Capstone Project -2

Bike Sharing Demand Prediction

By - Nirmesh Gupta

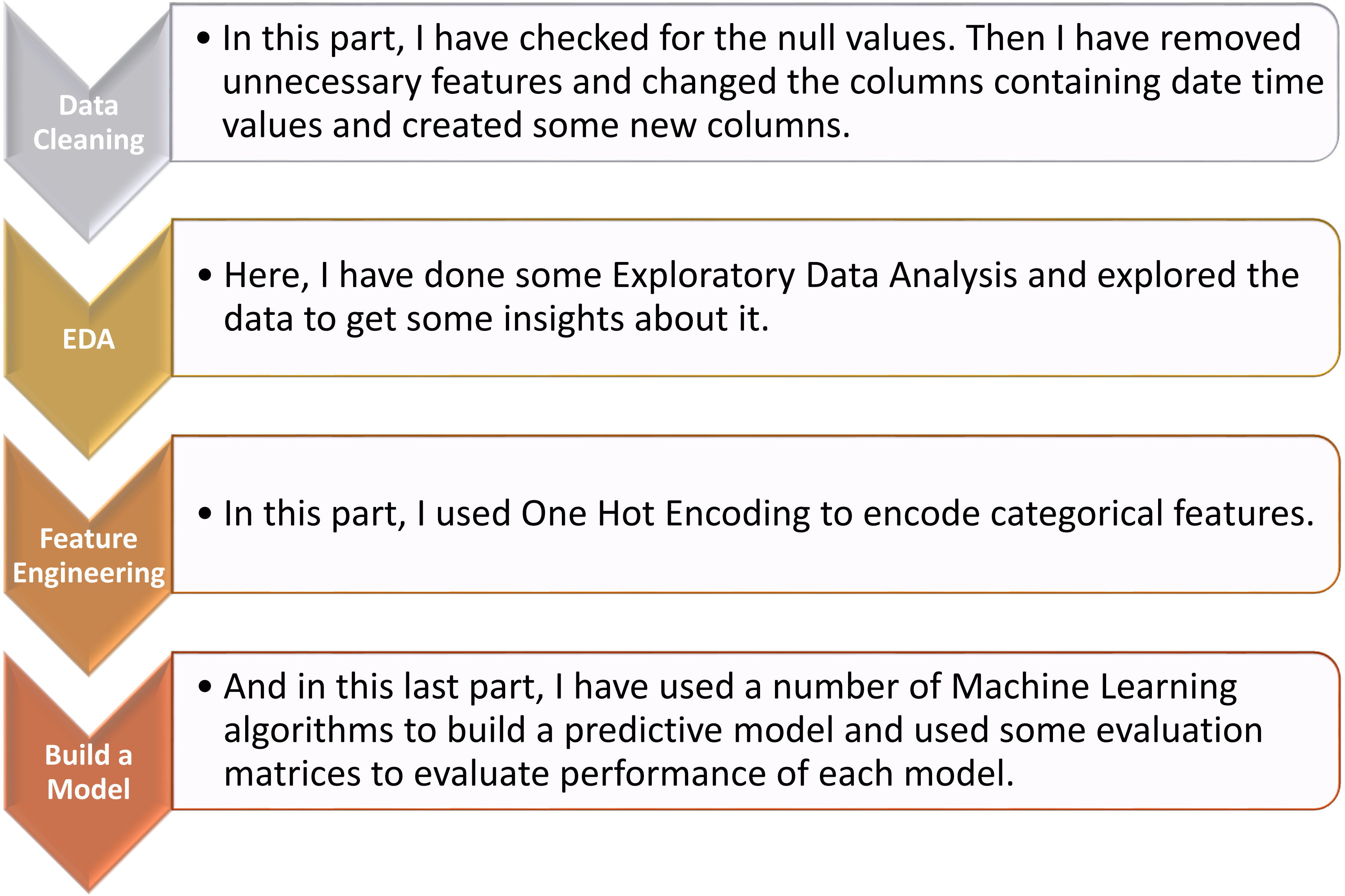
Points For Discussion

- Defining Problem Statement
- Data Summary
- Exploratory Data Analysis
- Feature Engineering
- Preparing Dataset for Modeling
- Applying Models
- Model Evaluation & Validation
- Model Selection

Problem Statement

- Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bikes 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.
- We are tasked with predicting the number of bikes rented each hour so as to make an approximate estimation of the number of bikes to be made available to the public given a particular hour of the day.

Data Pipeline



Data Cleaning

- In this part, I have checked for the null values. Then I have removed unnecessary features and changed the columns containing date time values and created some new columns.

EDA

- Here, I have done some Exploratory Data Analysis and explored the data to get some insights about it.

Feature Engineering

- In this part, I used One Hot Encoding to encode categorical features.

Build a Model

- And in this last part, I have used a number of Machine Learning algorithms to build a predictive model and used some evaluation matrices to evaluate performance of each model.

Data Summary

The dataset contains rental data and weather data corresponding to bike renting spread across two years. It has 8760 observations and 13 independent and 1 target feature.

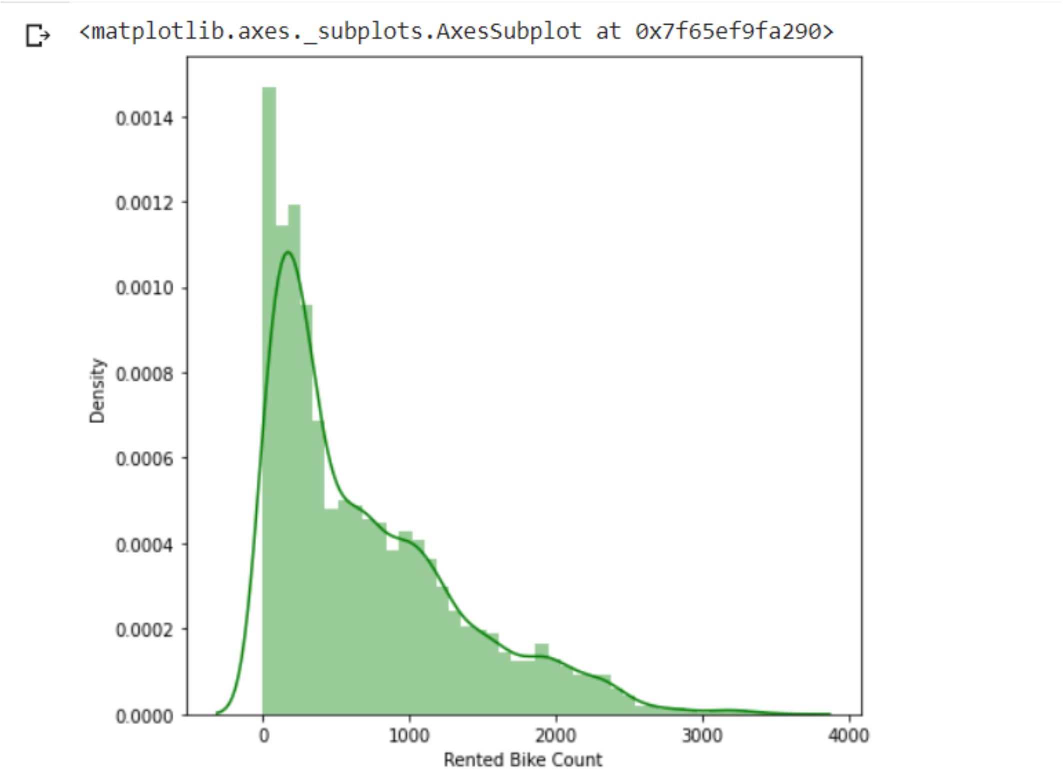
- **Rental Bike count** : This is our Target Feature. It contains information about count of bikes rented at each hour for a period of 365 days.
- **Date** : Date of bike rental.
- **Hour** : Hour of the day of bike rental.
- **Temperature** : Temperature in degree Celsius at the time of bike rental.
- **Humidity** : Humidity in percent at the time of bike rental.
- **Wind Speed** : Wind Speed in miles per second at the time of bike rental.
- **Visibility** : Total visibility at the time of bike rental.

Data Summary

- **Dew Point Temperature** : Dew point temperature in degree Celsius at the time of bike rental
- **Solar Radiation** : Solar radiation at the time of bike rental.
- **Rainfall** : Rainfall in millimeter at the time bike rental.
- **Snowfall** : Snowfall at the time of bike rental.
- **Seasons** : Season at the time of bike rental(Winter/Summer/Autumn/Spring)
- **Holiday** : Whether it was a holiday or not when the bike was rented.
- **Functioning Day** : If it was functioning day or not.

Exploratory Data Analysis(EDA)

❖ Analyzing the Target Variable



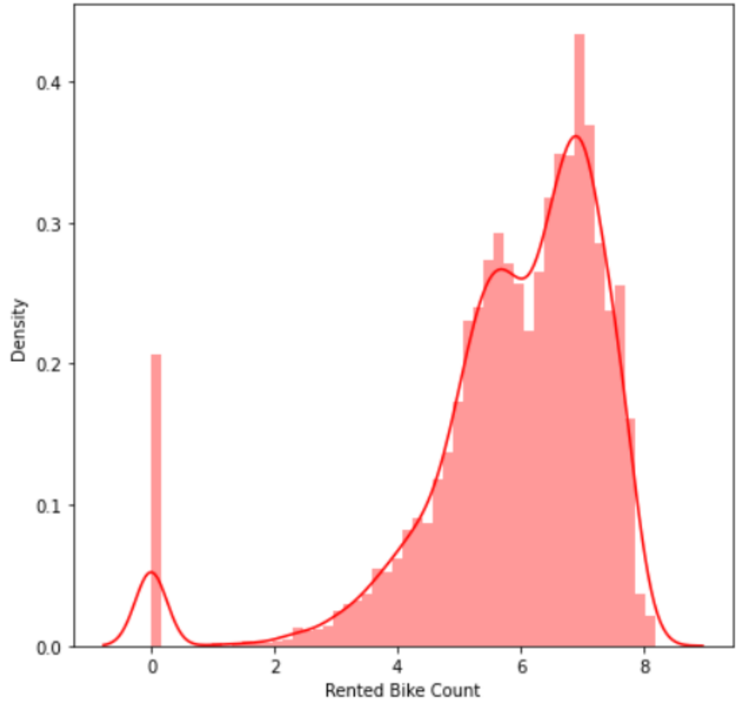
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f65ef9fa290>
```

- This plot shows the distribution of out target feature, which we can see is normally distributed with positive skewness.
- So we will need to apply some transformation over it to treat the skewness.

Analyzing the Target Variable

Log Transformation

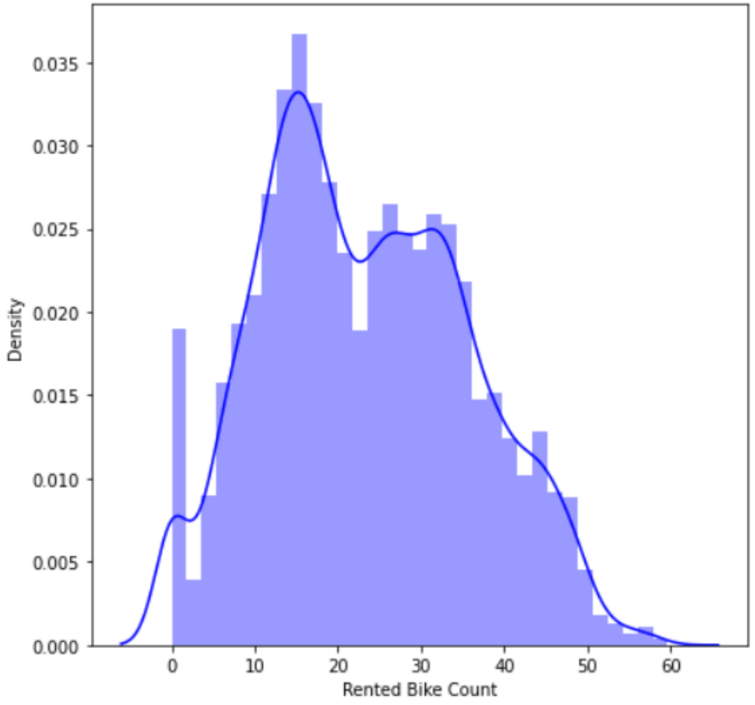
```
Text(0.5, 0, 'Rented Bike Count')
```



First, I tried log transformation on the feature which converted the positive skew to negative skew.

Square root Transformation

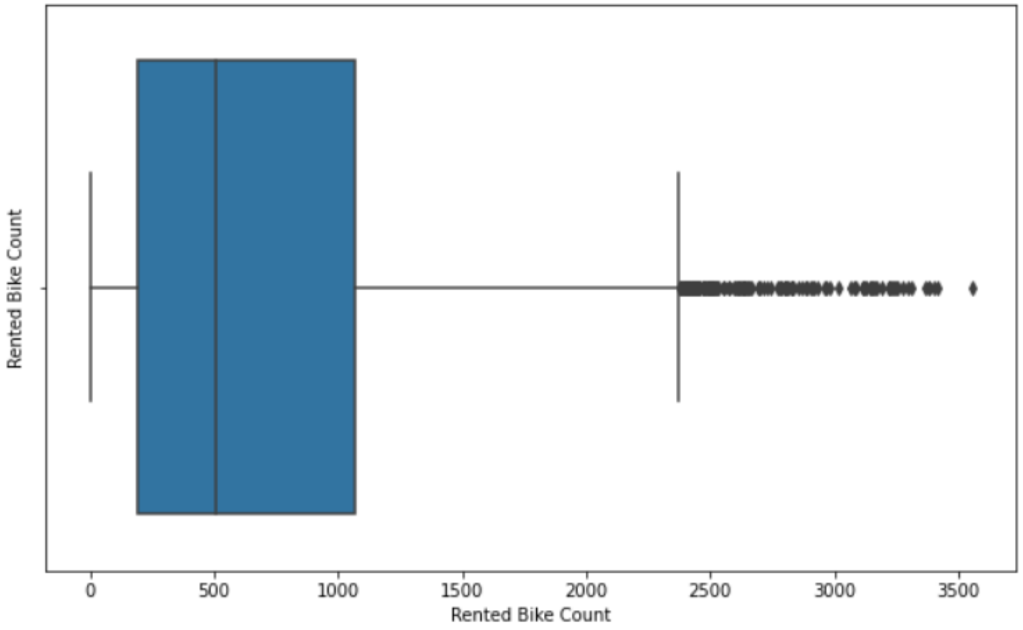
```
Text(0.5, 0, 'Rented Bike Count')
```



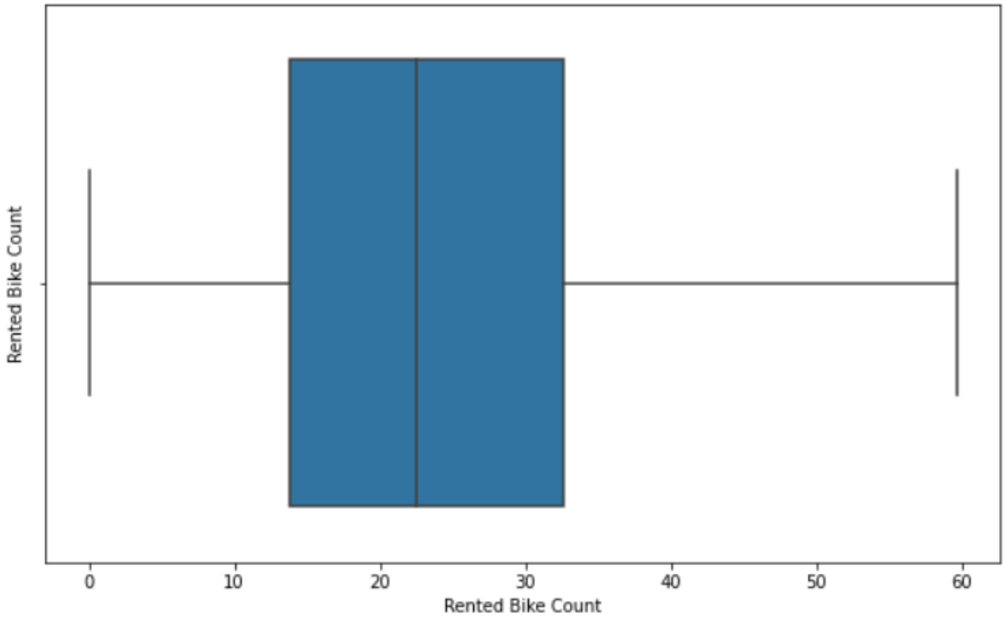
Then I used square root transformation on the feature to make the normal distribution more uniform and remove the skewness.

Analyzing the Target Variable

- ❖ **Rented Bike Count Boxplot before Transformation**

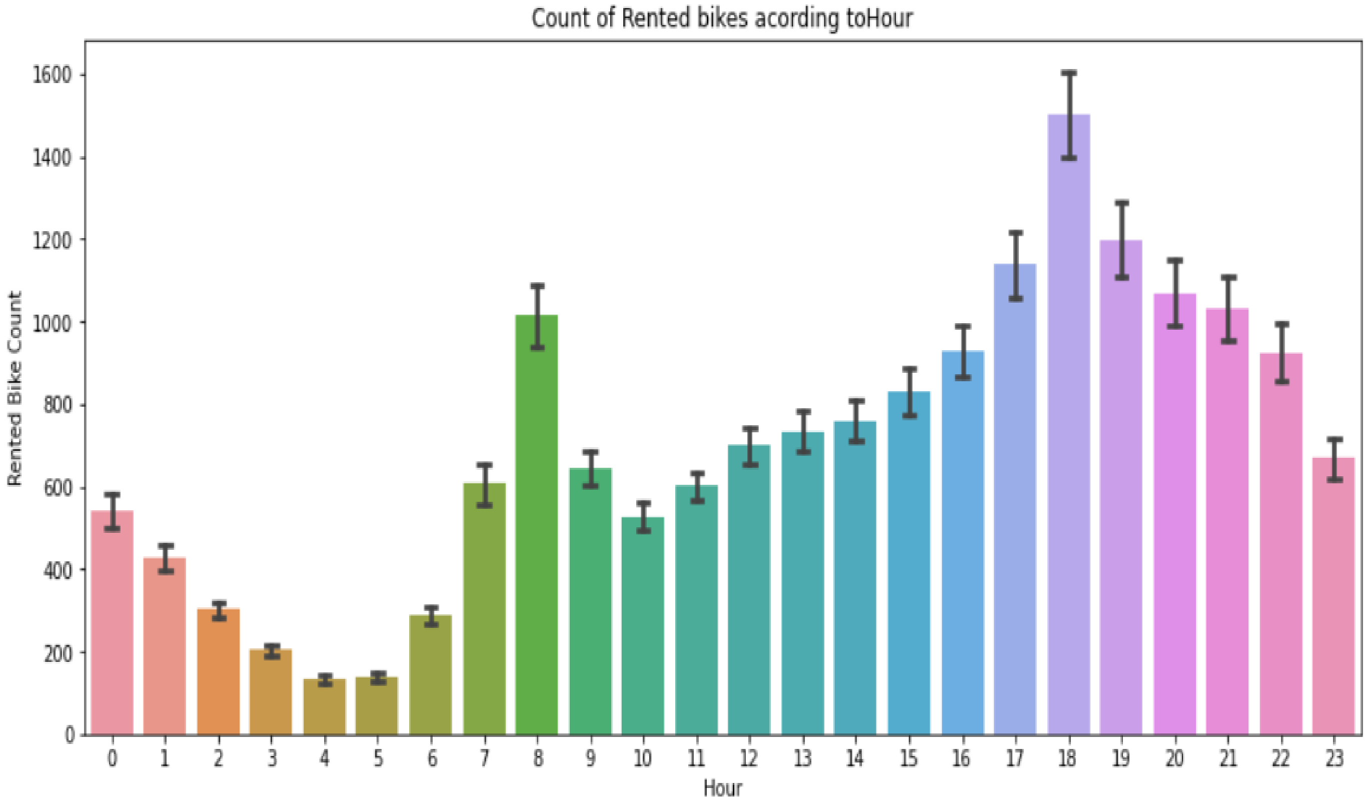


- ❖ **Rented Bike Count Boxplot after Transformation**



Analysis of Categorical Features

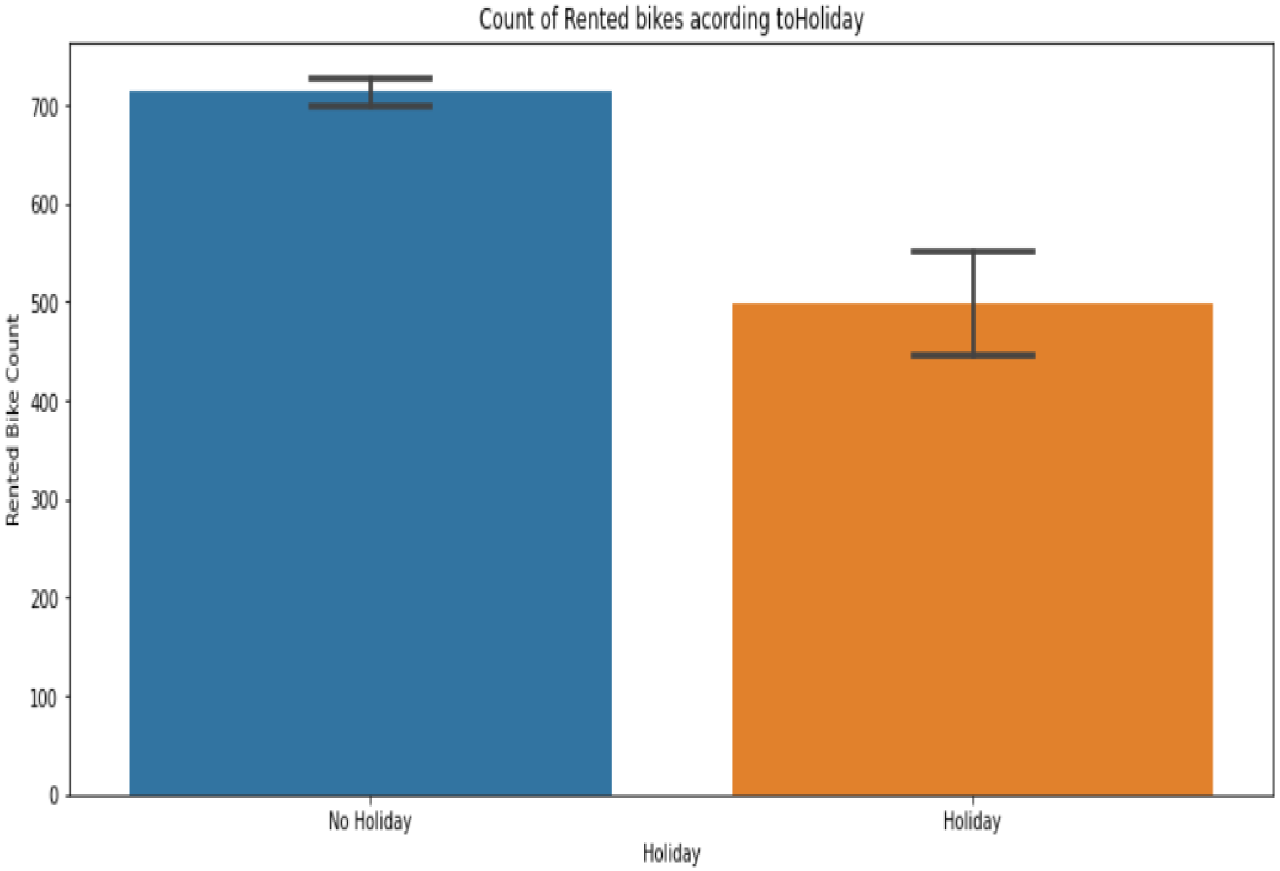
❖ Rented Bike Count vs Hour



- Here, we can see that most of the bookings were made at work going hours (7am to 9am) and during later parts of the day(3pm to 10pm).

Analysis of Categorical Features

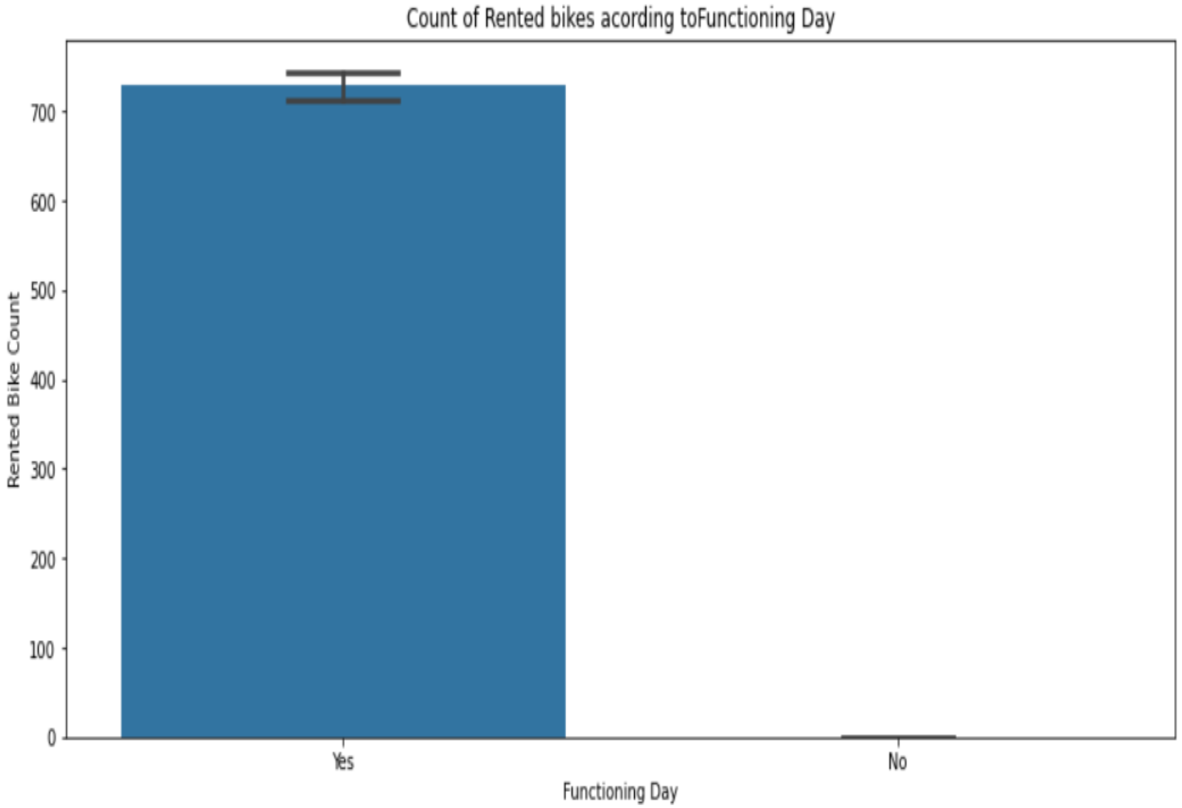
❖ Rented Bike Count vs Holiday



- More number of bikes are booked by people on working days as compared with holiday.

Analysis of Categorical Features

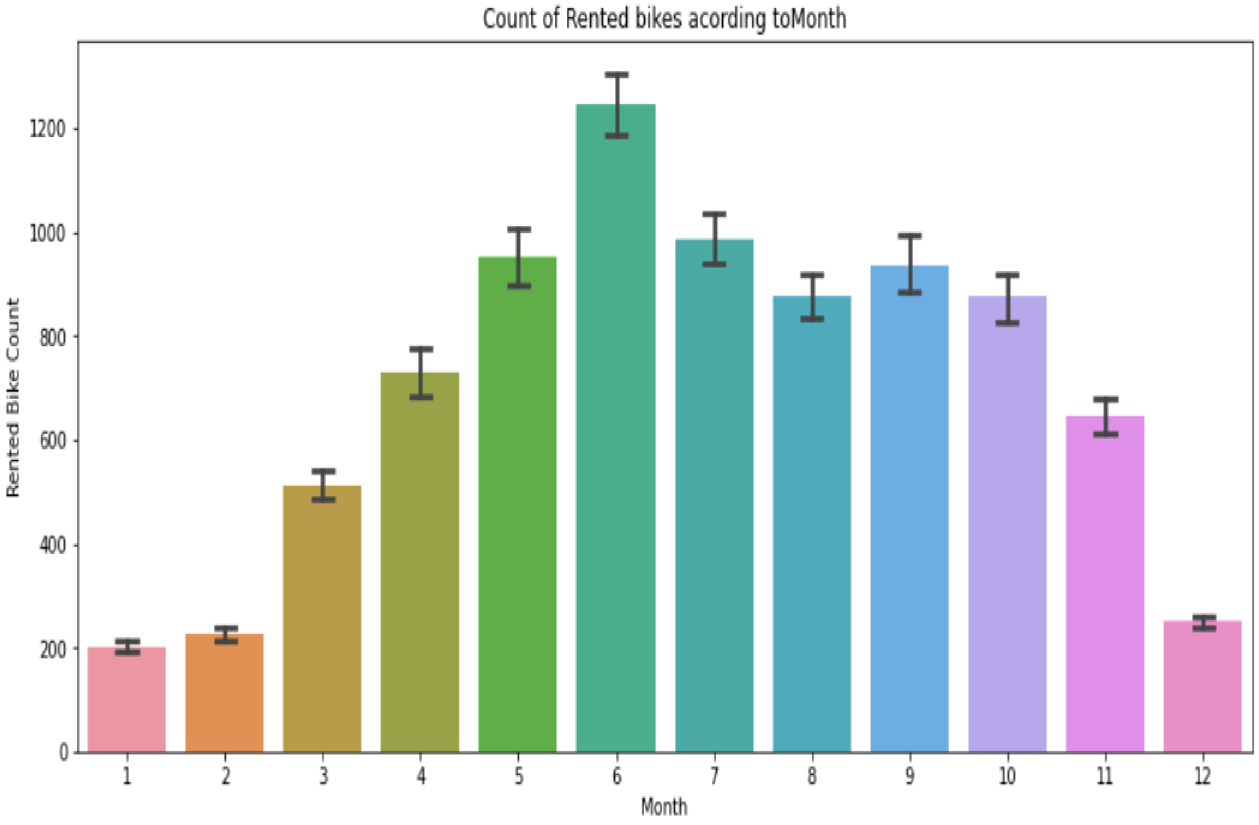
❖ Rented Bike Count vs Functioning Day



- Close to zero percent of bikes are booked on non functioning days.

Analysis of Categorical Features

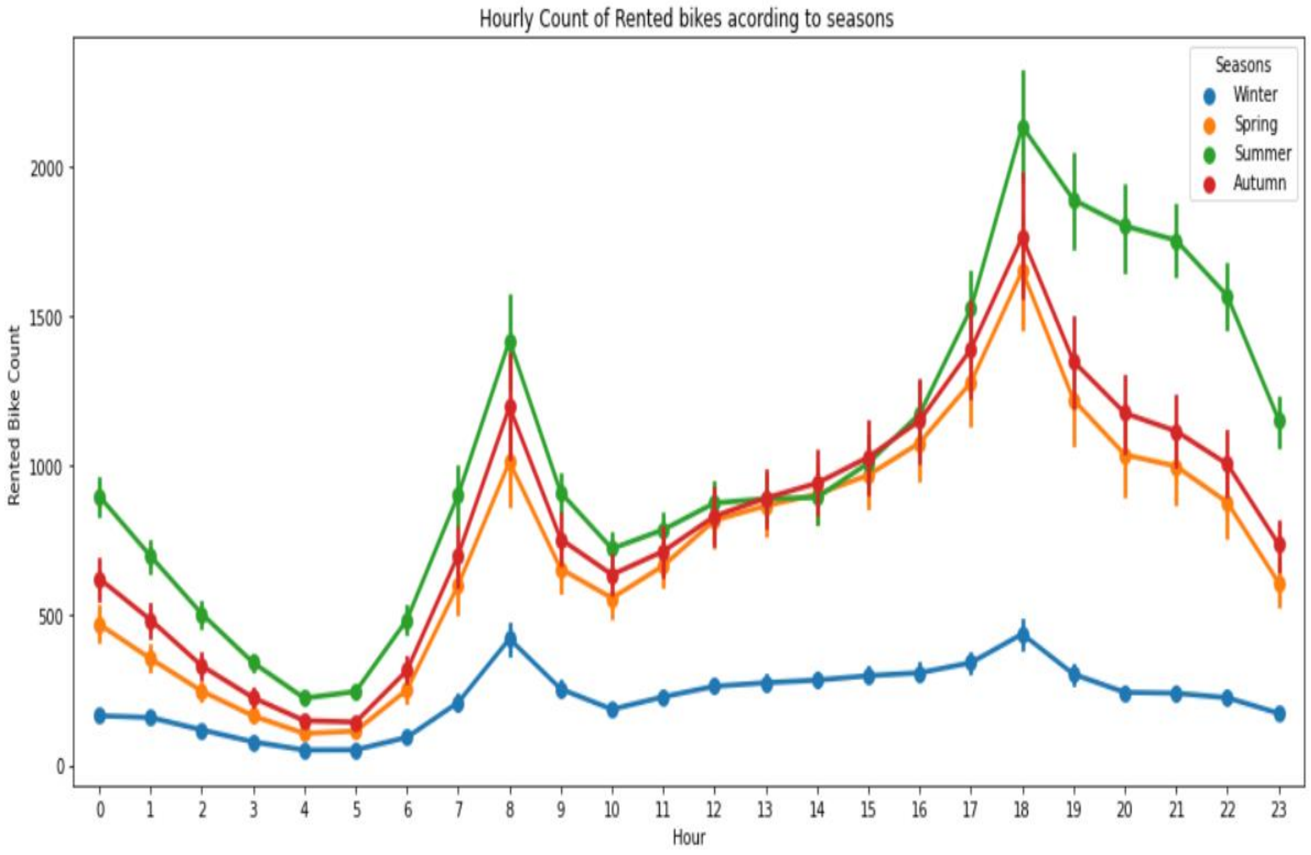
❖ Rented Bike Count vs Month



- As we can see, most of the bookings are made from April to October, which is during summer season.

Analysis of Categorical Features

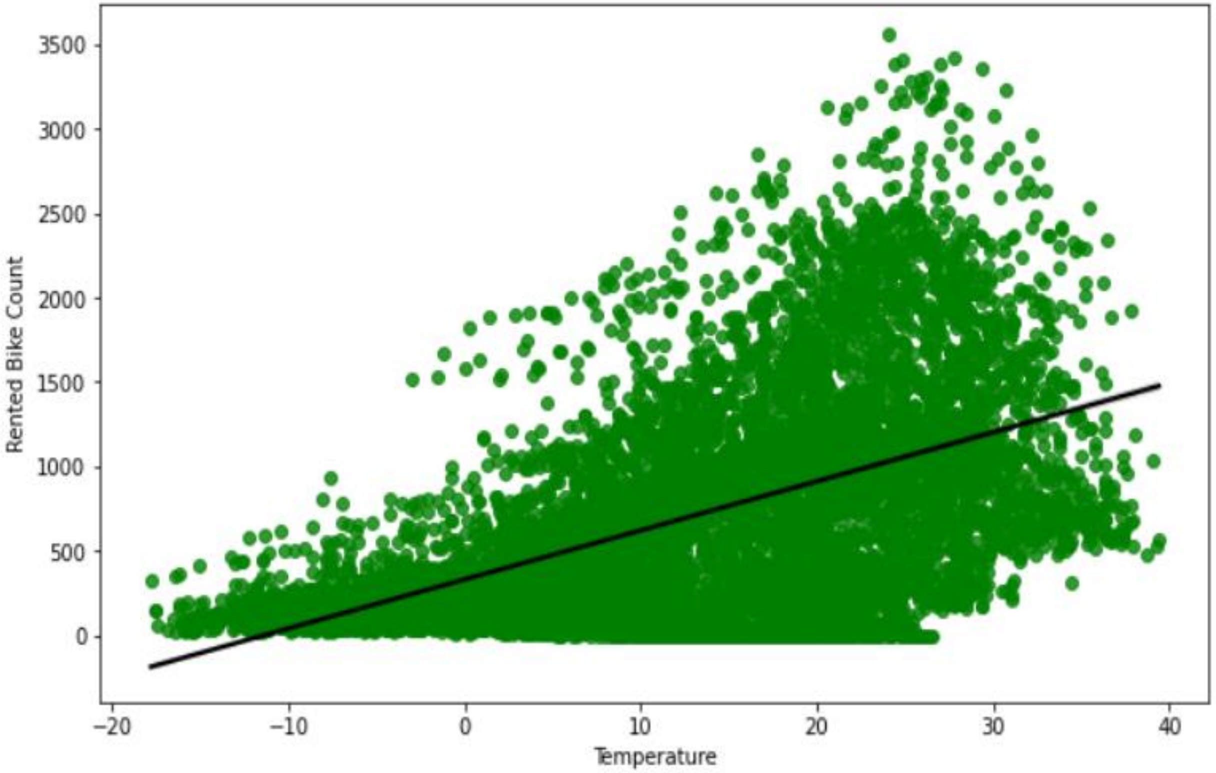
❖ **Hourly count of rented bikes according to Season**



- As we can see, very few bikes are booked in winter season as compared with other seasons.
- So people do not like riding bikes during winter season.

Analysis of Numerical Features

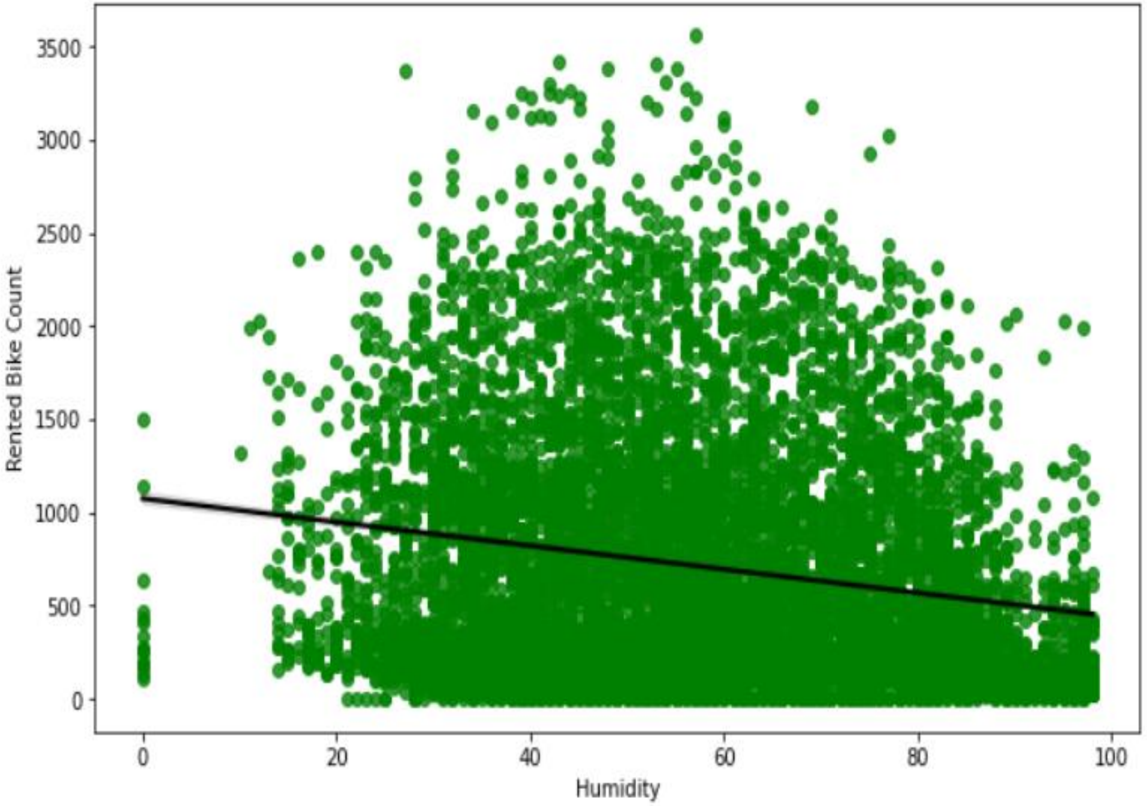
❖ Rented Bike Count vs Temperature



- Here, we can see that Temperature is positively correlated with Rented Bike Count.
- Which again shows that with increasing temperature people prefer riding bikes and at low temperatures bike rentals decrease.

Analysis of Numerical Features

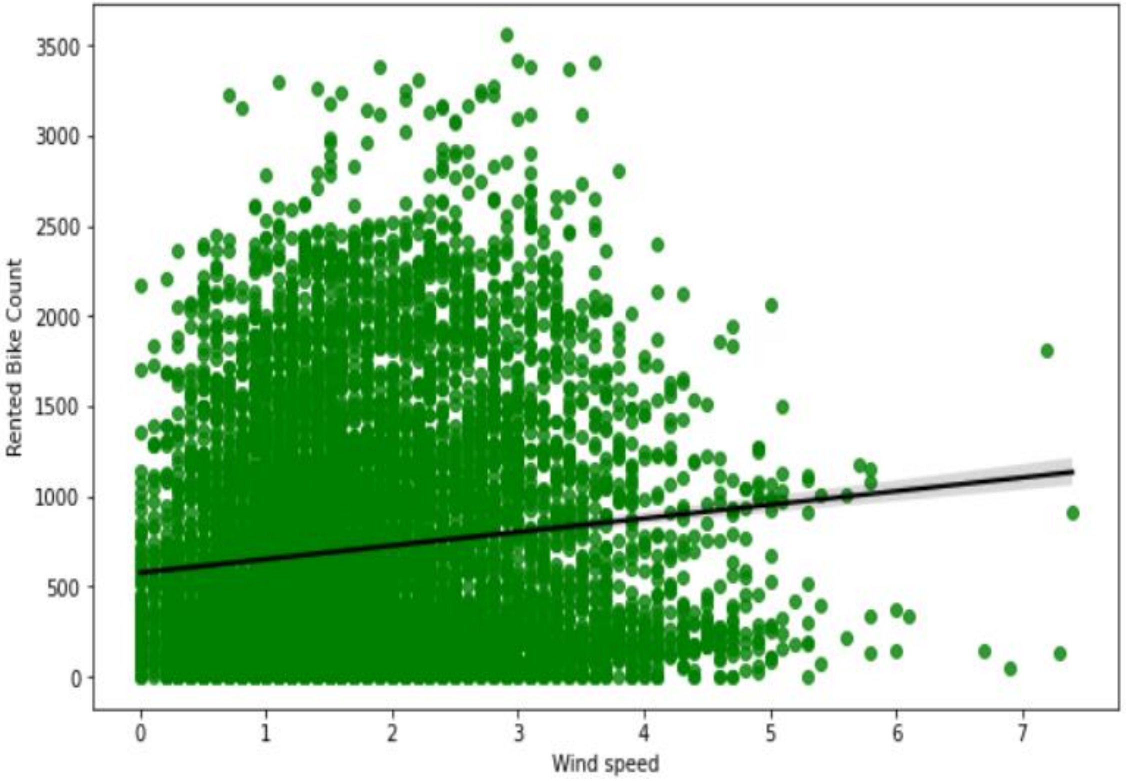
❖ Rented Bike Count vs Humidity



- Humidity is negatively correlated with rented bike count.
- So, when there is more rain and humidity increases, lesser bikes are rented.

Analysis of Numerical Features

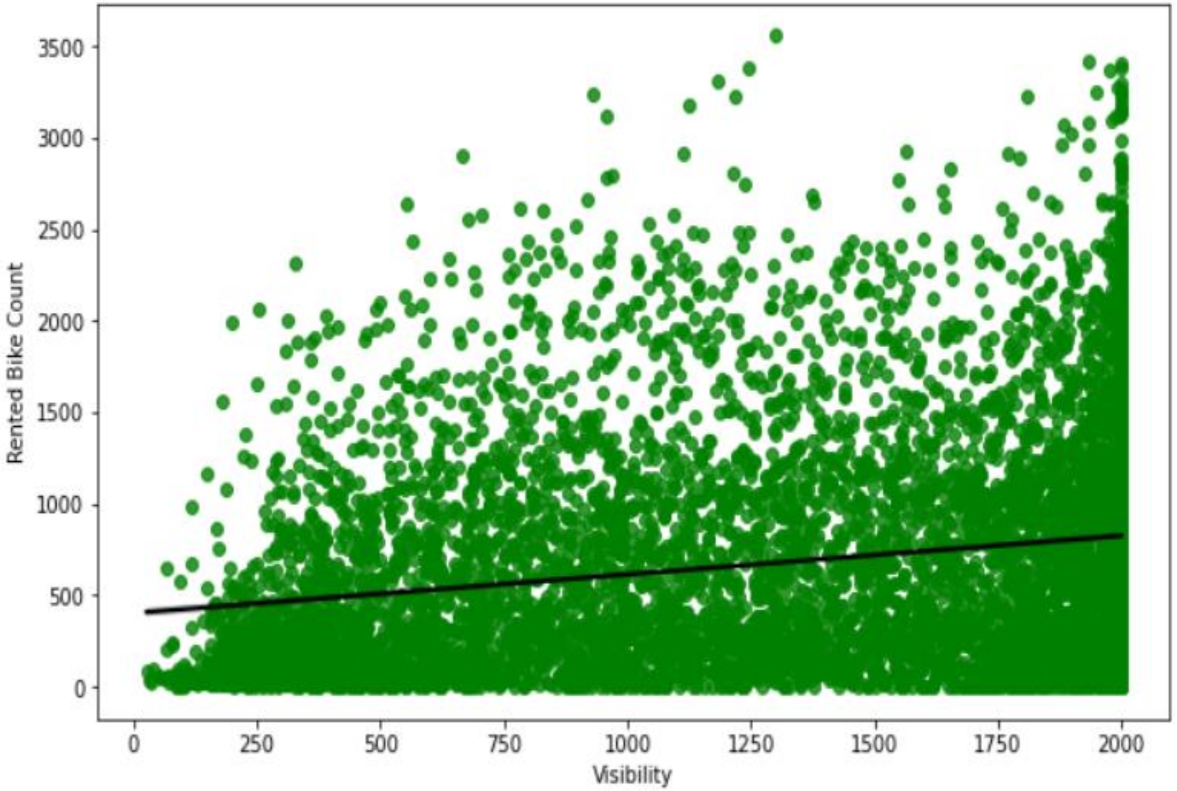
❖ Rented Bike Count vs Wind Speed



- Wind speed is positively correlated with rented bike count.
- People like to ride bikes when it’s windy.

Analysis of Numerical Features

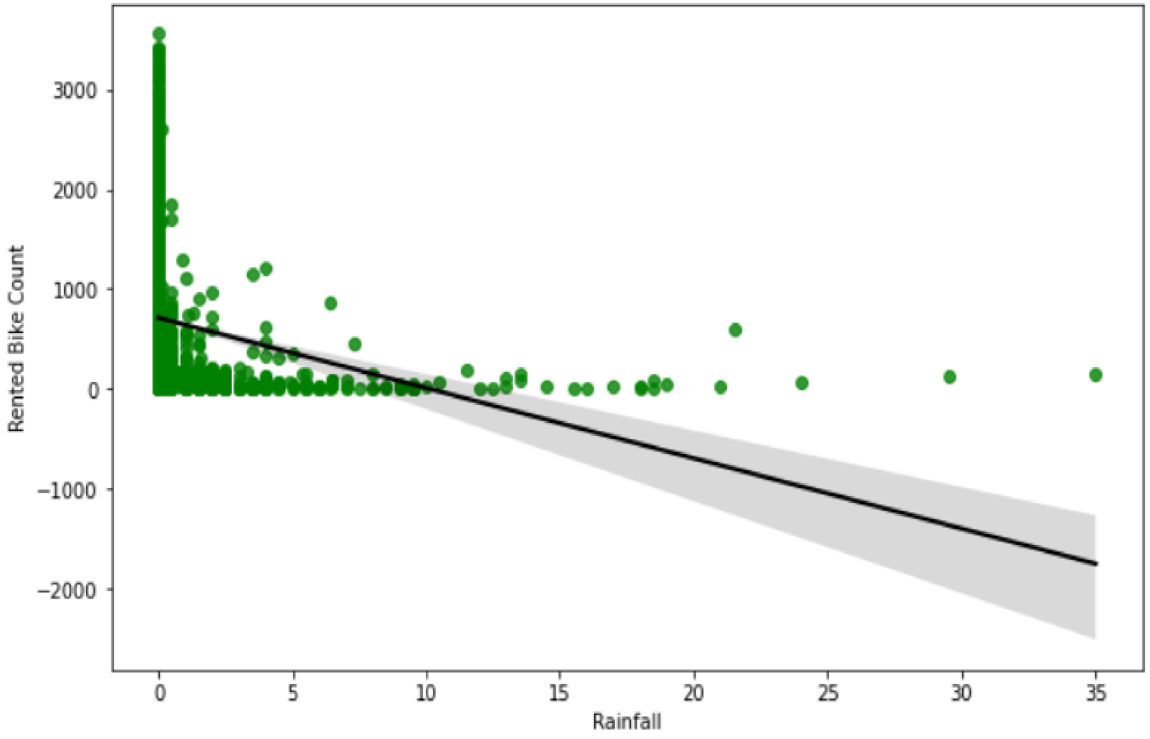
❖ Rented Bike Count vs Visibility



- There is positive correlation between rented bike count and visibility.
- So increased visibility increases bike rentals.

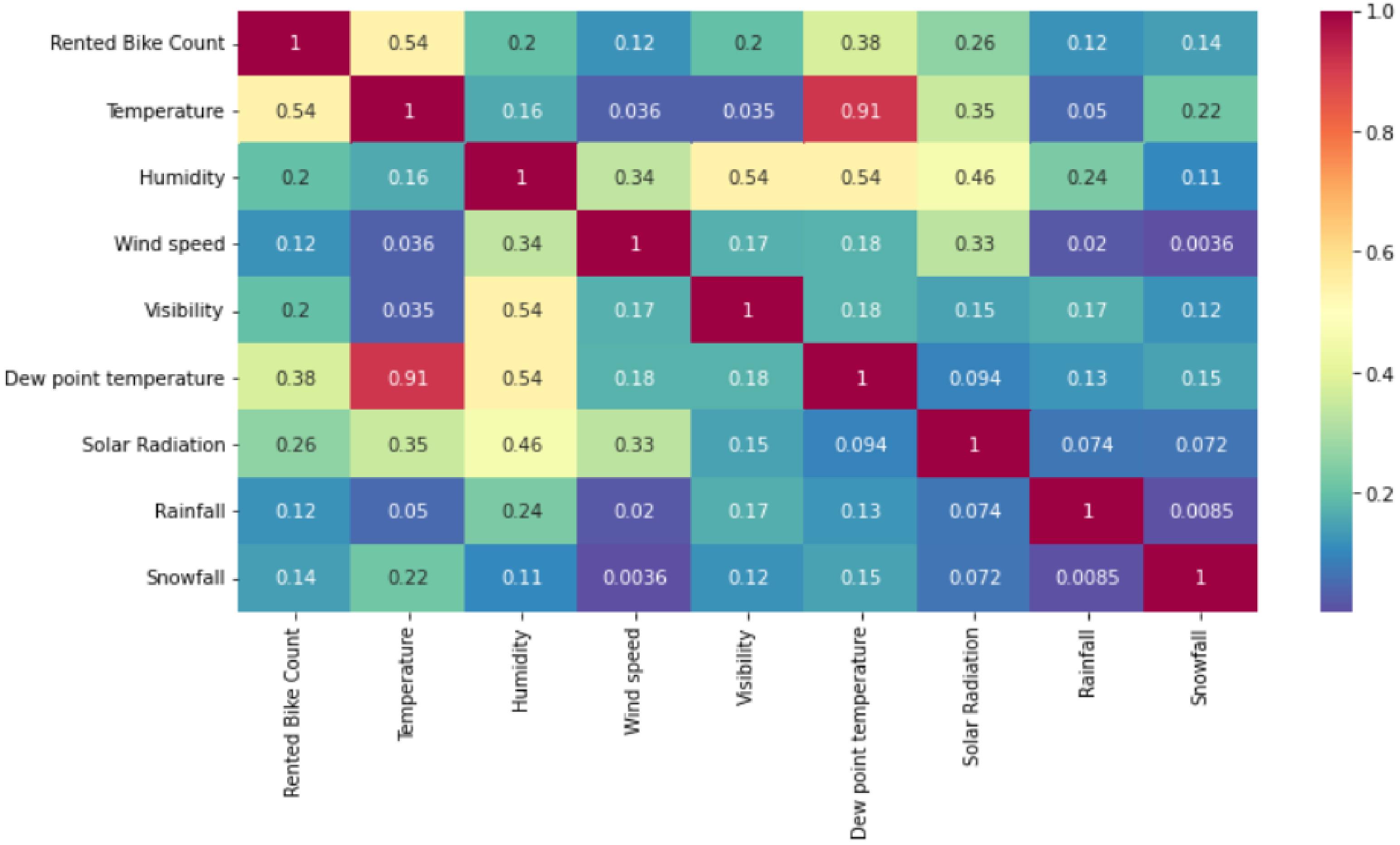
Analysis of Numerical Features

❖ Rented Bike Count vs Rainfall



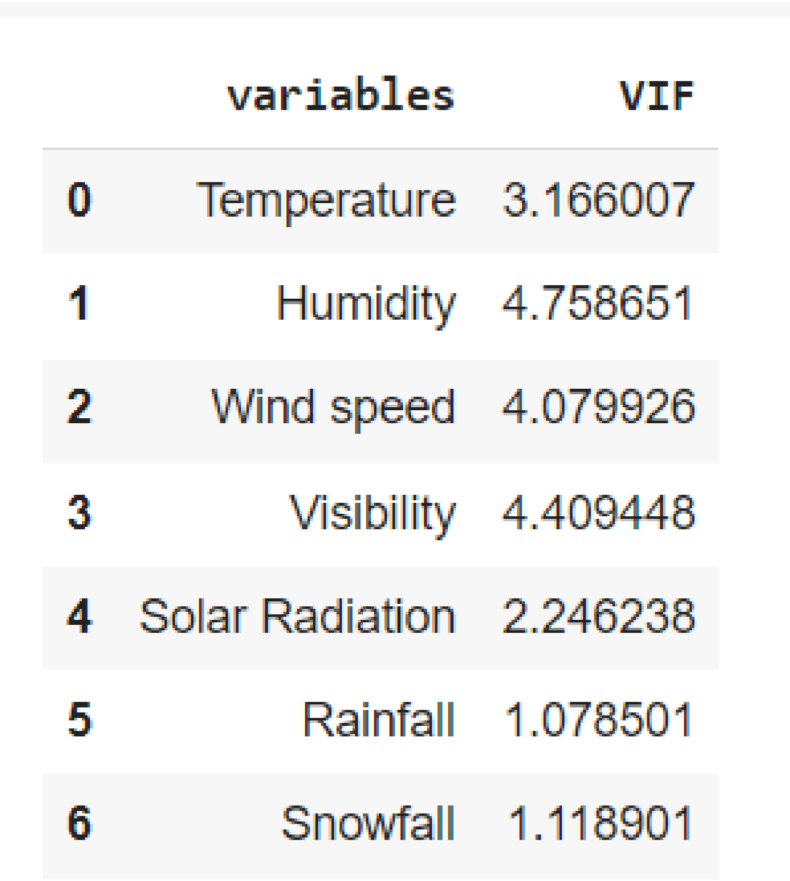
- There is negative correlation between Rainfall and rented bike count.
- Which again is obvious as no one likes to ride bikes when it’s raining.

Correlation Heatmap



- As we can see, there is high correlation between Temperature and Dew point temperature.
- As both of them are a measure of temperature only, we can remove Dew Point Temperature from our dataset.
- No other features seem highly correlated with one another. We can check the VIF(variance inflation factor) to confirm it.

Checking Multicollinearity

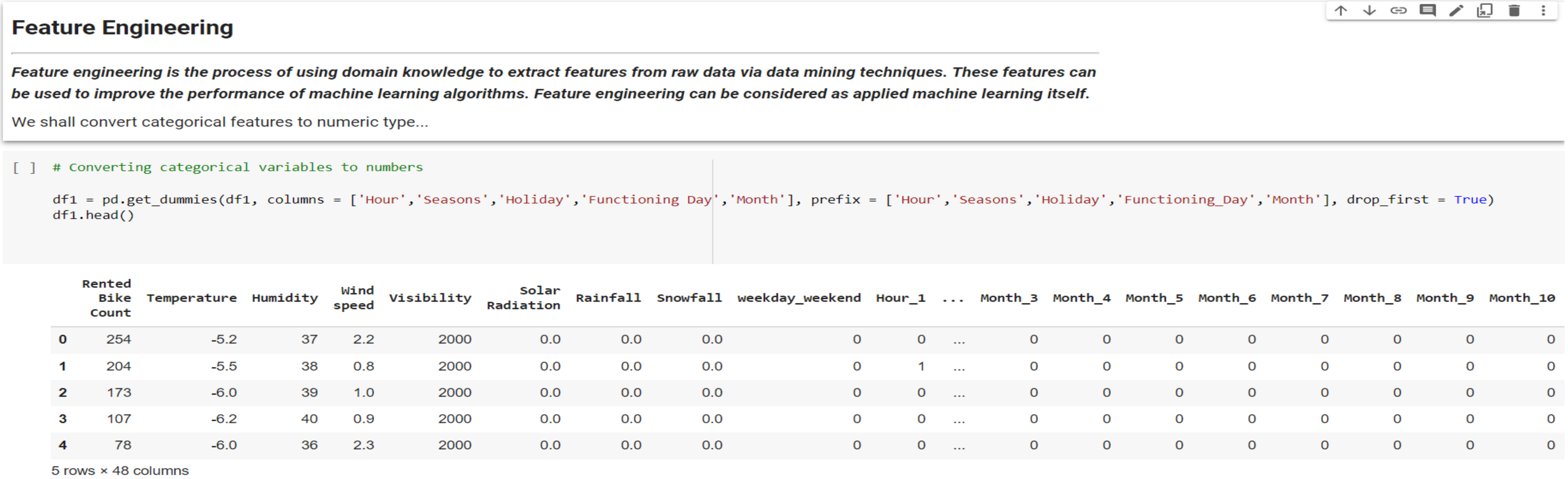


	variables	VIF
0	Temperature	3.166007
1	Humidity	4.758651
2	Wind speed	4.079926
3	Visibility	4.409448
4	Solar Radiation	2.246238
5	Rainfall	1.078501
6	Snowfall	1.118901

- Here we can see that the VIF(variance inflation factor) values of all the features is low.
- So, no Multicollinearity is observed.

Feature Engineering

❖ One hot encoding



Feature Engineering

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

We shall convert categorical features to numeric type...

```
[ ]  # Converting categorical variables to numbers

     df1 = pd.get_dummies(df1, columns = ['Hour','Seasons','Holiday','Functioning Day','Month'], prefix = ['Hour','Seasons','Holiday','Functioning_Day','Month'], drop_first = True)
     df1.head()
```

	Rented Bike Count	Temperature	Humidity	Wind speed	Visibility	Solar Radiation	Rainfall	Snowfall	weekday_weekend	Hour_1	...	Month_3	Month_4	Month_5	Month_6	Month_7	Month_8	Month_9	Month_10
0	254	-5.2	37	2.2	2000	0.0	0.0	0.0	0	0	...	0	0	0	0	0	0	0	0
1	204	-5.5	38	0.8	2000	0.0	0.0	0.0	0	1	...	0	0	0	0	0	0	0	0
2	173	-6.0	39	1.0	2000	0.0	0.0	0.0	0	0	...	0	0	0	0	0	0	0	0
3	107	-6.2	40	0.9	2000	0.0	0.0	0.0	0	0	...	0	0	0	0	0	0	0	0
4	78	-6.0	36	2.3	2000	0.0	0.0	0.0	0	0	...	0	0	0	0	0	0	0	0

5 rows × 48 columns

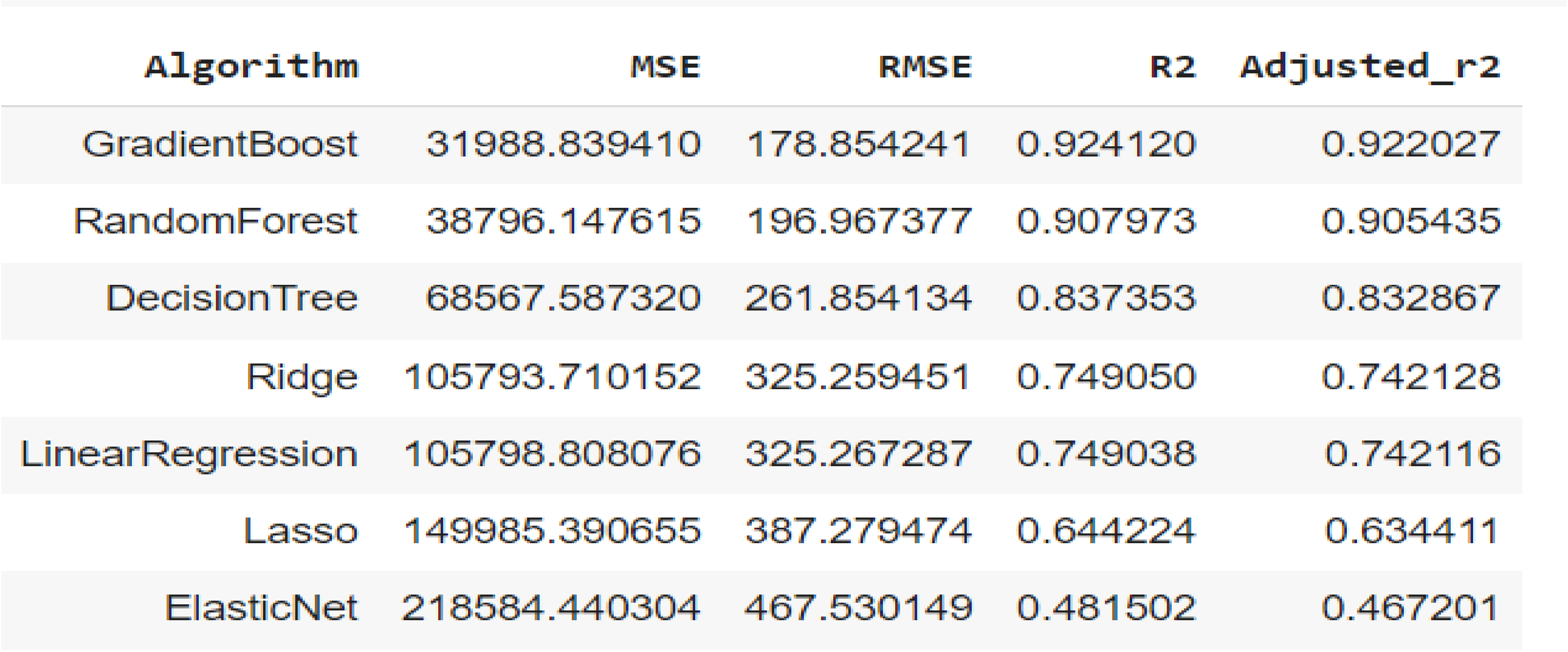
- Here I have used One Hot Encoding method for encoding all the categorical features like Hour, Month, Seasons, Holiday and Functioning Day.

Model Implementation

- After the EDA and the data preprocessing, next task was to split the data into test and train data. As all the independent feature values were of different scales, so I did scaling of the independent features before splitting them into test and train data.
- The next step was Model Implementation. I used following Machine Learning algorithms for modeling:

1. Linear Regression
2. Lasso Regression
3. Ridge Regression
4. ElasticNet
5. Decision Tree Regression
6. Random Forest Regression
7. Gradient Boosting Regression

Model Performance



Algorithm	MSE	RMSE	R2	Adjusted_r2
GradientBoost	31988.839410	178.854241	0.924120	0.922027
RandomForest	38796.147615	196.967377	0.907973	0.905435
DecisionTree	68567.587320	261.854134	0.837353	0.832867
Ridge	105793.710152	325.259451	0.749050	0.742128
LinearRegression	105798.808076	325.267287	0.749038	0.742116
Lasso	149985.390655	387.279474	0.644224	0.634411
ElasticNet	218584.440304	467.530149	0.481502	0.467201

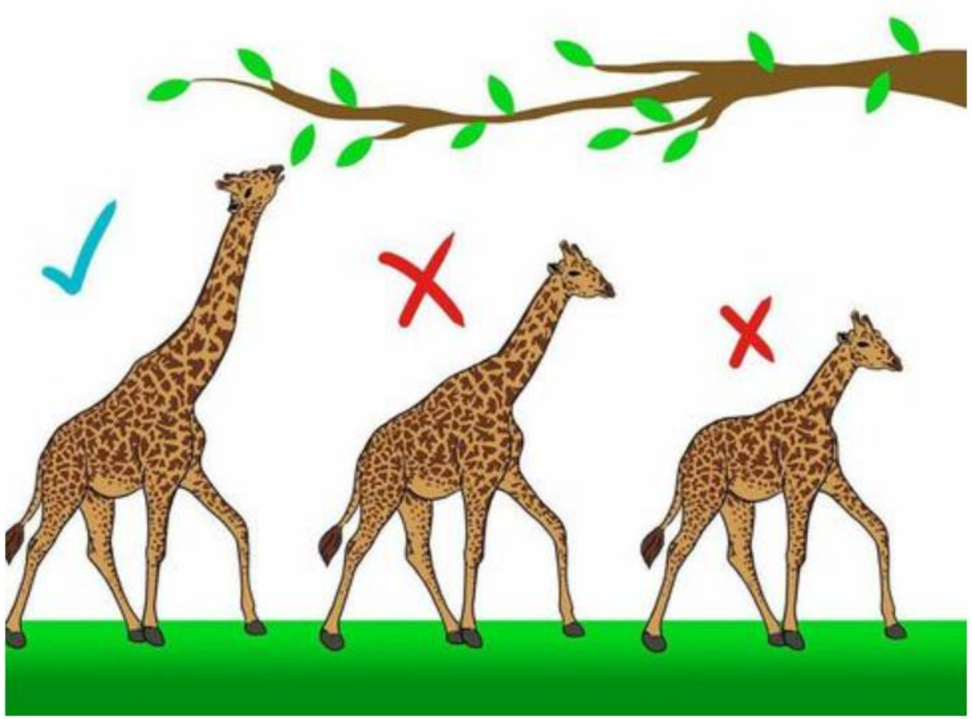
- I have used 4 evaluation metrics(MSE, RMSE, R2, Adjusted R2) to evaluate the model performance.

Model Validation & Selection

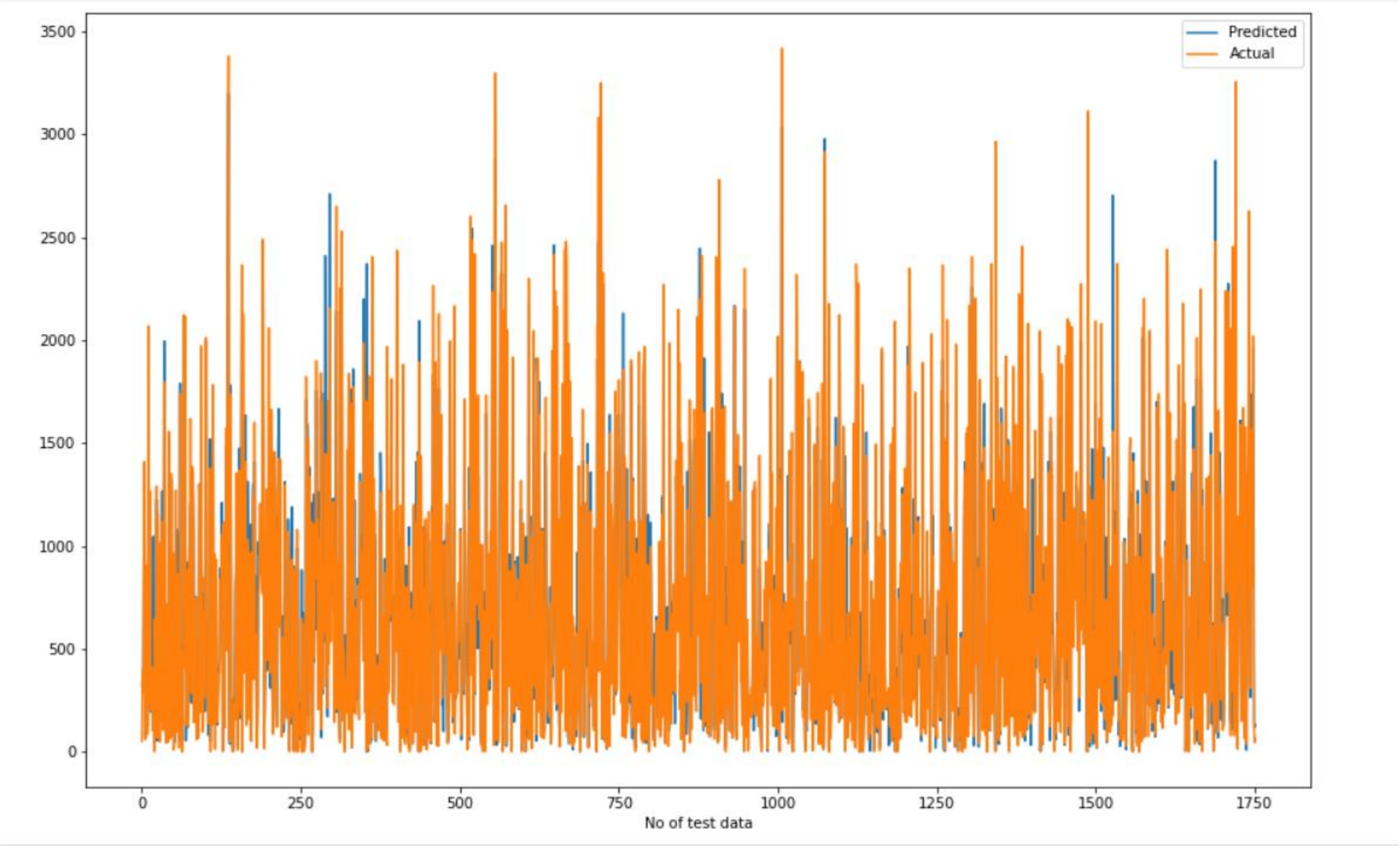
- As we can see from above table, The ElasticNet and Lasso Regression perform the worst among all the models with an R2 score of 48% and 64% respectively.
- The Gradient Boosting Regressor and Random Forest Regressor perform the best with least values of Mean Squared Errors and Root Mean Squared Errors among all the models.

Model Validation & Selection

- Both Gradient Boosting Regressor and Random Forest Regressor perform best with R2 scores of 92% and 90% respectively.
- From above observations, we conclude that we can deploy these two models and specifically Gradient Boosting Regressor for further predictions.

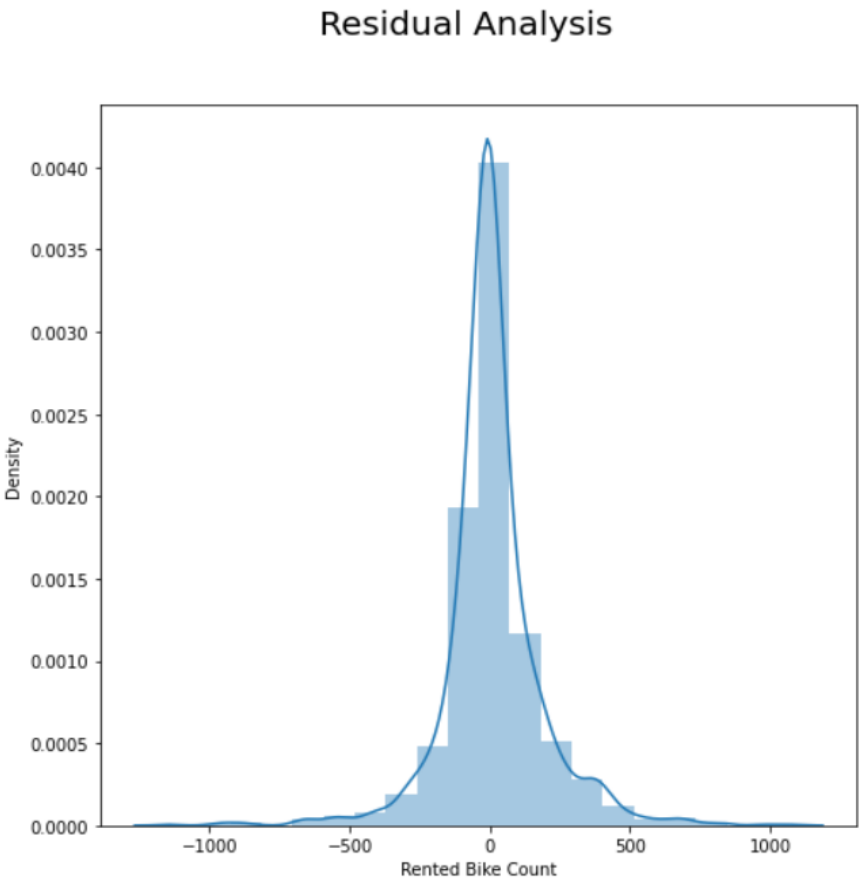


Actual vs Predicted

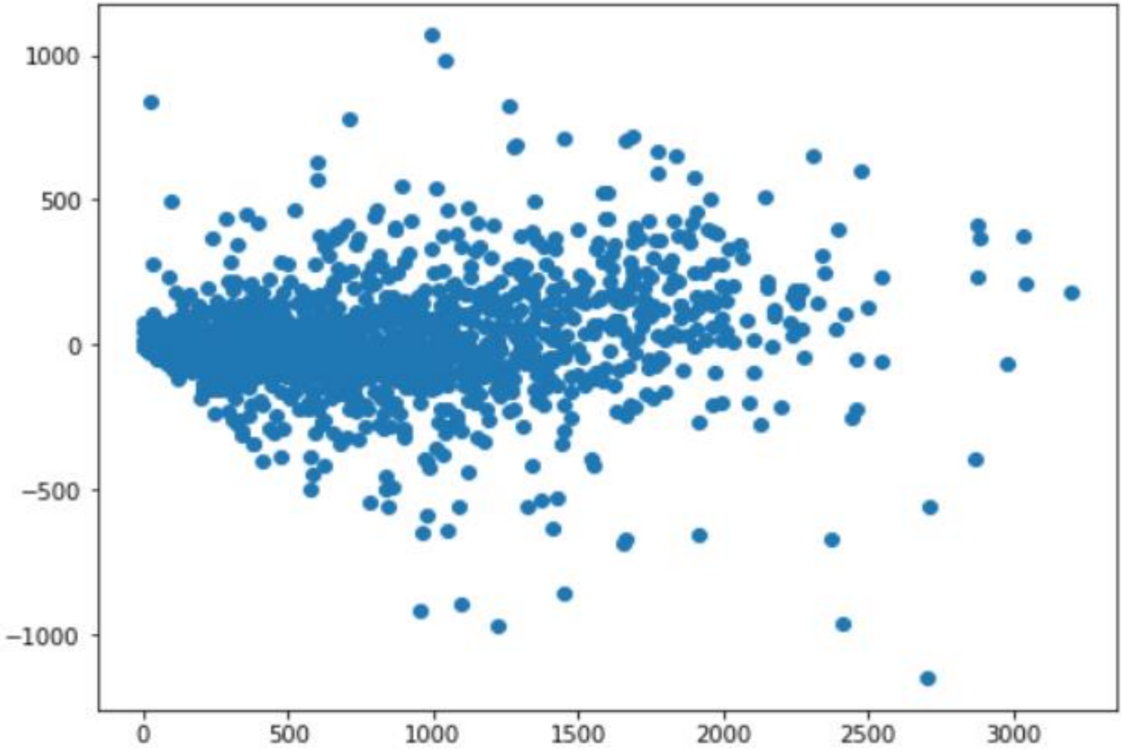


- We can see from the plot how well our predicted values fit with the actual values.

Residual Analysis



Homoscedasticity

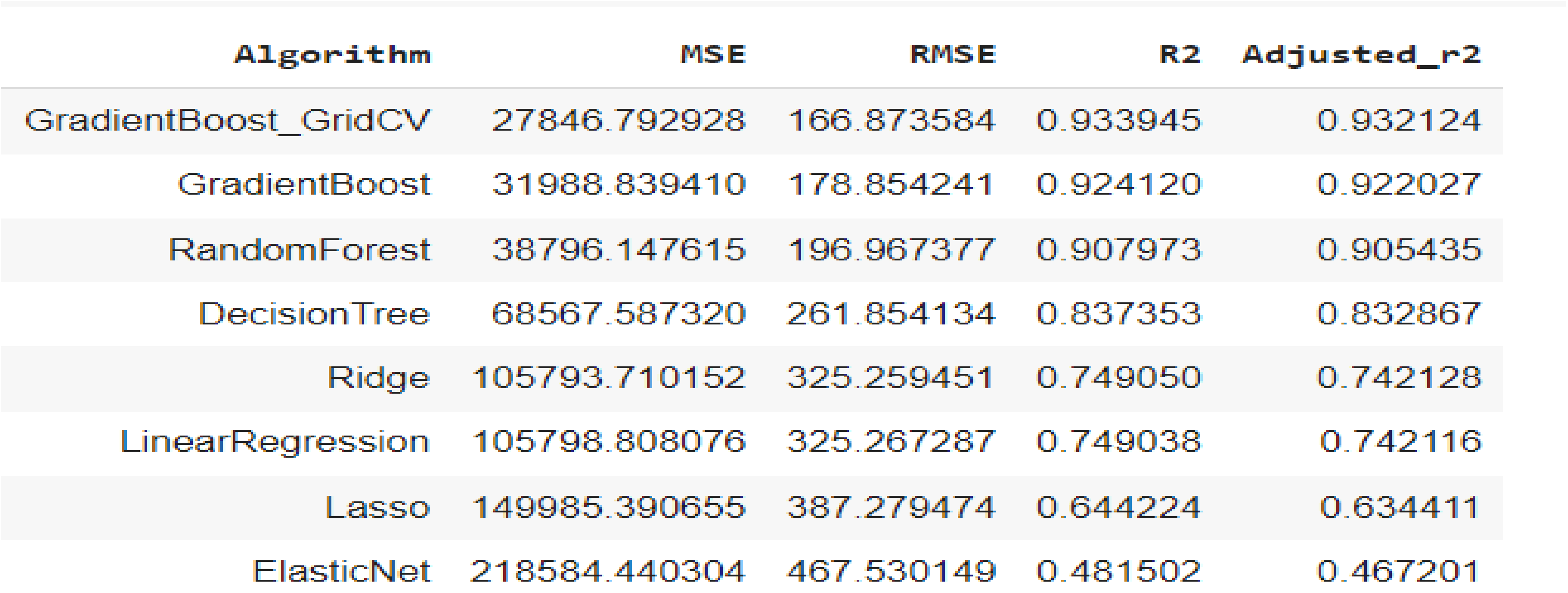


- We can see, the residuals are also normally distributed.
- And we can see from the scatter plot between the predicted values and the residuals, that there is no proper pattern between them, hence, there is Homoscedasticity

Hyperparameter Tuning

- Next, I did Hyperparameter Tuning with Cross Validation for Gradient Boosting Regression model to see if I can further enhance the performance of the model.
- I have used GridSearchCv for this and provided the model with grids of hyperprameters like: n_estimators, max_depth, min_sample_split and min_sample_leaf.

Post Hyper Parameter Tuning Performance



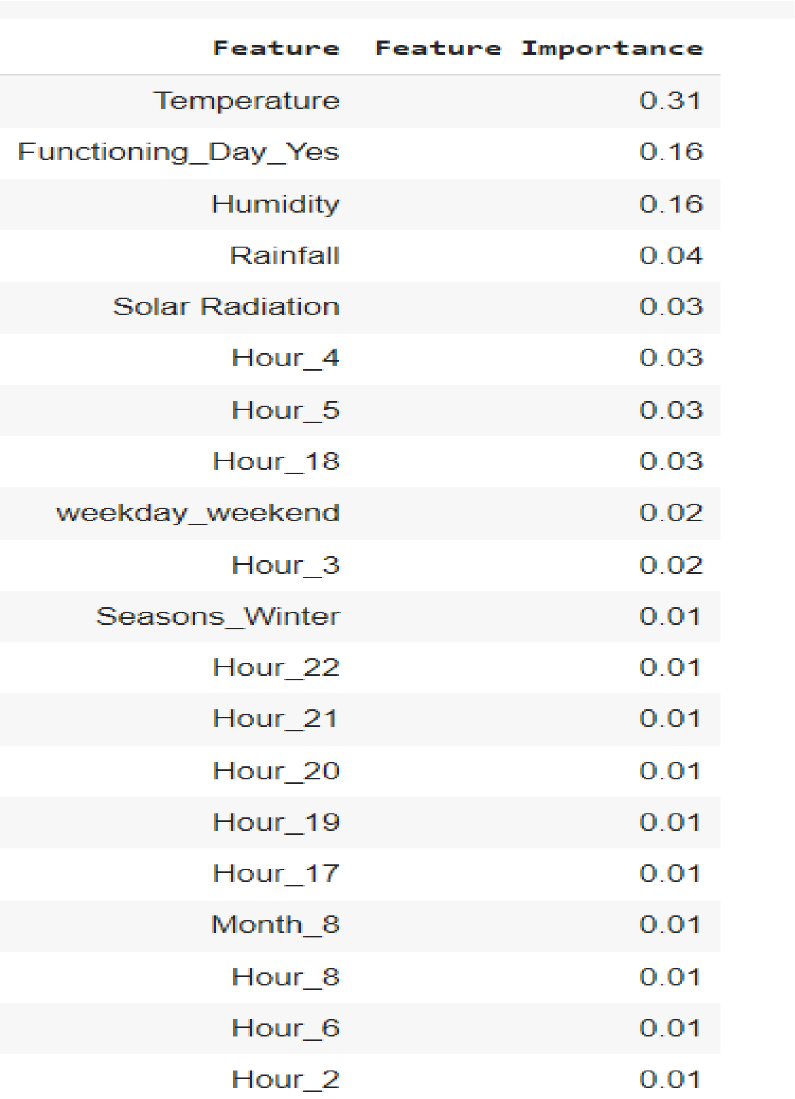
Algorithm	MSE	RMSE	R2	Adjusted_r2
GradientBoost_GridCV	27846.792928	166.873584	0.933945	0.932124
GradientBoost	31988.839410	178.854241	0.924120	0.922027
RandomForest	38796.147615	196.967377	0.907973	0.905435
DecisionTree	68567.587320	261.854134	0.837353	0.832867
Ridge	105793.710152	325.259451	0.749050	0.742128
LinearRegression	105798.808076	325.267287	0.749038	0.742116
Lasso	149985.390655	387.279474	0.644224	0.634411
ElasticNet	218584.440304	467.530149	0.481502	0.467201

Model Validation & Selection

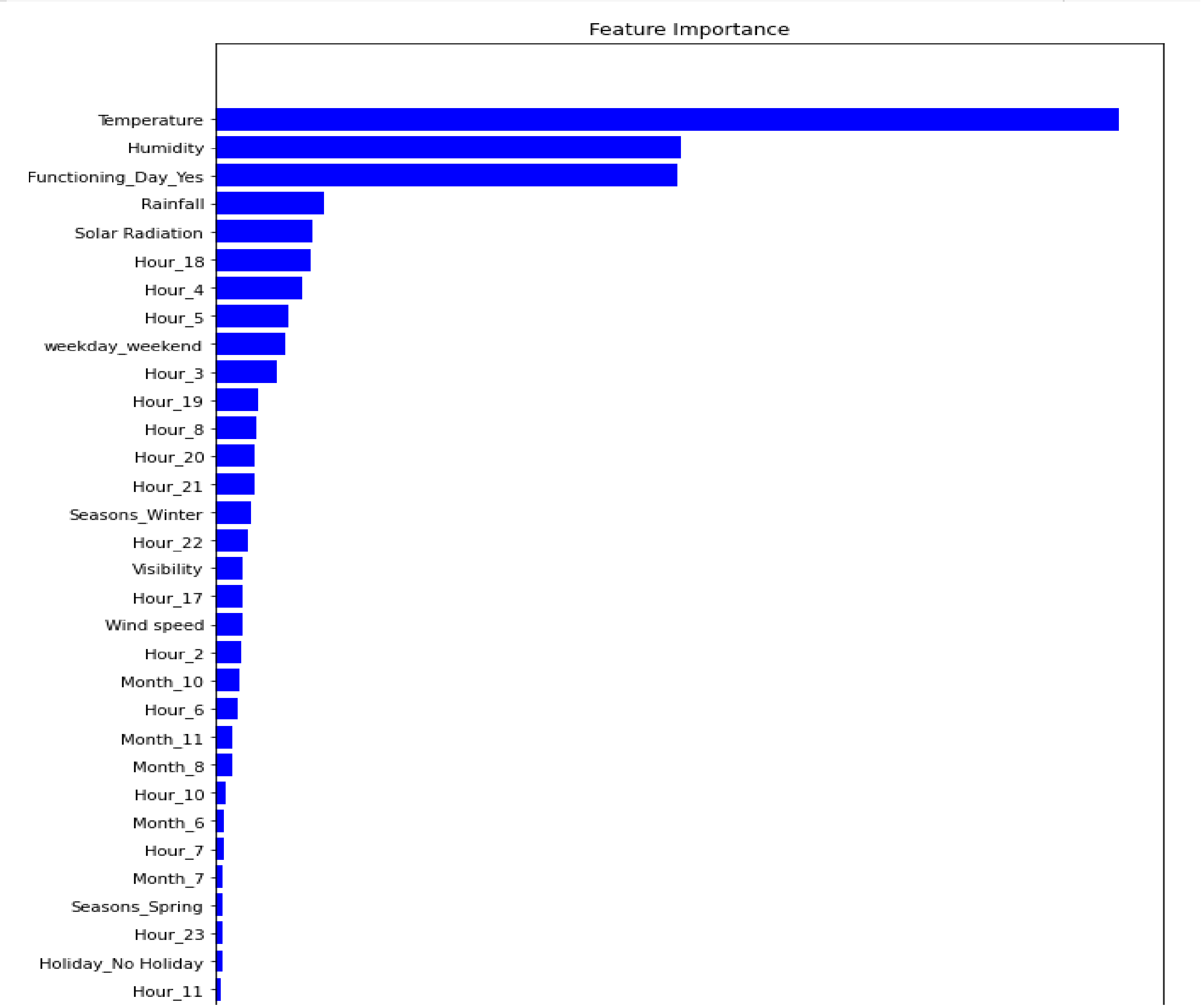
- We can see from the above table that, the performance of Gradient Boosting Regression has improved after using hyper parameter tuning with an R2 score of 93%.
- So we can deploy this model with these set of hyper parameters.
- The best parameters are :

1. Max_depth = 8
2. Min_sample_leaf = 40
3. Min_sample_split = 10
4. n_ estimators = 4500

Feature Importance



Feature	Feature Importance
Temperature	0.31
Functioning_Day_Yes	0.16
Humidity	0.16
Rainfall	0.04
Solar Radiation	0.03
Hour_4	0.03
Hour_5	0.03
Hour_18	0.03
weekday_weekend	0.02
Hour_3	0.02
Seasons_Winter	0.01
Hour_22	0.01
Hour_21	0.01
Hour_20	0.01
Hour_19	0.01
Hour_17	0.01
Month_8	0.01
Hour_8	0.01
Hour_6	0.01
Hour_2	0.01



Conclusion

From EDA

- Bike rental count is positively correlated with Temperature and Wind Speed, which means bike rentals count is low when temperature is low and people like to rent bike when its windy.
- Bike rental count is negatively correlated with Rainfall and Snowfall.
- More number of bikes are rented on working days.
- Most of the bike rentals were from months April to October, which again means people don't like to rent bikes during winters.
- Bike rental count is high during work going hours(7pm to 9pm) and then during later parts of the day(from 3pm to 10pm)

Conclusion

From Modeling

- Random Forest Regressor, Gradient Boost Regressor and Gradient Boost GridSearchCV perform best in predicting with R2 score of 90%, 92% and 93% respectively.
- Feature importance value of Gradient Boost and Gradient Boost GridSearchCV are slightly different
- We can deploy these three models for predicting the rental bike count at any given hour.

Thank You