Forest-GridSearchCv	2.9373	18.5110	4.3024	0.8795
	7	Gradient Boosting Regression	3.2825	21.6738	4.6555	0.8589
	8	Gradient Boosting Regression(GridSearchCV)	2.3747	13.1347	3.6242	0.9145

After hyperparameter tuning we got a R2 score of 90% on training data and 87% on test data which is a very good result and model is very good to be deployed.

For **Gradient Boosting Regression** model, without hyperparameter tuning we got r2 score of 86% on training data and 85% on test data.

Our model is performing well even without hyperparameter tuning.

After hyperparameter tuning we got a R2 score of 96% on training data and 91% on test data.

Thus our model performance increased by hyperparameter tuning.

Hence **Gradient Boosting Regression(GridSearchCV)** and **Random forest Regression (GridSearchCV)** are the models with high performance and can be deployed.