Regression Analysis on Bike Sharing Demand Prediction

Submitted By:
Lakshmi Narayana
Data Science Trainee
At Alma Better

Demand for Bike Sharing

- Problem Statement
- EDA and Feature engineering
- Feature selection
- Data preparation
- Implementation of model
- Evaluation of Model

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.

Data Pipeline

- **Data processing-:** In this phase I have checked for any presence of null values and change the columns containing date time.
- **Data processing-2:** In this phase I have gone through each feature which are selected in the first phase and encoded some of the numerical features and few categorical features into binary form.
- **EDA:** In this part some exploratory data analysis was done on the features selected in part 1 and 2.
- **VIF:** Selected only the features which does not have multi correlation among them.
- **Create a model:** Finally in this part we create models trained and tested it on the available dataset.

Data Description

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

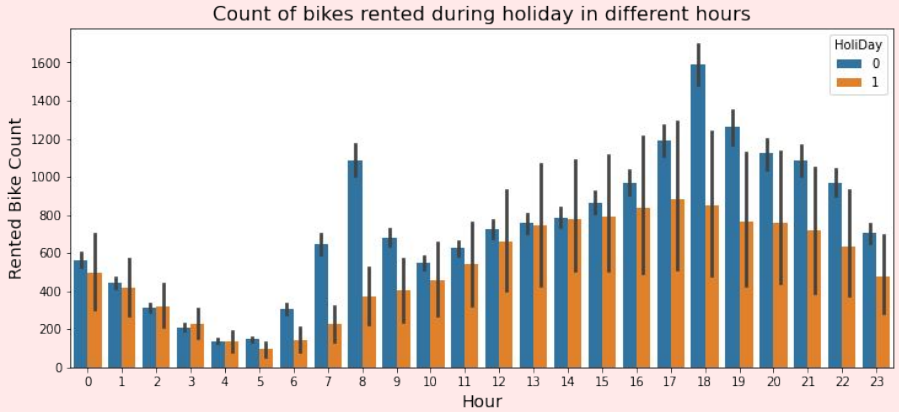
Attribute Information:

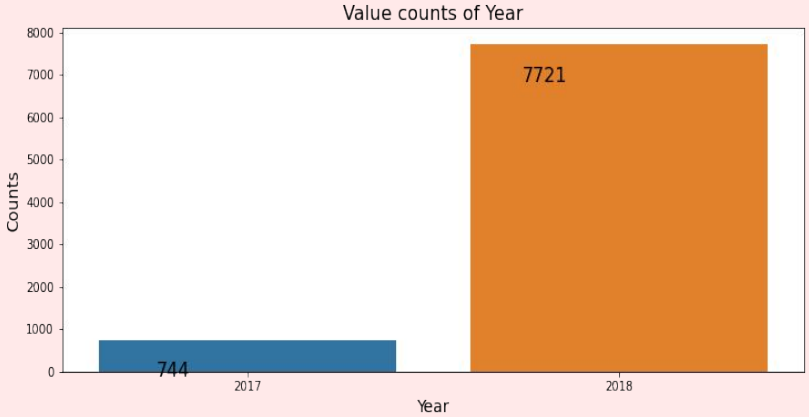
- Date : year-month-day
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of the day
- Temperature-Temperature in Celsius
- Humidity - %
- Wind Speed - m/s
- Visibility - 10m
- Dew point temperature - Celsius
- Solar radiation - MJ/m2
- Rainfall - mm
- Snowfall - cm
- Seasons - Winter, Spring, Summer, Autumn
- Holiday - Holiday/No holiday
- Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours)

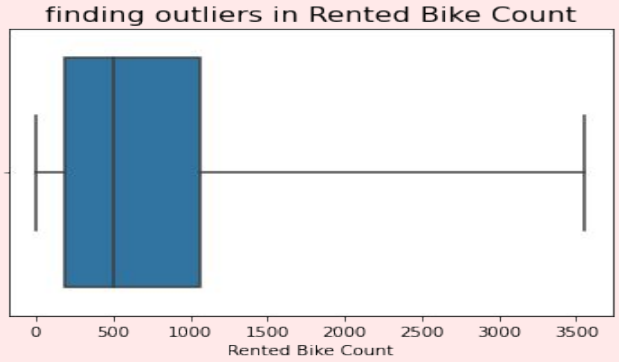
Defining the Dependent Variable

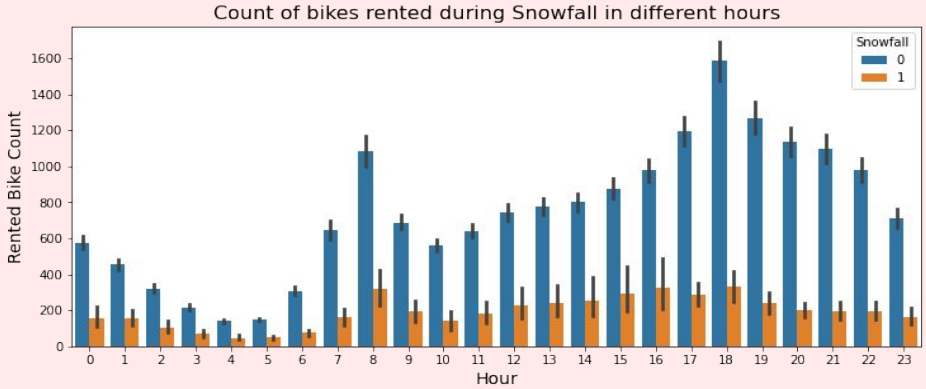
Rented Bike count is taken as the Dependent Variable in the dataset.

- Rented Bike count - Count of bikes rented at each hour

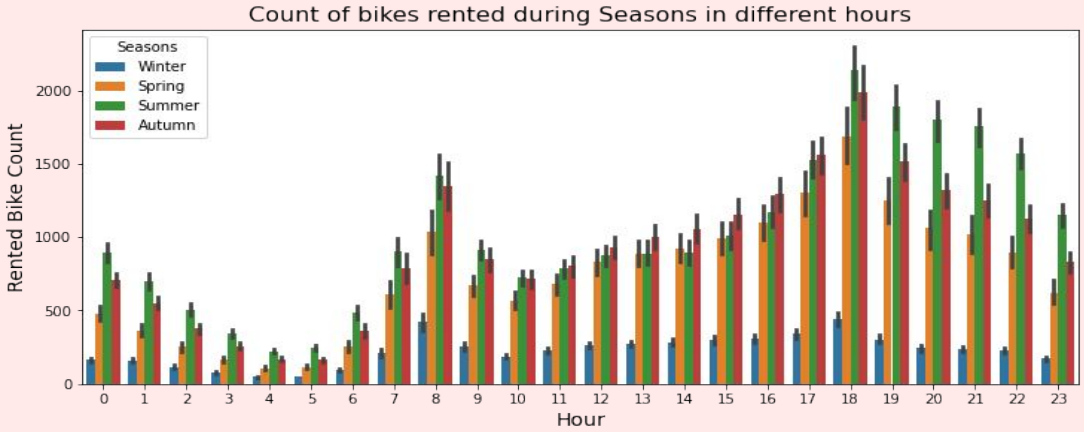


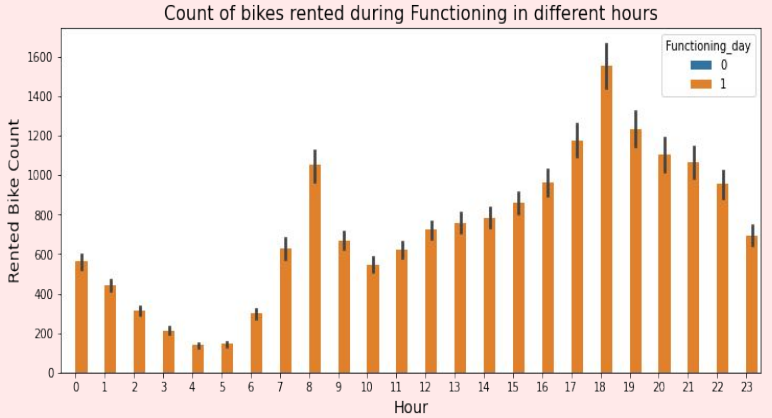


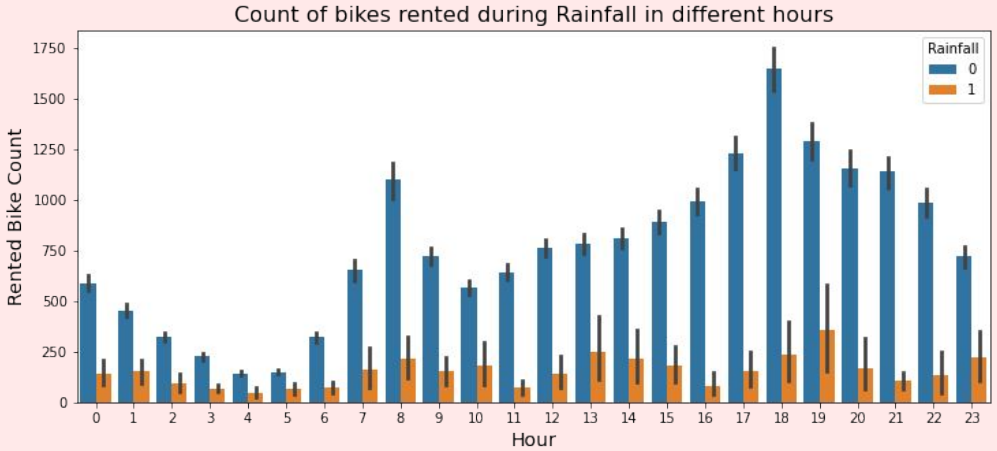


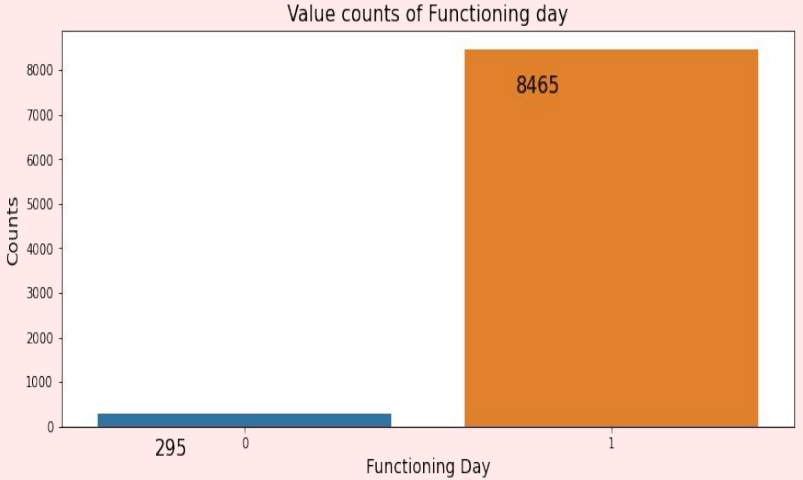


Defining the Dependent Variable



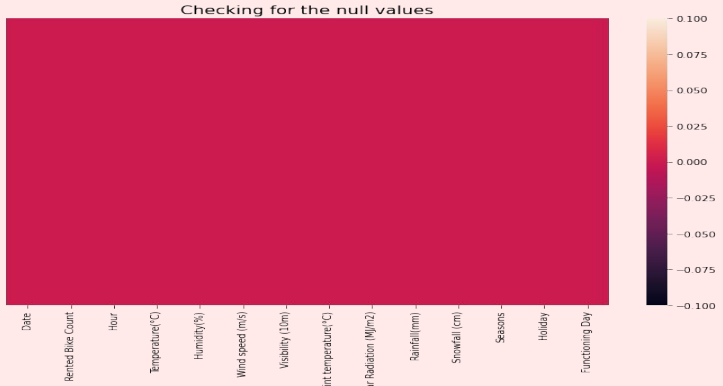


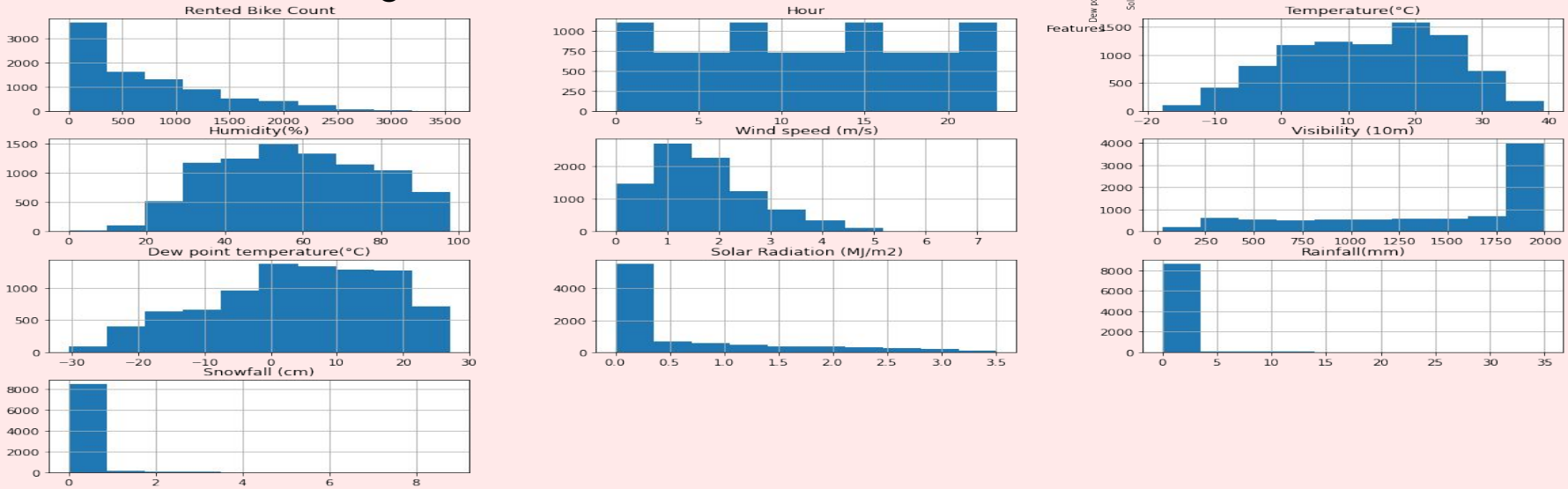




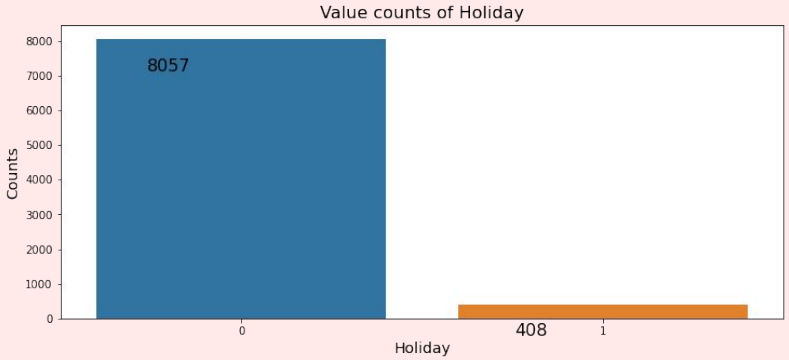
Exploratory Data Analysis

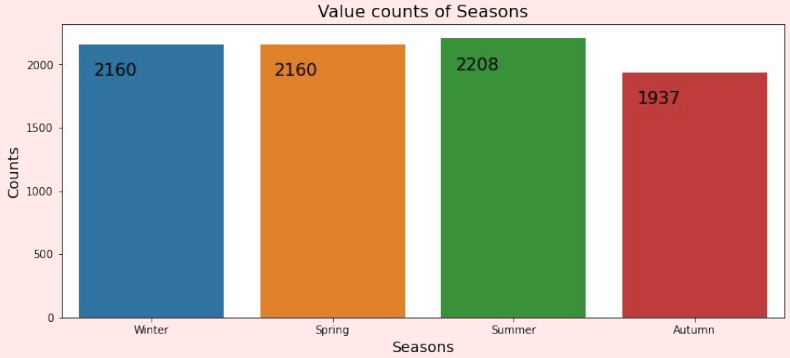
- Cleaning the null values.
- Changed the columns of date time.
- Dropped unwanted features.
- Binary encoding.
- Renamed featured.
- Exploratory Analysis.
- One hot encoding.

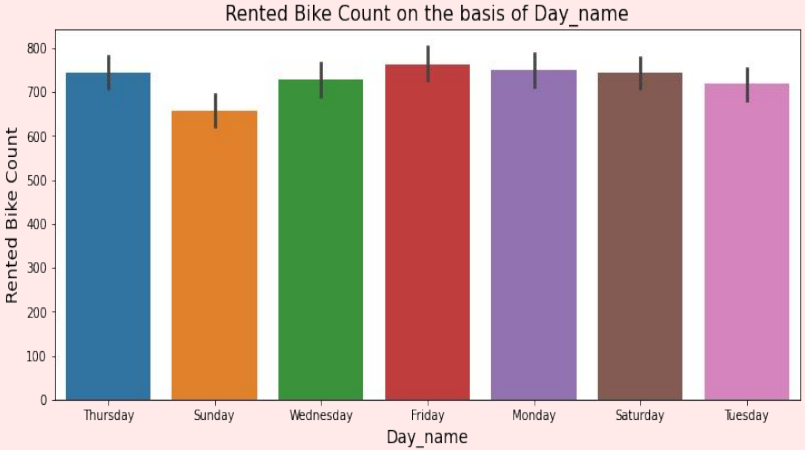


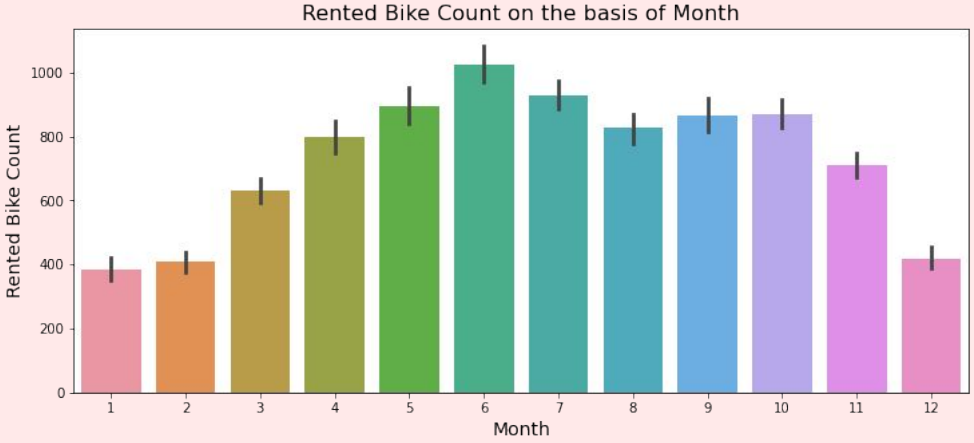


Exploratory Data Analysis

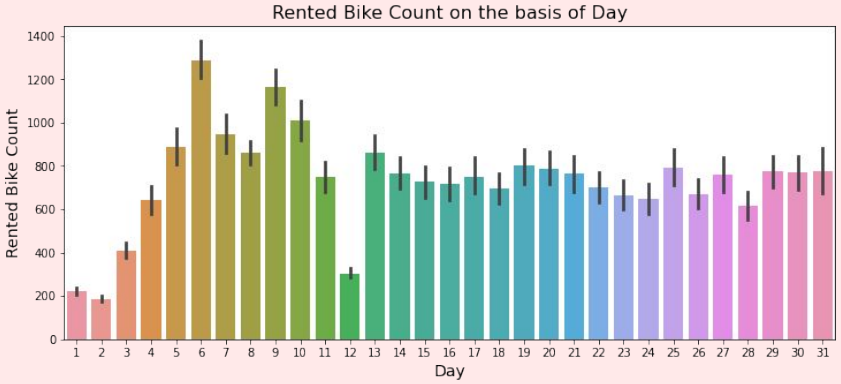


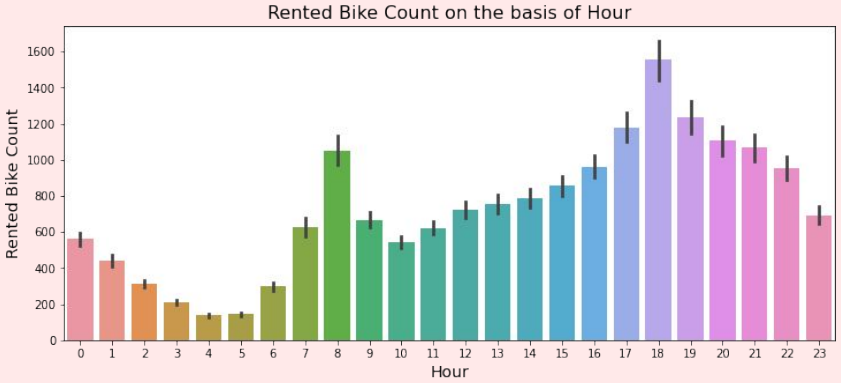


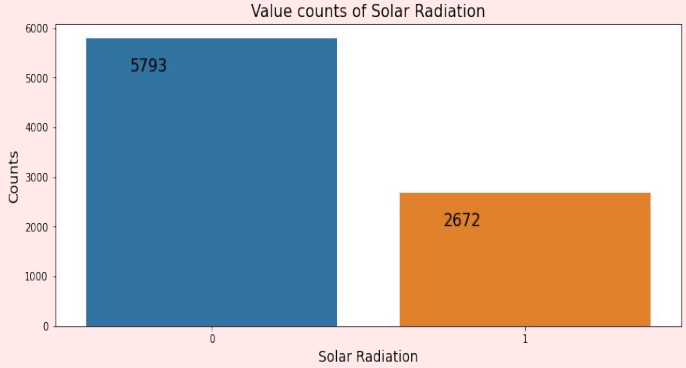




Exploratory Data Analysis

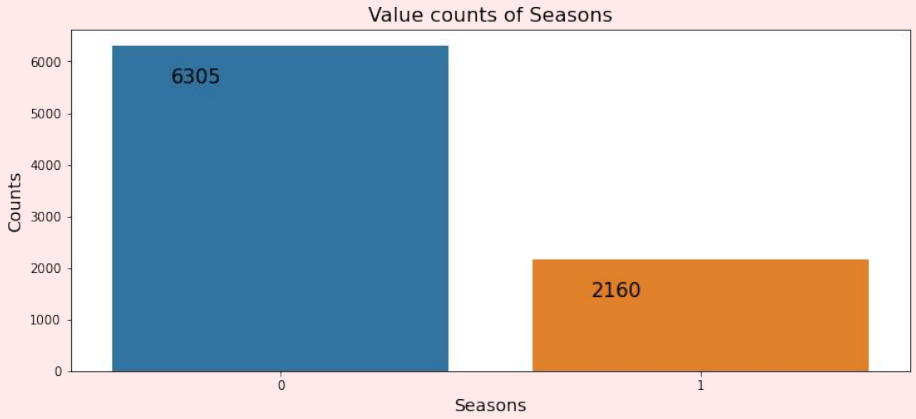






Data Preparation

- One hot encoding was done for seasons feature.
- Variance inflation factor was used to check the correlation among the variables.
- Few features are dropped.
- New data frame was made and named it as bike_df.
- Divided the dataset into train and test set in the ratio of 80:20
- StandardScaler was used to scale the data.



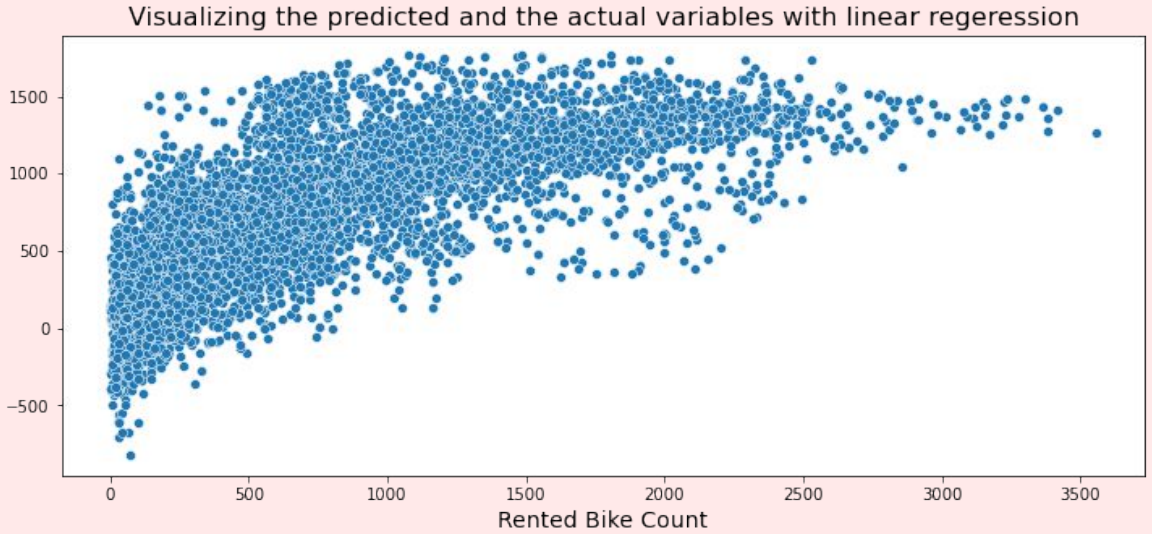

```
#size of train and test datasets
print(f'Size of X_train is: {X_train.shape}')
print(f'Size of X_test is: {X_test.shape}')
print(f'Size of y_train is: {y_train.shape}')
print(f'Size of y_test is: {y_test.shape}')
```

```
Size of X_train is: (6772, 15)
Size of X_test is: (1693, 15)
Size of y_train is: (6772,)
Size of y_test is: (1693,)
```

Model Implementation

Linear Regression:

- The Scaled data was used for the model implementation.
- Fit the trained dataset to the model.
- The regressor score got from the model is 0.536337.
- Defining the predicted values form the model.



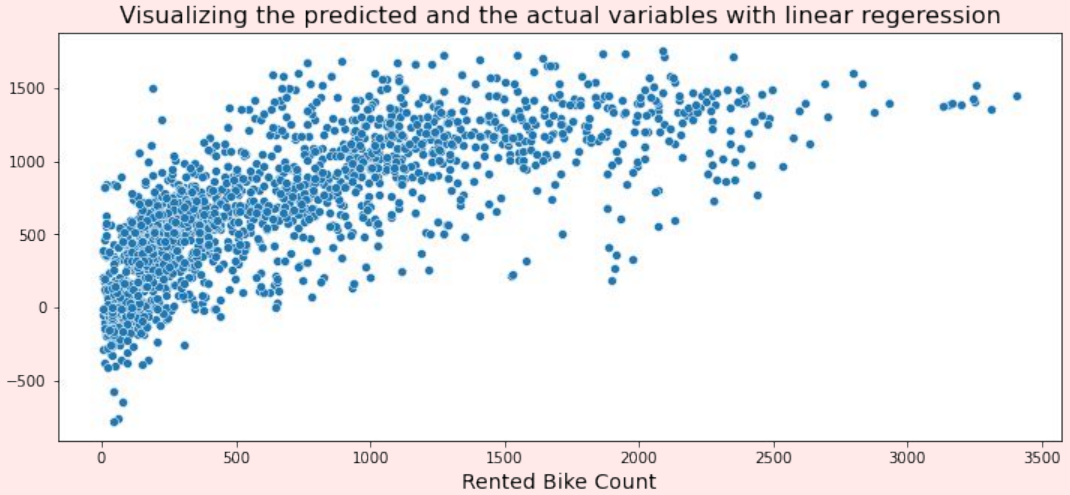
Linear Regression

```
MSE_test = mean_squared_error(y_test, pred_test)
print(f'MSE= {MSE_test}')

RMSE_test = np.sqrt(MSE_test)
print(f'RMSE= {RMSE_test}')

R2_Score_test = r2_score(y_test, pred_test)
print(f'R2_Score= {R2_score_test}')
```

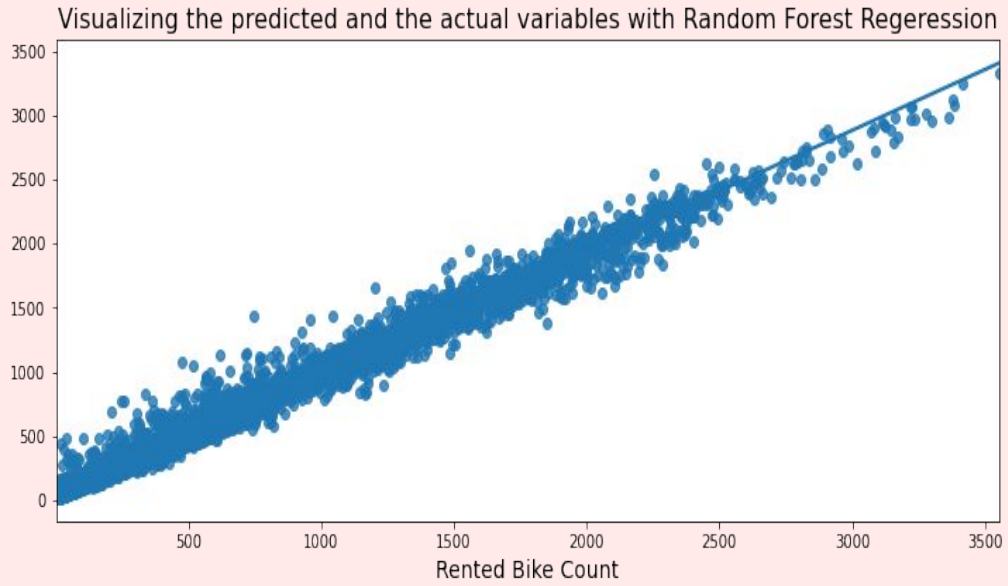
```
MSE= 191262.5049036467
RMSE= 437.33568903491823
R2_Score= 0.5133524012839992
```



Model Implementation

Random Forest Regressor:

- The Scaled data was stored in the dataframe.
- Fit the trained dataset to the model.
- Defining the predicted values form the model.




```
MSE_train = mean_squared_error(y_train, pred_train)
print(f'MSE= {MSE_train}')

RMSE_train = np.sqrt(MSE_train)
print(f'RMSE= {RMSE_train}')

R2_Score_train = r2_score(y_train, pred_train)
print(f'R2_Score= {R2_Score_train}')
```

```
MSE= 7243.796700926611
RMSE= 85.11049700786978
R2_Score= 0.9822279811540209
```

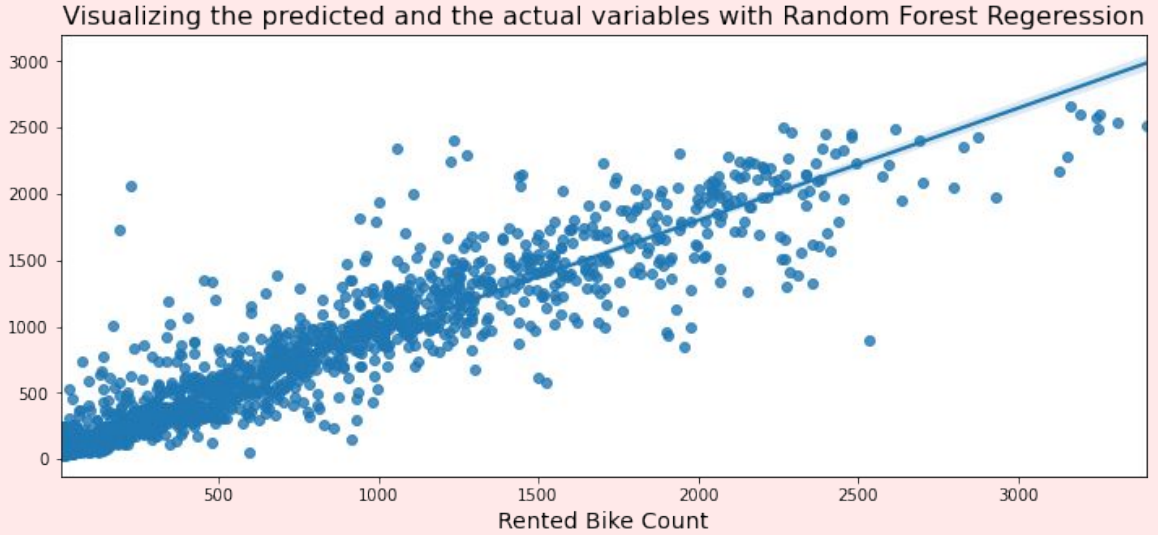
Random Forest Regressor

```
MSE_test = mean_squared_error(y_test, pred_test)
print(f'MSE= {MSE_test}')

RMSE_test = np.sqrt(MSE_test)
print(f'RMSE= {RMSE_test}')

R2_Score_test = r2_score(y_test, pred_test)
print(f'R2_Score= {R2_Score_test}')
```

```
MSE= 57075.793045832834
RMSE= 238.90540606238451
R2_Score= 0.8679930131350906
```



Conclusion

- The rented bike count is good when the weather is clear in all the seasons.
- And within the seasons the demand is somewhat high during summer.
- There is more demand at 8 in the morning and 6 in the evening, which seems to the office opening and closing hours.
- After 6 in the evening i.e., office hours also we have slightly more demand in the bike rentals.
- As per the evaluation its better to implement the Random Forest Regression rather that going for Linear Regression.
- When it comes to the accuracy the Random Forest Regression is performing well on the test dataset with the accuracy of 86.79%

Thank you