Capstone Project-2

Bike Sharing Demand Prediction

(Supervised Machine Learning regression)

BY

Prasad Kanagi

Problem Statement:

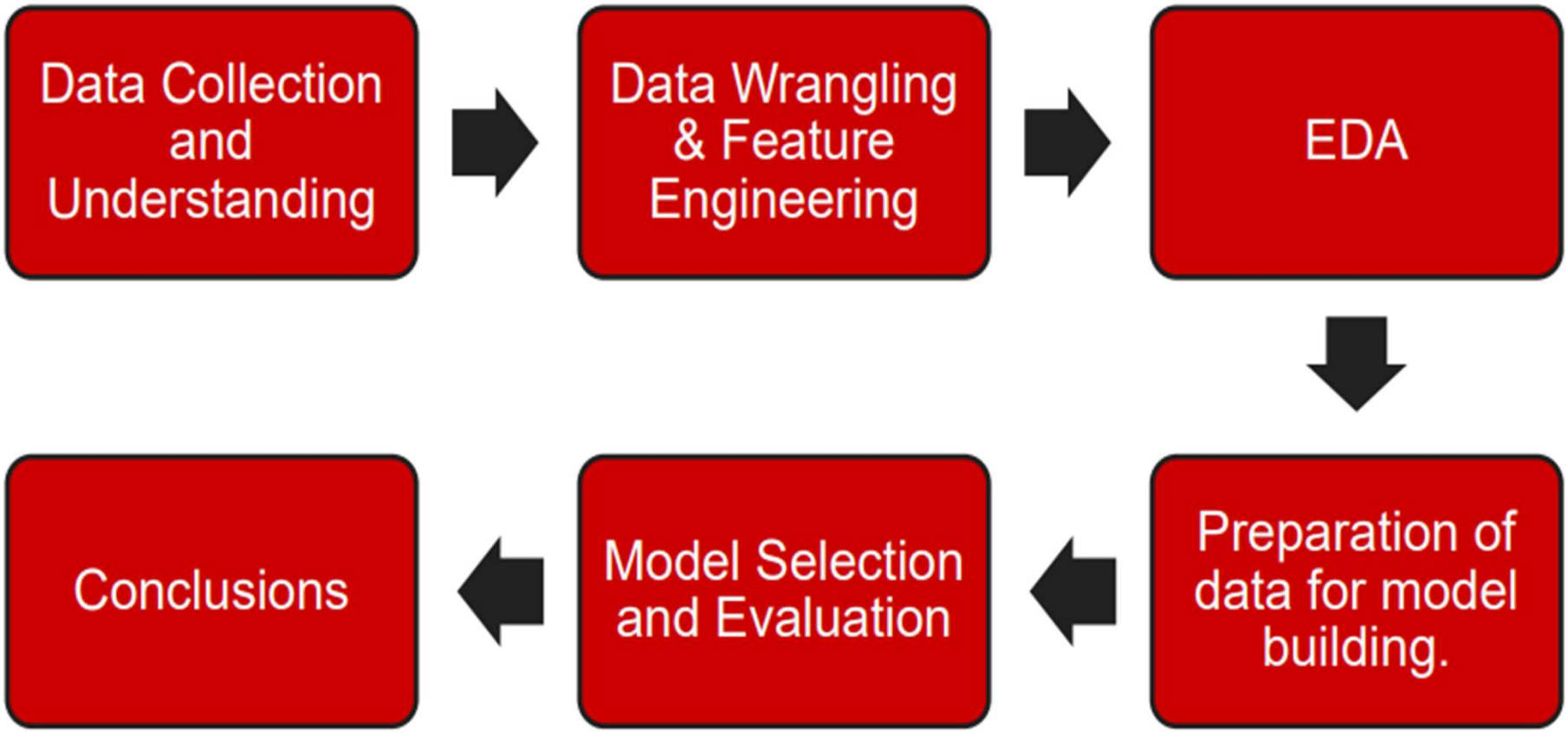
Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. The client is Seoul Bike, which participates in a bike share program in Seoul, South Korea.

An accurate prediction of bike count is critical to the success of the Seoul bike share program. It is important to make the rental 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 final aim of this project is the prediction of bike count required at each hour for the stable supply of rental bikes.

Work Flow :


Data Collection and Understanding
Data Wrangling & Feature Engineering
EDA
Preparation of data for model building.
Model Selection and Evaluation
Conclusions

Data Collection and Understanding:

- The dataset contains weather information (Temperature, Humidity, Wind speed, Visibility, Dew point, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.

Dependent variable:

Rented Bike count - Count of bikes rented at each hour

Independent variables:

- Date : year-month-day
- Hour - Hour of he day
- Temperature-Temperature in Celsius
- Humidity - %
- Windspeed - m/s
- Visibility - 10 m
- Dew point temperature - Celsius

Solar radiation - MJ/m2

Rainfall - mm

Snowfall - cm

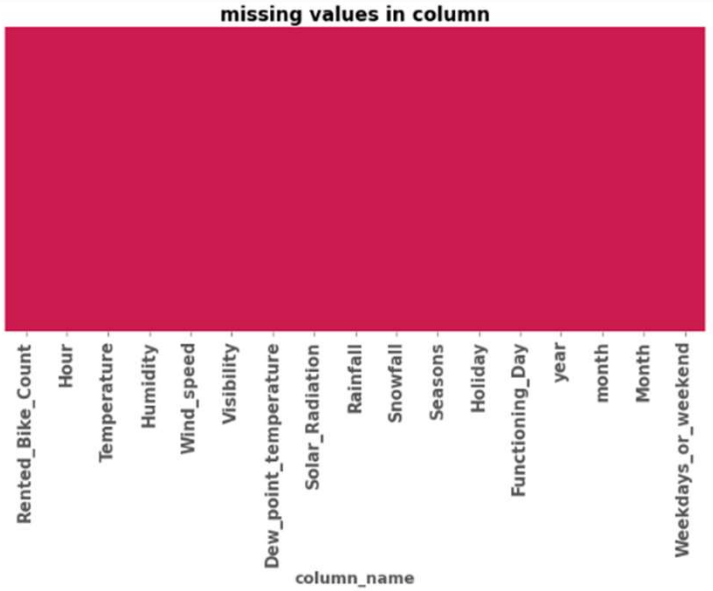
Seasons - Winter, Spring, Summer, Autumn

Holiday - Holiday/No holiday

Functional Day - NoFunc(No Functional Hours), Fun(Functional hours)

Data Wrangling and Feature Engineering:

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

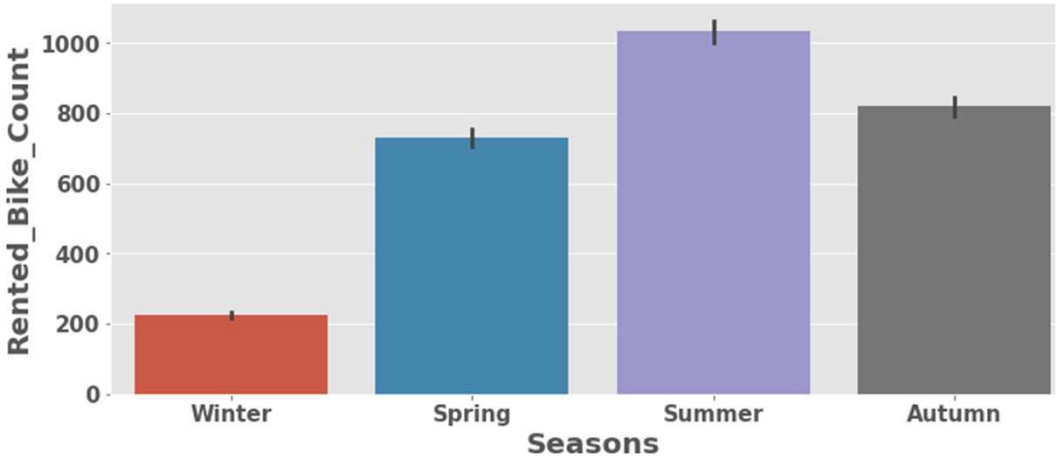



```
[ ]  # Change The datatype of Date columns to extract 'Month' ,'Day', "year"
     df['Date']=df['Date'].astype('datetime64[ns]')
```

```
[ ]  # checking Duplicate rows in our BikeData.
     duplicates=df.duplicated().sum()
     print(f"We have {duplicates} rows in our Bike Data.")
     # No duplicate rows found

     We have 0 rows in our Bike Data.
```

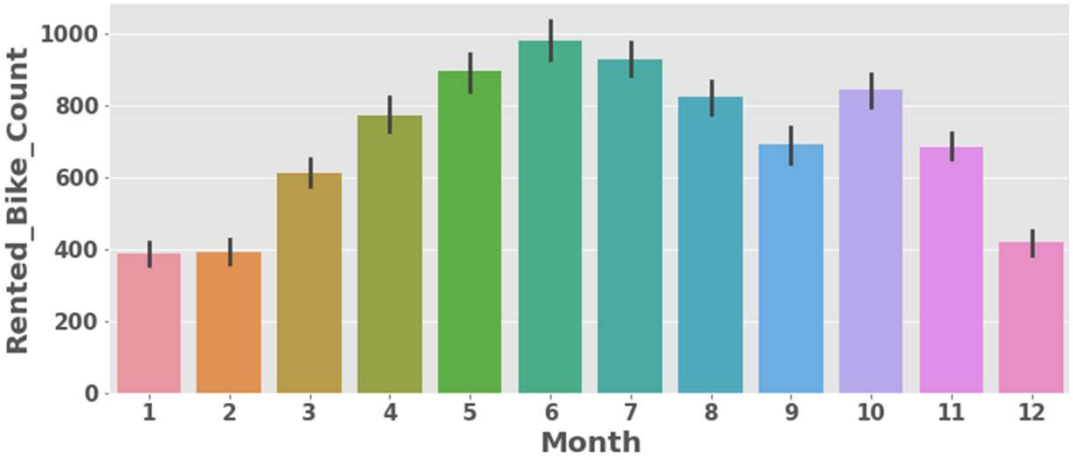
```
     # Creating new columns 'Month', 'Year' ,'Day.
     df['Month']=df['Date'].dt.month

     df['Day']=df['Date'].dt.day_name()
```

EDA (Exploratory Data Analysis):



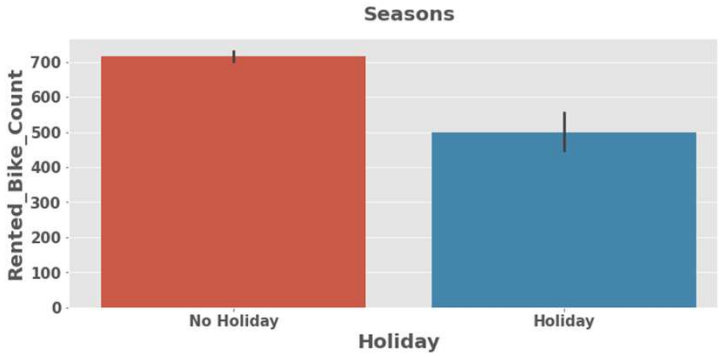
Relation of rented bike count with categorical features:

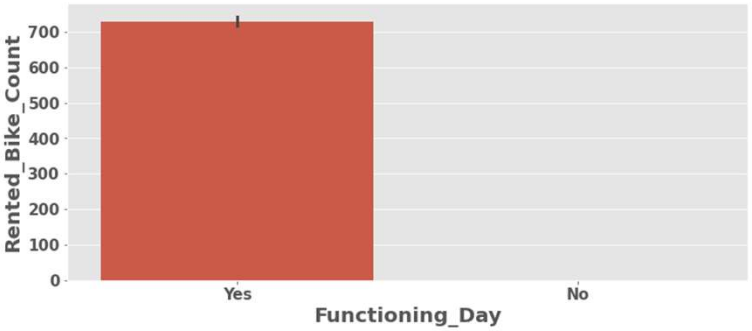
Summer season had the highest Bike Rent Count.
People are more likely to take rented bikes in summer. Bike rentals in winter is very less compared to other seasons.

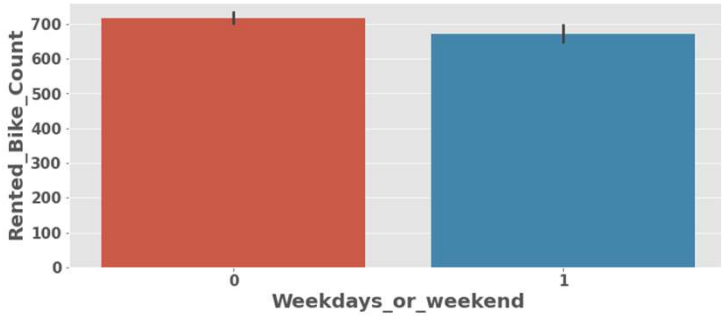


From March Bike Rent Count started increasing and it was highest in June.

EDA (Exploratory Data Analysis):







Conclusions:

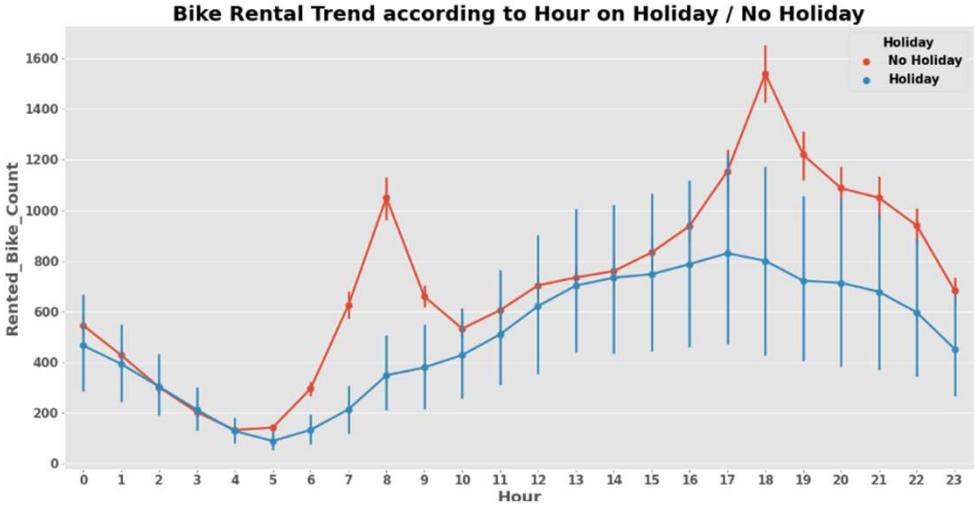
High number of bikes were rented on No Holidays. Which is almost 700 bikes.

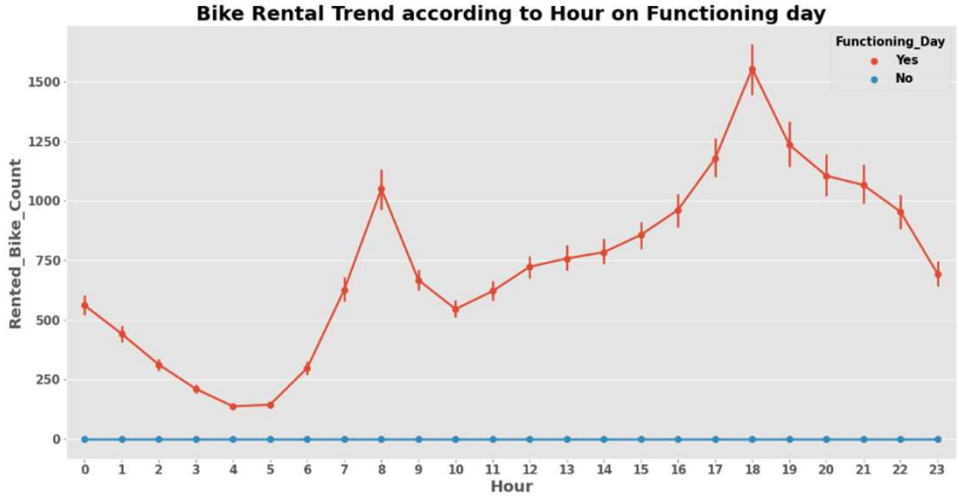
Zero Bikes were rented on no functioning day.
More than 700 bikes rented on functioning day.

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

EDA (Exploratory Data Analysis):

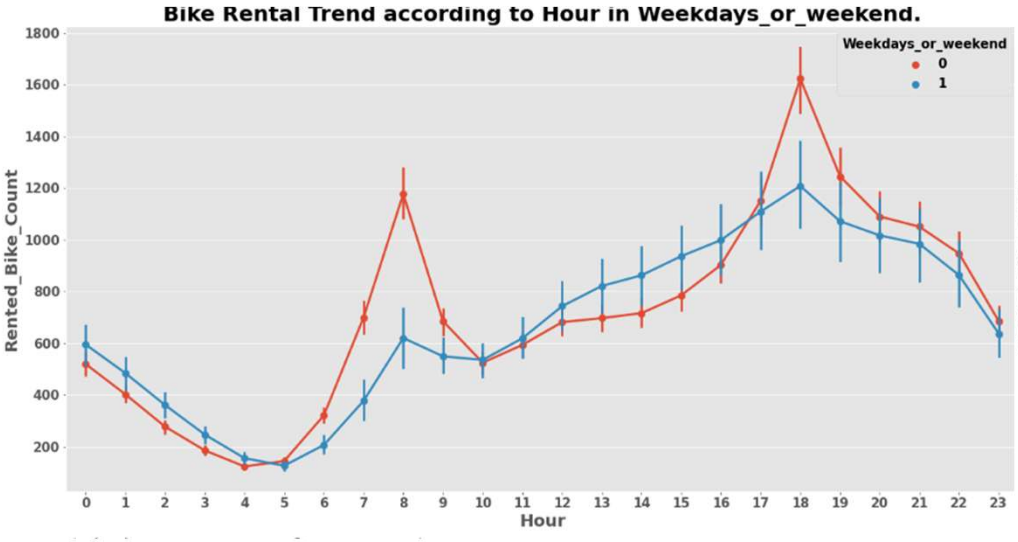
Bike Rent Trend according to hour in different scenarios.

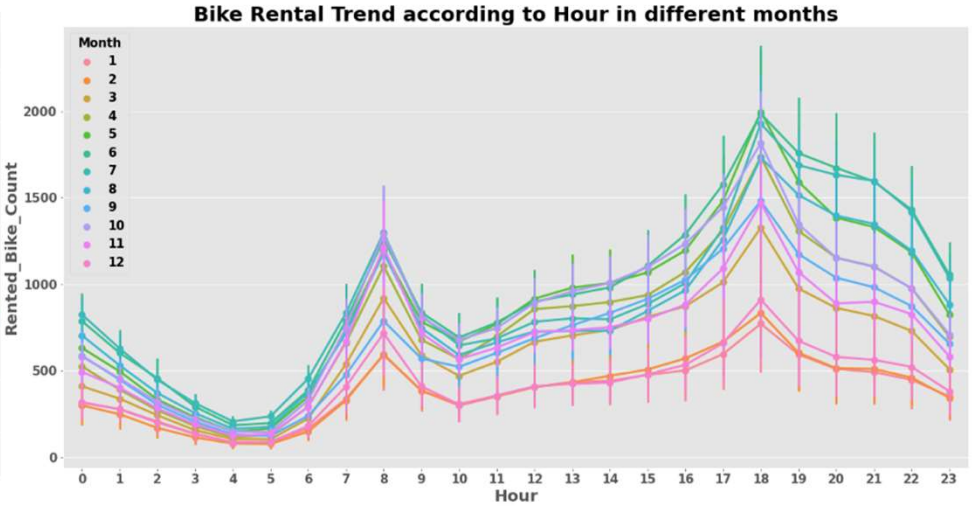


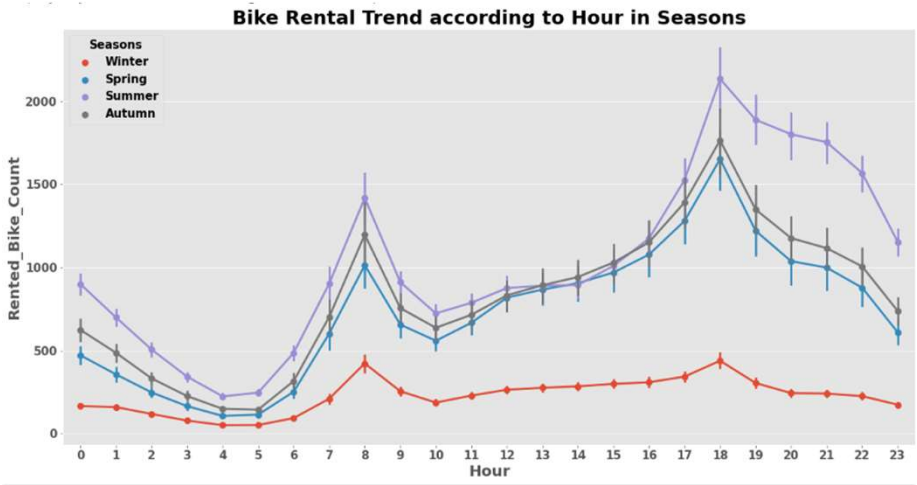


- Here we observed that, Bike rental trend according to hours is almost similar in all scenarios.
- There is sudden peak between 6/7AM to 10 AM. Office /College going time could be the reason for this sudden peak on NO Holiday. But on Holiday the case is different,very less bike rentals happened.
- Again there is peak between 4PM to 7 PM. may be its office leaving time for the above people.(NO Holiday).
- Here the trend for functioning day is same as of No holiday. Only the difference is on No functioning day there were zero bike rentals

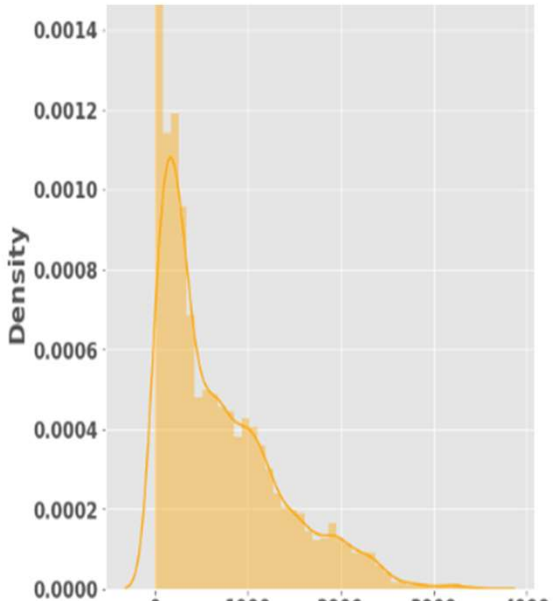
EDA (Exploratory Data Analysis):

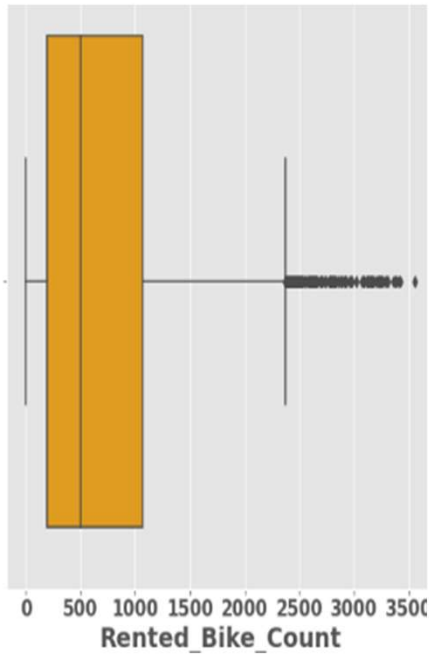


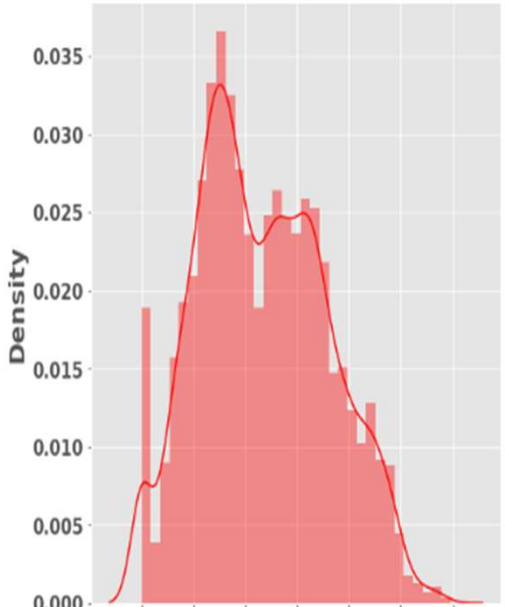


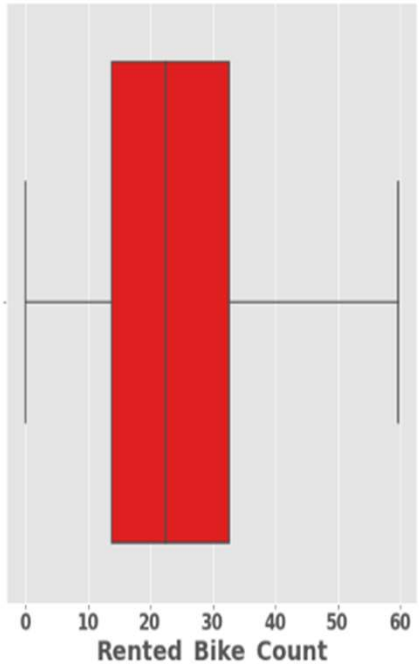


EDA (Exploratory Data Analysis):





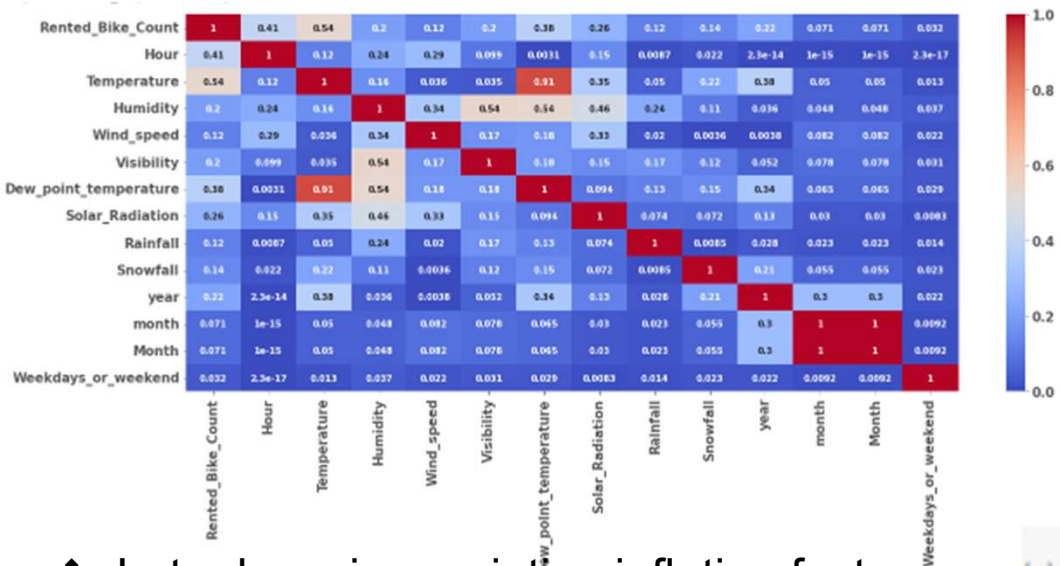




Distribution is rightly skewed and some outliers are observed.

To normalize the distribution we applied square root method. After normalization no outliers were found.

Preparation of data for model building:



- With the heat map we dropped highly correlated variables. As we can see Temperature and Dew point temperature are 91 % correlated. So we dropped the Dew point temperature because it has very low correlation with our target variable as compared to temperature.

- Later by using variation inflation factor we dropped ‘Visibility’ and ‘Humidity’ features as they had VIF value more than 5.
- Next we created dummy variables for categorical Seasons column and did mapping with 0 and 1 for holiday and functioning column.
- Thus we prepared our data for model building

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

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

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

Model Selection and Evaluation:

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

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

Assumptions of regression line:

1.The relation between the dependent and independent variables should be almost linear.

2.Mean of residuals should be zero or close to 0 as much as possible. It is done to check whether our line is actually the line of “best fit”.

3.There should be homoscedasticity or equal variance in a regression model. This assumption means that the variance around the regression line is the same for all values of the predictor variable (X).

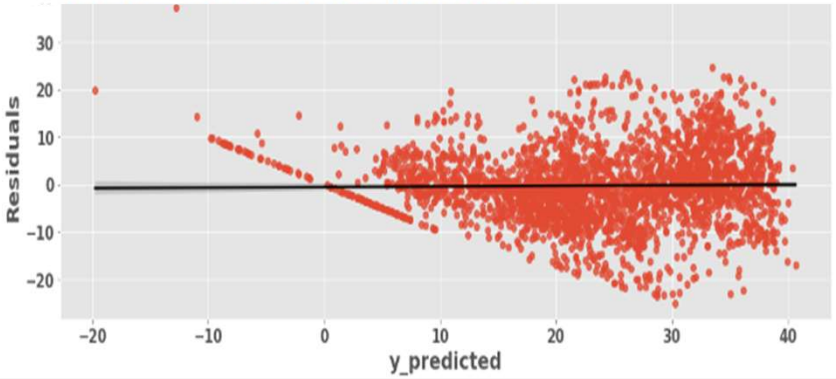
4.There should not be multicollinearity in regression model. Multicollinearity generally occurs when there are high correlations between two or more independent variables

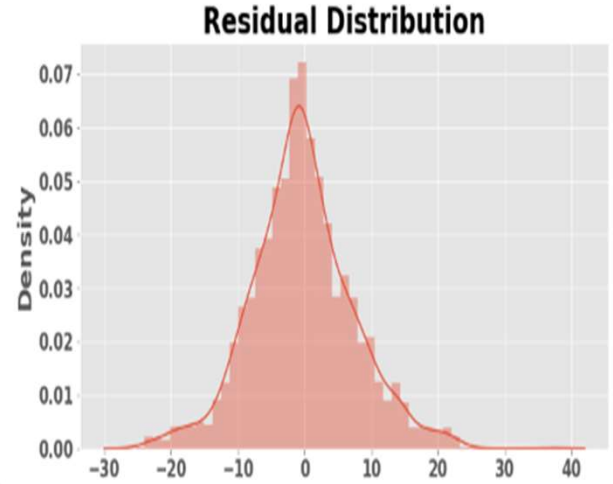
Model Selection and Evaluation :

Linear regression, Lasso and Ridge Regression:

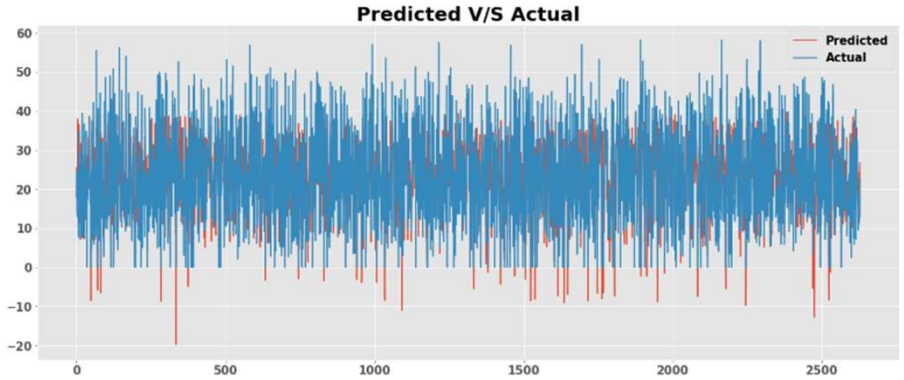
Linear Regression -Scores on Train set

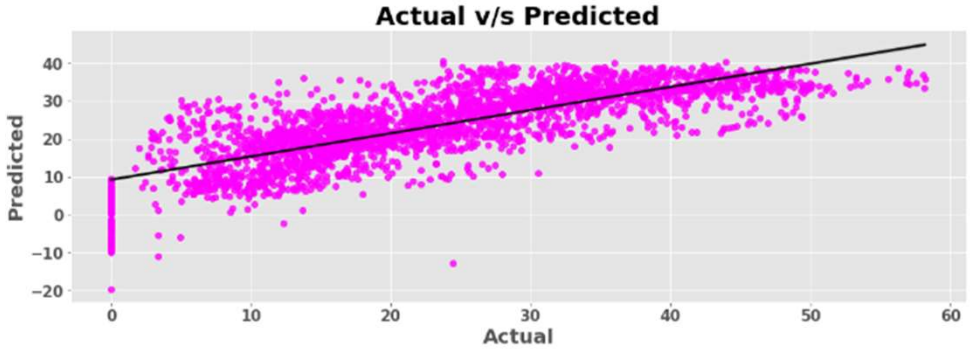
```
The Mean Absolute Error (MAE) is 5.8469961806347905.
The Mean Squred Error(MSE) is 60.23719291606169.
The Root Mean Squared Error(RMSE) is 7.761262327486533.
The R2 Score is 0.6127533164602634.
```





Mean of residuals should be zero or close to 0 as much as possible. It is done to check whether our line is actually the line of “best fit”



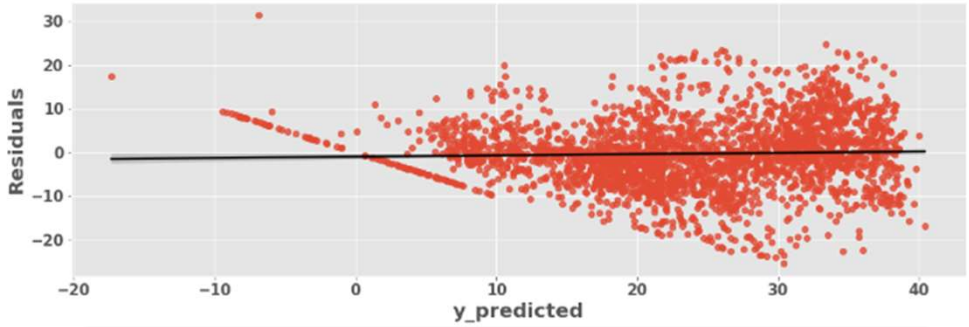


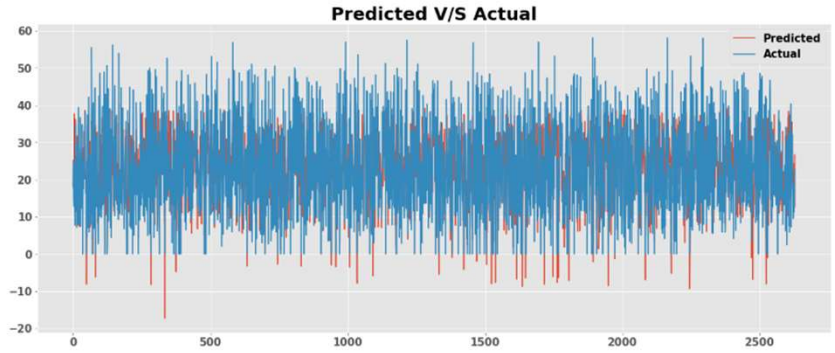
Model Selection and Evaluation :

Lasso (Hyper-parameter tuned- alpha=0.01)

Scores on Train set-

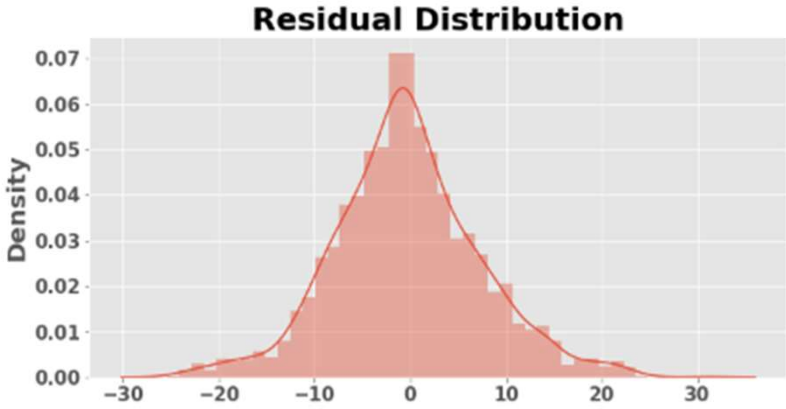
```
The Mean Absolute Error (MAE) is 5.863101271707961.
The Mean Squred Error(MSE) is 60.40857006397928.
The Root Mean Squared Error(RMSE) is 7.772295031969597.
The R2 Score is 0.6116515846405541.
```

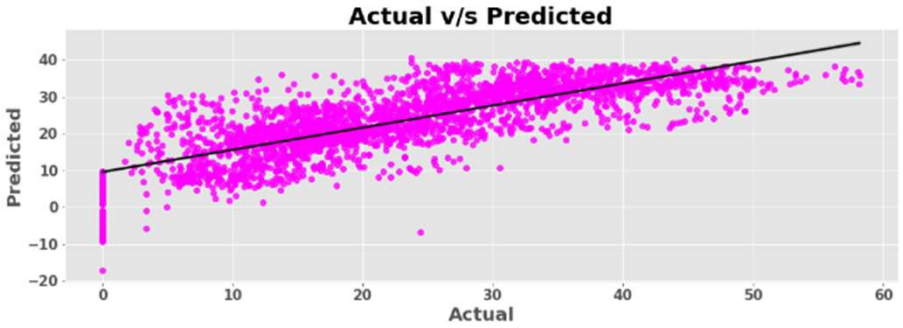




Scores on Test set-

```
The Mean Absolute Error (MAE) is 5.841333872852871.
The Mean Squred Error(MSE) is 58.71259160201371.
The Root Mean Squared Error(RMSE) is 7.662414214985621.
The R2 Score is 0.6177518014800556.
```



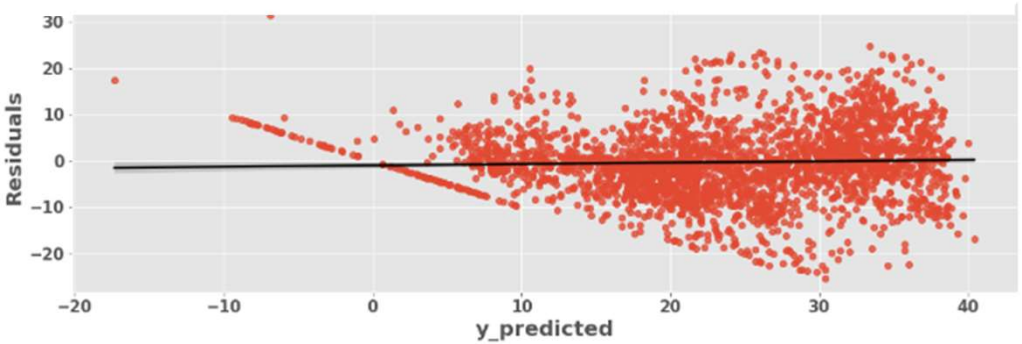


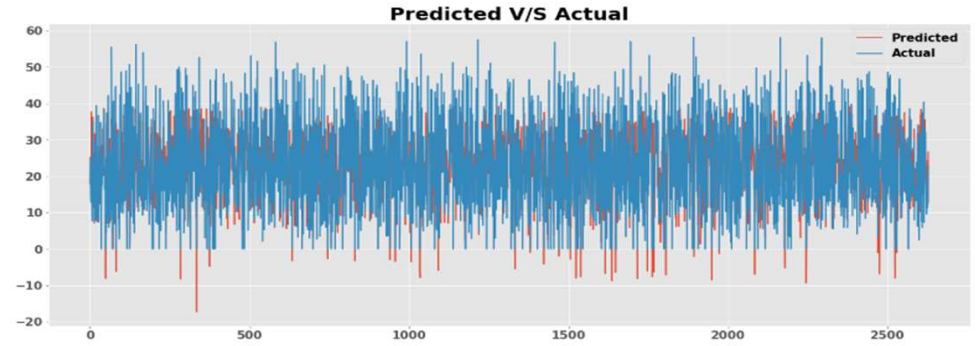
Model Selection and Evaluation :

Ridge (Hyper-parameter tuned- alpha=0.1)

Scores on Train set-

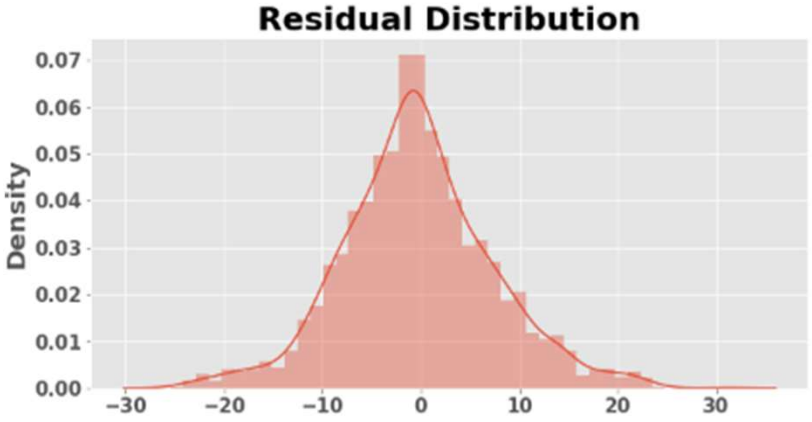
```
The Mean Absolute Error (MAE) is 5.863101271707961.
The Mean Squred Error(MSE) is 60.40857006397928.
The Root Mean Squared Error(RMSE) is 7.772295031969597.
The R2 Score is 0.6116515846405541.
```

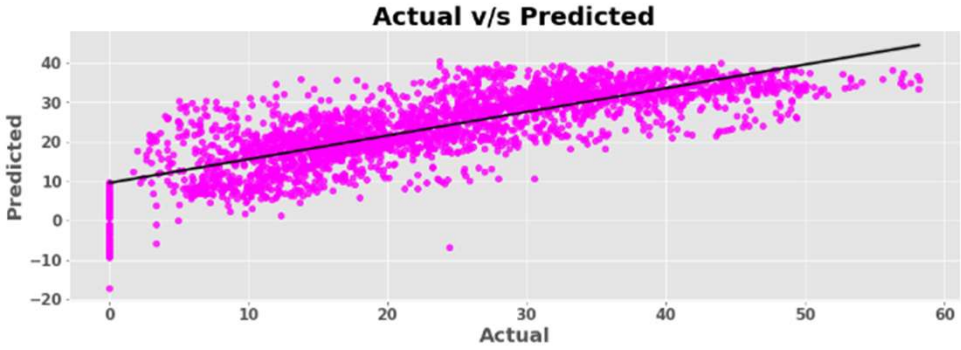




Scores on Test set-

```
The Mean Absolute Error (MAE) is 5.841333872852871.
The Mean Squred Error(MSE) is 58.71259160201371.
The Root Mean Squared Error(RMSE) is 7.662414214985621.
The R2 Score is 0.6177518014800556.
```





Model Selection and Evaluation :

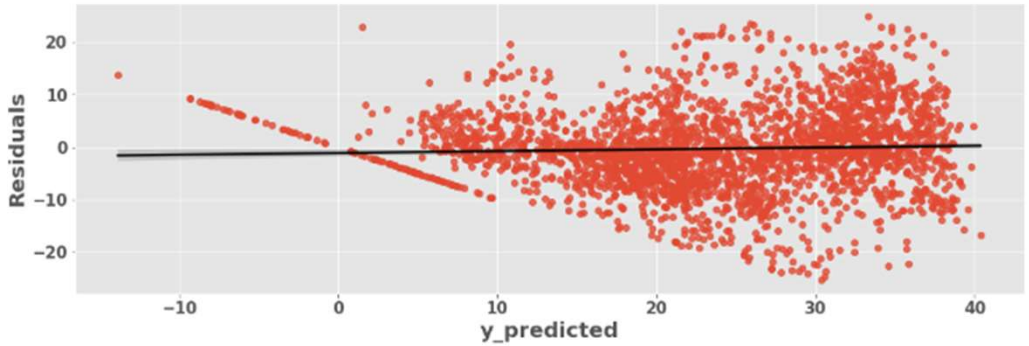
Elastic Net (Hyper-parameter tuned- alpha=0.001,l1_ratio=0.5)

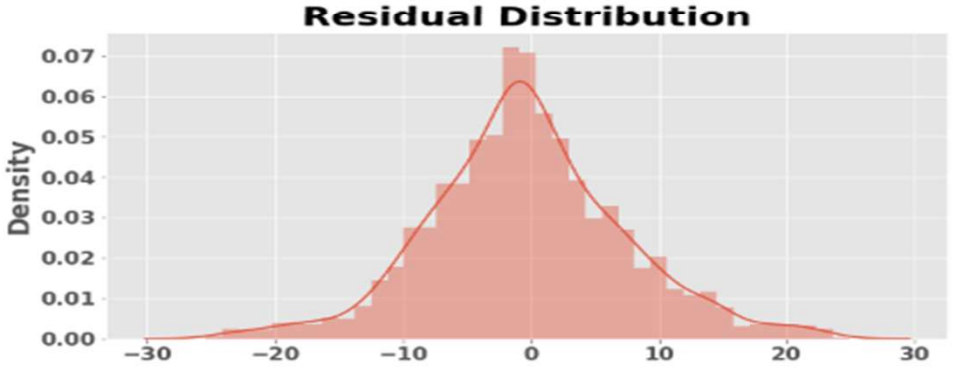
Scores on Train set-

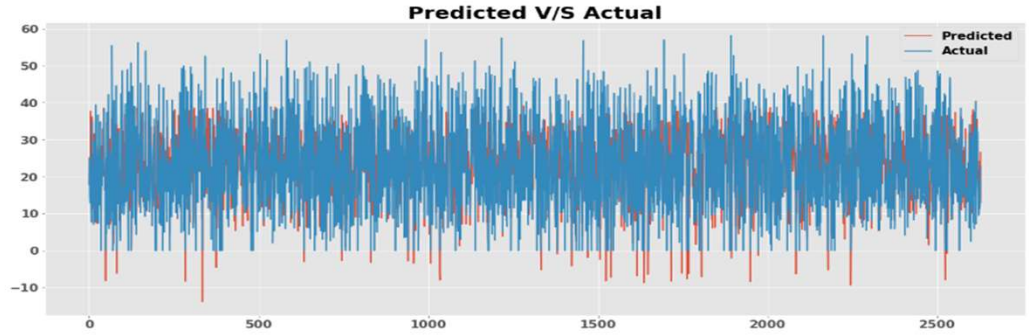
```
The Mean Absolute Error (MAE) is 5.8852560792103095.
The Mean Squred Error(MSE) is 60.84327535936255.
The Root Mean Squared Error(RMSE) is 7.80020995610775
The R2 Score is 0.6088569958523796.
```

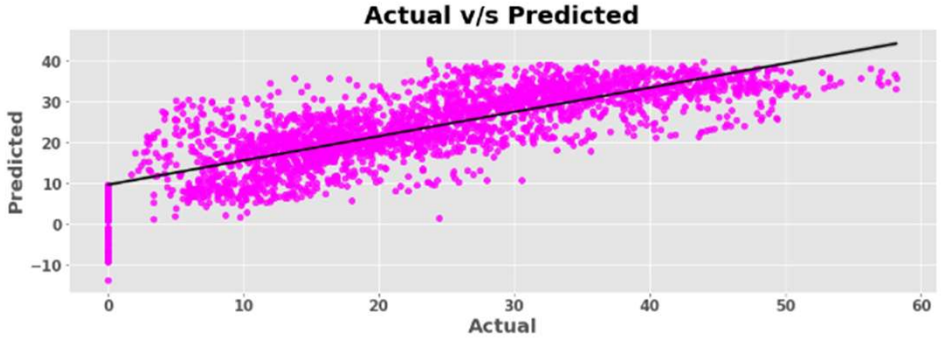
Scores on Test set-

```
The Mean Absolute Error (MAE) is 5.860710926662991.
The Mean Squred Error(MSE) is 59.19453794315168.
The Root Mean Squared Error(RMSE) is 7.69379866796315
The R2 Score is 0.6146140908858437.
```







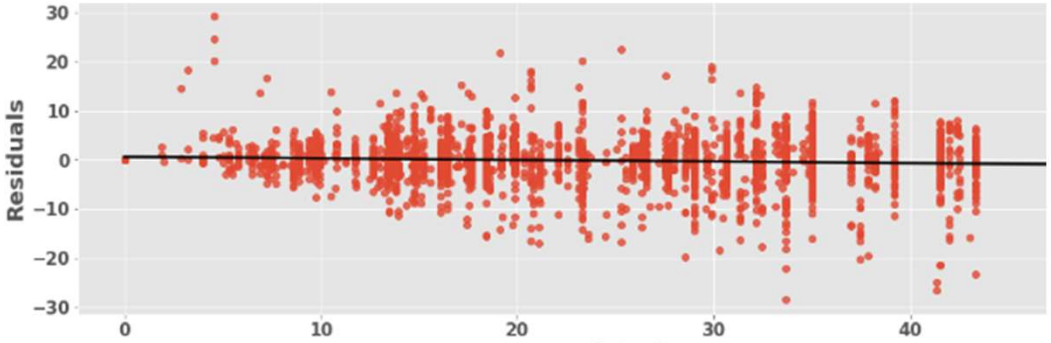


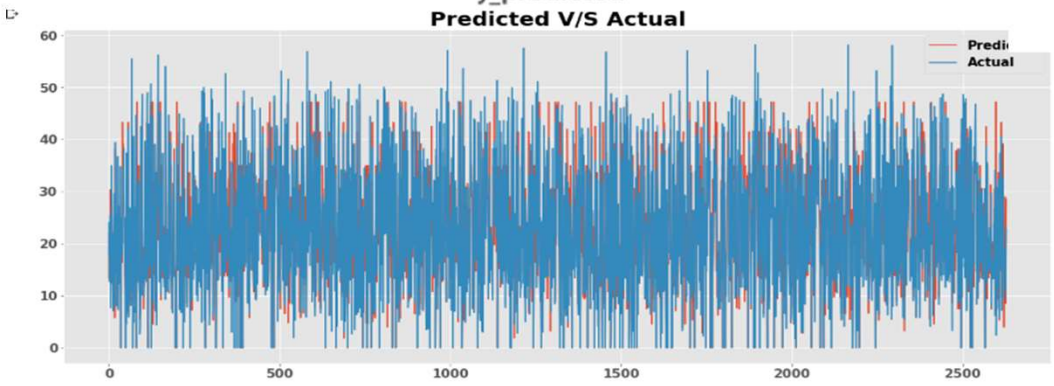
Model Selection and Evaluation :

Decision Tree regression(Hyper-parameter tuned- max_depth=9,max_features='auto')

Scores on Train set-

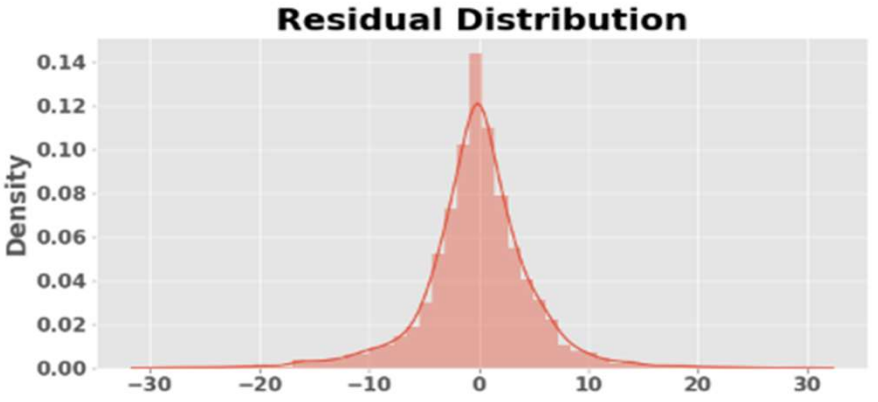
```
The Mean Absolute Error (MAE) is 2.88854118515446.
The Mean Squred Error(MSE) is 18.44510864869925.
The Root Mean Squared Error(RMSE) is 4.294776903251116.
The R2 Score is 0.8814219785823663.
```

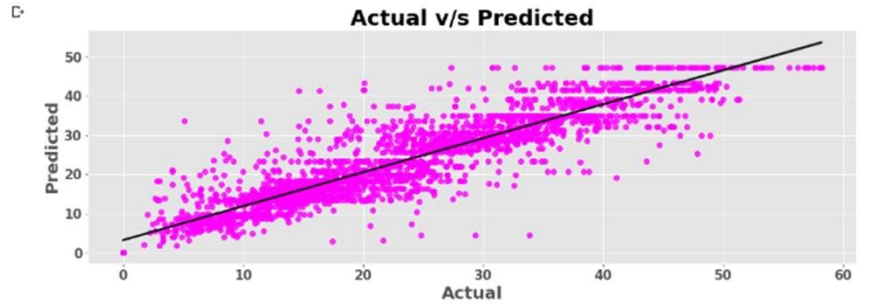




Scores on Test set-

```
The Mean Absolute Error (MAE) is 3.3978326228703413.
The Mean Squred Error(MSE) is 24.87155380927006.
The Root Mean Squared Error(RMSE) is 4.9871388399833085.
The R2 Score is 0.8380738036155898.
```





Model Selection and Evaluation :

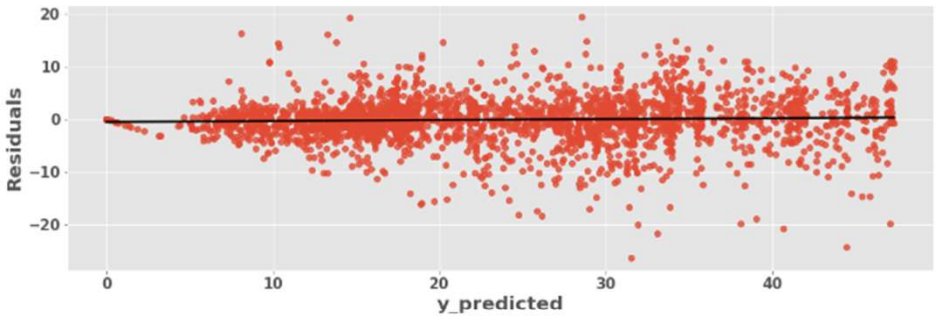
Random forest regression(Hyper-parameter tuned- 'max_depth': 9, 'n_estimators': 100')

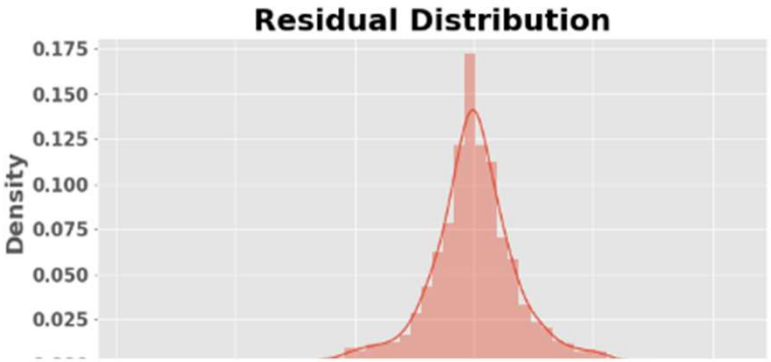
Scores on Train set-

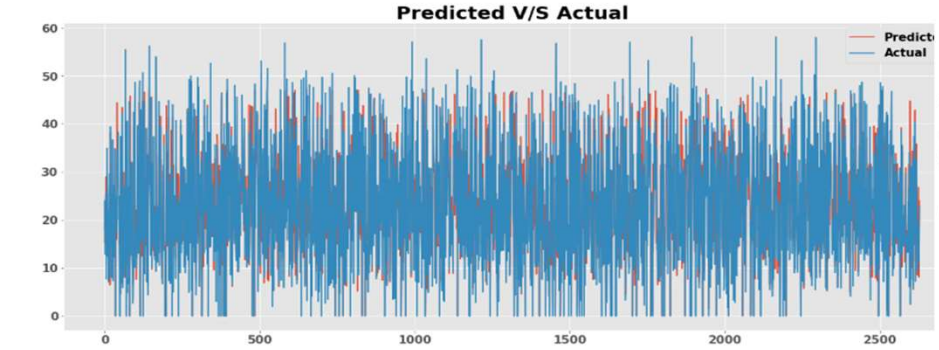
```
The Mean Absolute Error (MAE) is 2.6103364057787903.
The Mean Squred Error(MSE) is 14.729473483991933.
The Root Mean Squared Error(RMSE) is 3.837899618800879.
The R2 Score is 0.905308672585215.
```

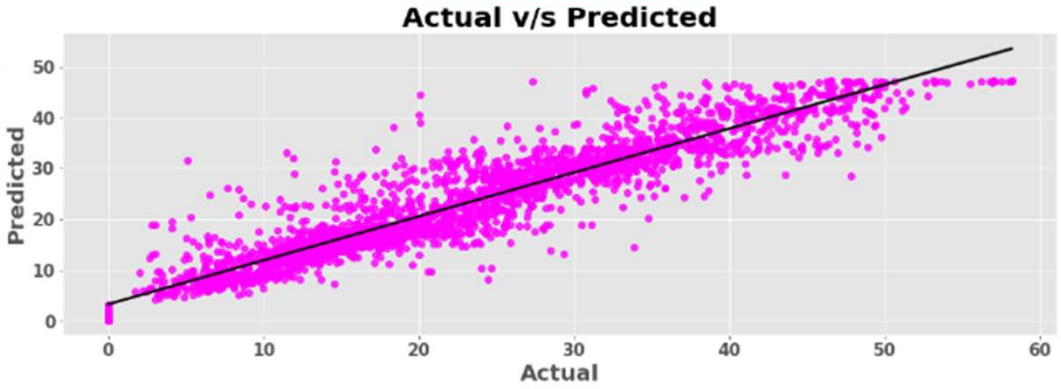
Scores on Test set-

```
The Mean Absolute Error (MAE) is 2.943172641496349.
The Mean Squred Error(MSE) is 18.63616054504846.
The Root Mean Squared Error(RMSE) is 4.31696195779491.
The R2 Score is 0.878669317751102.
```









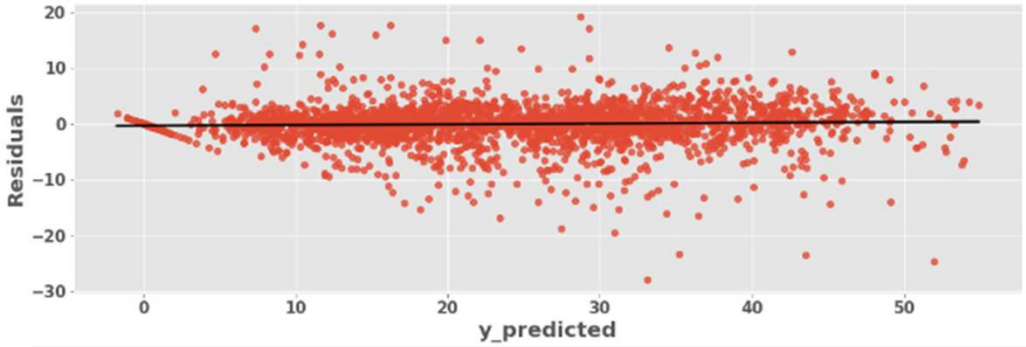
Model Selection and Evaluation :

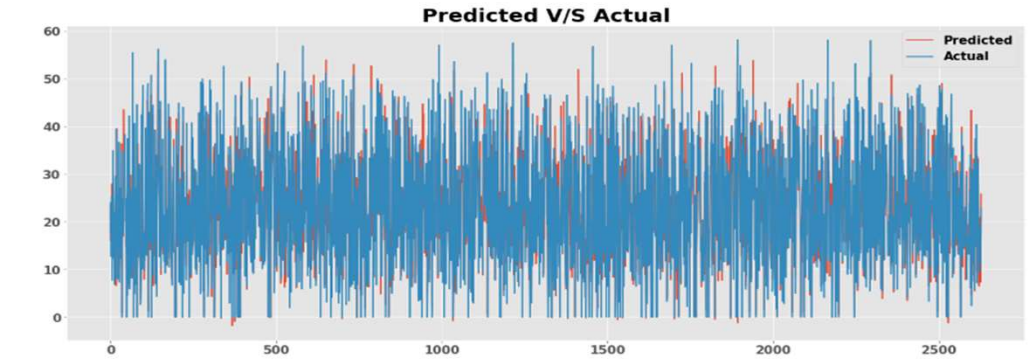
Gradient boosting regression(Hyper-parameter tuned-

'learning_rate': 0.04, 'max_depth': 8, 'n_estimators': 150, 'subsample':

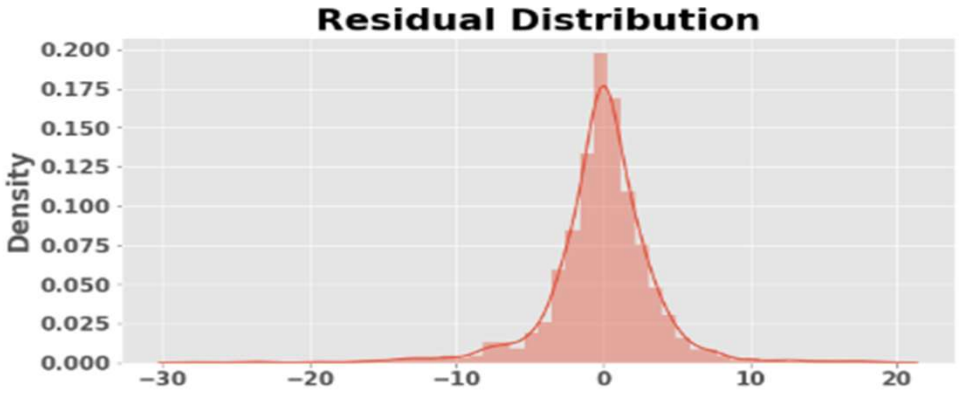
Scores on Train set-

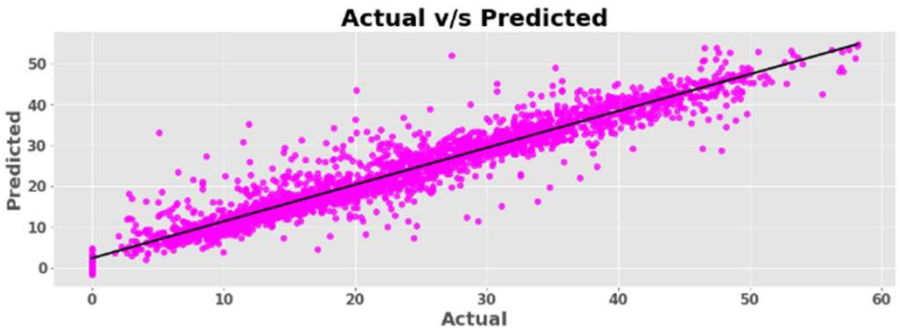
```
The Mean Absolute Error (MAE) is 1.492428024177122.
The Mean Squred Error(MSE) is 4.767500128611065.
The Root Mean Squared Error(RMSE) is 2.1834605855410043.
The R2 Score is 0.9693511844724818.
```






```
The Mean Absolute Error (MAE) is 2.3607037184077466.
The Mean Squred Error(MSE) is 13.171727089599313.
The Root Mean Squared Error(RMSE) is 3.629287407963072.
The R2 Score is 0.914245499746889.
```





Conclusion:

		Model	MAE	MSE	RMSE	R2_score
Training set	0	Linear Regression	5.8470	60.2372	7.7613	0.6128
	1	Lasoo	5.8631	60.4086	7.7723	0.6117
	2	Ridge GridSearchCV	5.8631	60.4086	7.7723	0.6117
	3	ElasticNet(GridSearchCV-Tunned)	5.8853	60.8433	7.8002	0.6089
	4	Decision Tree Regressor-GridSearchCV	2.8885	18.4451	4.2948	0.8814
	5	Random Forest	0.9296	2.1314	1.4599	0.9863
	6	Random Forest-GridSearchCv	2.6103	14.7295	3.8379	0.9053
	7	Gardient boosting Regression	3.1467	20.2872	4.5041	0.8696
	8	Gradient Boosting Regression(GridSearchCV)	1.4924	4.7675	2.1835	0.9694
Test set	0	Linear Regression	5.8234	58.5218	7.6500	0.6190
	1	Lasso	5.8413	58.7126	7.6624	0.6178
	2	Ridge(Gridsearchcv Tunned)	5.8413	58.7126	7.6624	0.6178
	3	ElasticNet(GridSearchCV-Tunned)	5.8607	59.1945	7.6938	0.6146
	4	Decision Tree Regressor(GridsearchCV)	3.3978	24.8716	4.9871	0.8381
	5	Radom forest	2.4493	13.9406	3.7337	0.9092
	6	Random Forest-GridSearchCv	2.9432	18.6362	4.3170	0.8787
	7	Gradient Boosting Regression	3.2702	21.5763	4.6450	0.8595
	8	Gradient Boosting Regression(GridSearchCV)	2.3607	13.1717	3.6293	0.9142

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

Decision Tree Regression: On Decision tree regressor model, without hyper - parameter tuning, we got r2 score as 100% on training data and on test data it was very less. Thus our model memorized the data. So it was a over fitted model. After hyper -parameter tuning we got r2 score as 88% on training data and 83% on test data which is quite good for us.

Conclusion:

Random Forest:

On Random Forest regressor model, without hyper -parameter tuning we got r2 score as 98% on training data and 90% on test data. Thus our model memorized the data. So it was a over fitted model, as per our assumption After hyper -parameter tuning we got r2 score as 90% on training data and 87% on test data which is very good for us.

Gradient Boosting Regression(Gradient Boosting Machine):

On Random Forest regressor model, without hyper -parameter tuning we got r2 score as 86% on training data and 85% on test data. Our model performed well without hyper -parameter tuning. After hyper -parameter tuning we got r2 score as 96% on training data and 91% on test data, thus we improved the model performance by hyper -parameter tuning.

Thank You