Regression Analysis on the Bike Sharing Demand Prediction

Kanike Lakshmi Narayana
Data science Trainee at
AlmaBetter

Abstract

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.

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 main objective is to predict what demand for the bike rentals is on a certain day with the help of a given dataset.

Introduction

The population is increasing day-by-day. To make transportation possible, owning a vehicle is a good idea but when we think of pollution, fuel consumption and traffic it seems to be a problem. In developed cities bike pooling and sharing is viewed as a smart solution to overcome this problem. Therefore, bike sharing is a brilliant idea which provides people with another short range transportation option that allows them to travel without worrying about being stuck in traffic and maybe enjoy city view or even workout at the same

The problem for providing the bike rentals is this system should be made available in all over the remote places too. The demand for bike hiring is increasing, but it is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time.

Machine Learning is concerned with computer programs that automatically improve their performance through experience. In machine learning we have supervised learning, unsupervised learning and reinforcement learning. Again the supervised learning is further divided into regression and classification. In this project we are going to look after the supervised learning regression model. This regression model is used when we have to predict something from the continuous valued output (prices).

Objective

The main objective is to predict the demand for bike rentals.

Dataset Peeping

Owing to the size of the dataset, extensive cleaning of the dataset is not needed. But the following steps are performed for the analysis purpose:

- Dataset was clean and does not contain any NaN values.
- Changed the format of the Date.
- Added some columns which are extracted from the Date column.

Data Design

Exploring the fields and records in the dataset:

- Date: Fields in Date column contains only 2 separations, so it is assumed that it contains only the month and the year.
- Open: Day Opening price of Yes Bank Stock over the years.
- High: Consists the highest price of the stock recorded on that day.
- Low: Consists the lowest price of the stock recorded on that day.
- Close: Day Closing price of Yes Bank Stock over the years.
- Year: Consists the year extracted from the Date column.
- Month: Consists the month extracted from the Date column.
- Day: Consists the date extracted from the Date column.

Challenges Faced

The following are the challenges faced in the data analysis:

- Actual Data size was small, but clean and good.
- Added some of the columns.

Data Design

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)

Challenges Faced

The following are the challenges faced in the data analysis:

- The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.
- Converted the data and did binary encoding.
- Added some of the columns.

Approach

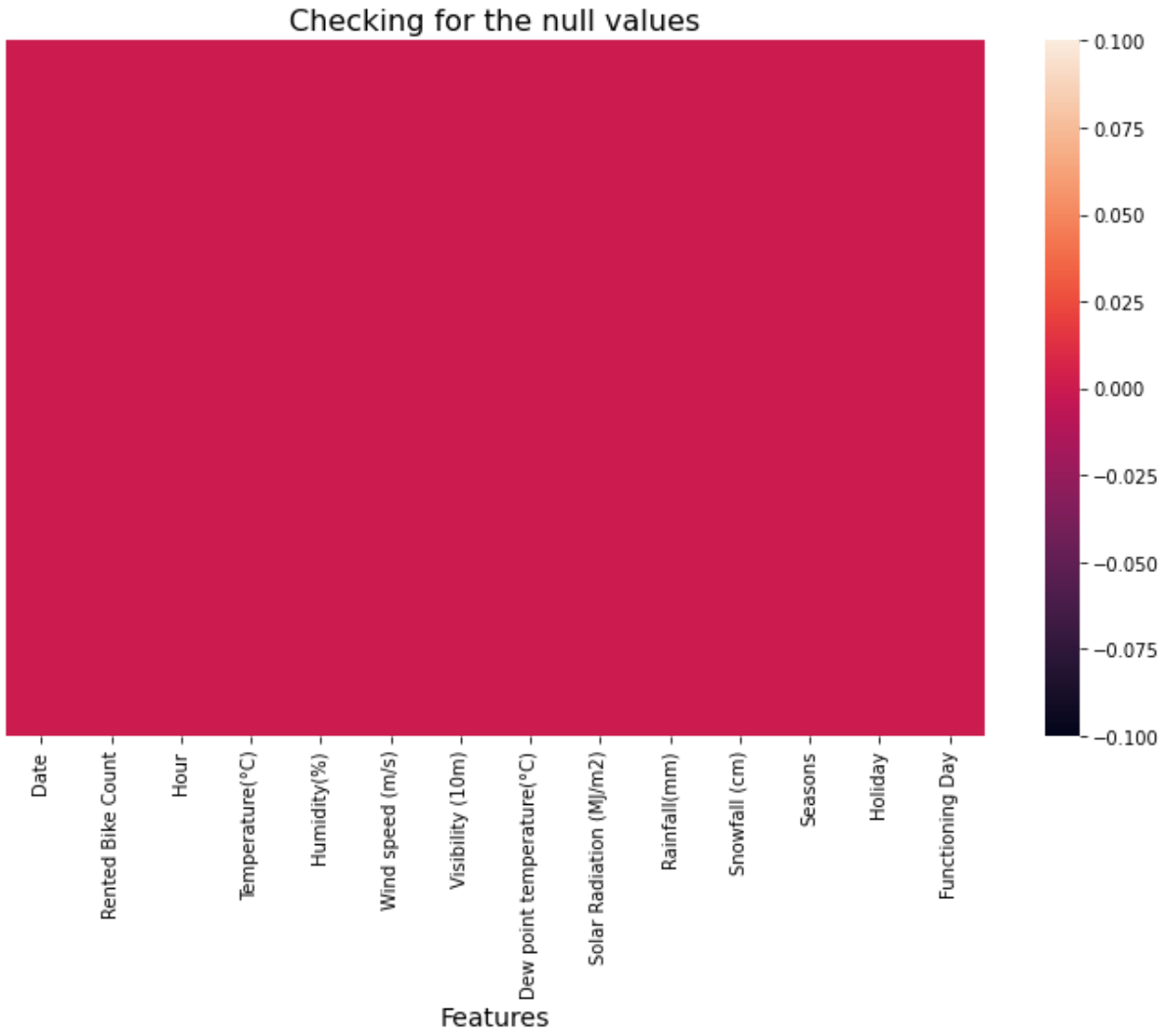
As the problem statement says the main object is to predict the demand for bike sharing, used the supervised learning regression analysis, Linear Regression, Lasso regression, Ridge Regression and Decision Tree classifier for the purpose of training the dataset to predict the closing price of the stock.

Tools Used

The whole project was done using python, in google colaboratory. Following libraries were used for analysing the data and visualizing:

- Pandas: Extensively used to load and wrangle with the dataset.
- Matplotlib: Used for visualization.
- Seaborn: Used for visualization.
- Datetime: Used for analysing the date variable.
- Warnings: For filtering and ignoring the warnings.
- Numpy: For some math operations in predictions.
- Sklearn: For the purpose of analysis and prediction.
- Datetime: For reading the date.
- Scipy: To know the skewness and kurtosis of the data.
- Statsmodels: For outliers influence.

Visualizing the presence of NaN values



The above figure shows that there are no NaN (Not a Number) values in the given dataset.

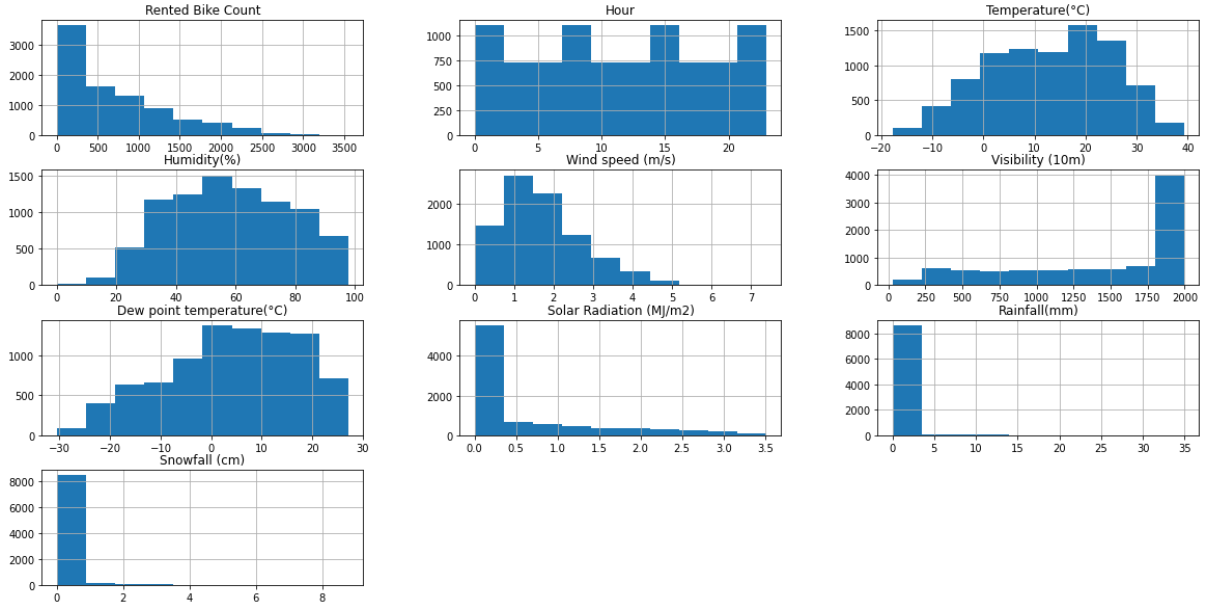
Pandas DataFrame

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes
3	01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
4	01/12/2017	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.0	Winter	No Holiday	Yes

The table shows the dataset in the form of Pandas DataFrame.
The dataset has 8760 rows and 14 columns wholly the shape of (8760, 14).
It contains the following columns:

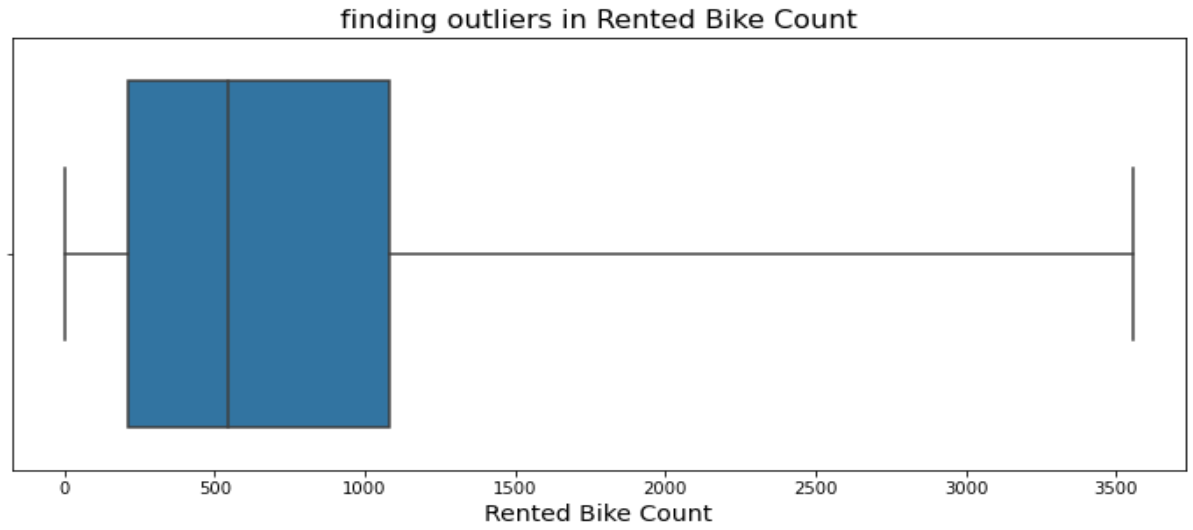
- Date
- Rented Bike Count
- Hour
- Temperature(°C)
- Humidity (%)
- Wind speed (m/s)
- Visibility (10m)
- Dew point temperature(°C)
- Solar Radiation (MJ/m2)
- Rainfall(mm)
- Snowfall (cm)
- Seasons
- Holiday
- Functioning Day

Histogram representation of the data



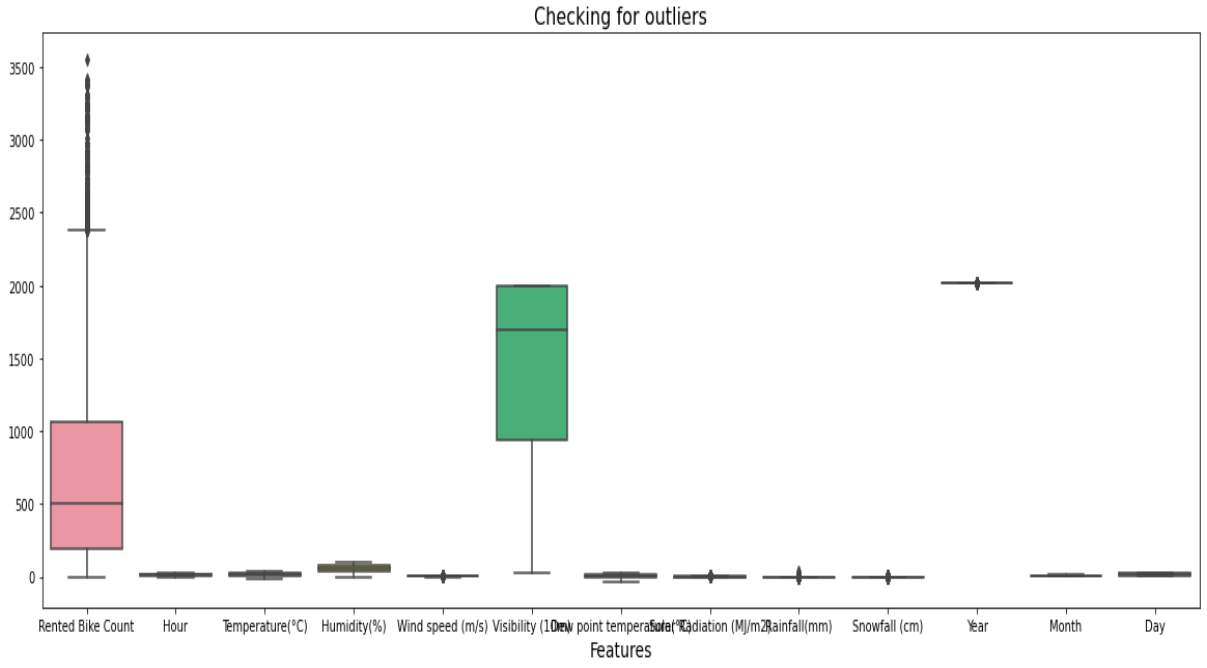
The above figure depicts the distribution of all the columns of the dataset in a separate histogram and shows the density of such columns. It includes some extra columns that are added for analysis purposes.

Distribution of Rented Bike count data



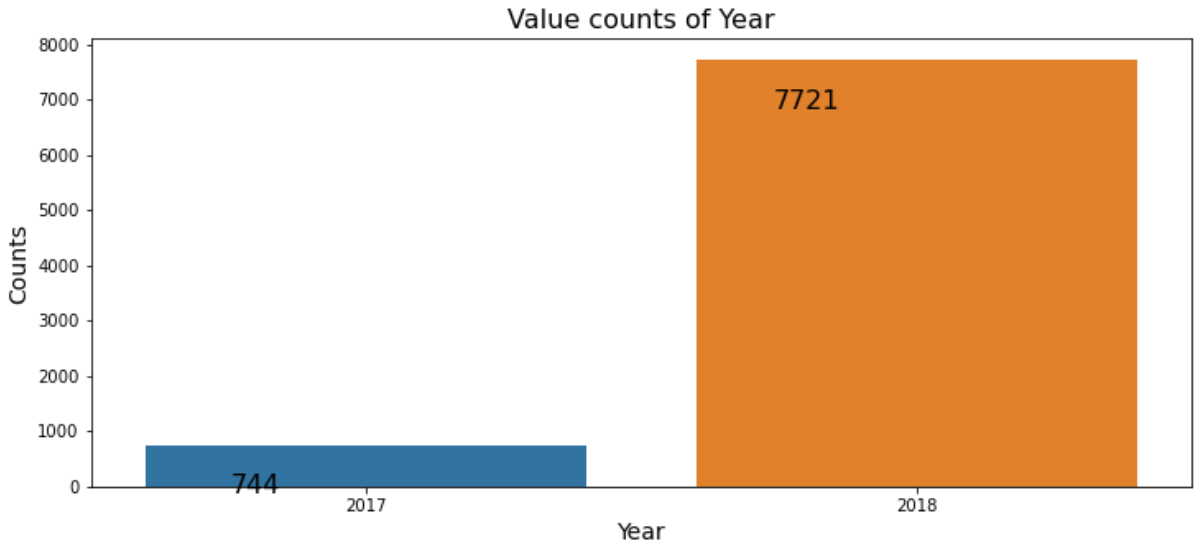
The boxplot shows that there are no outliers in the rented bike count feature. It has a skewness of 1.1397. The data is extended towards the right tail of the figure.

Distribution of data in boxplot



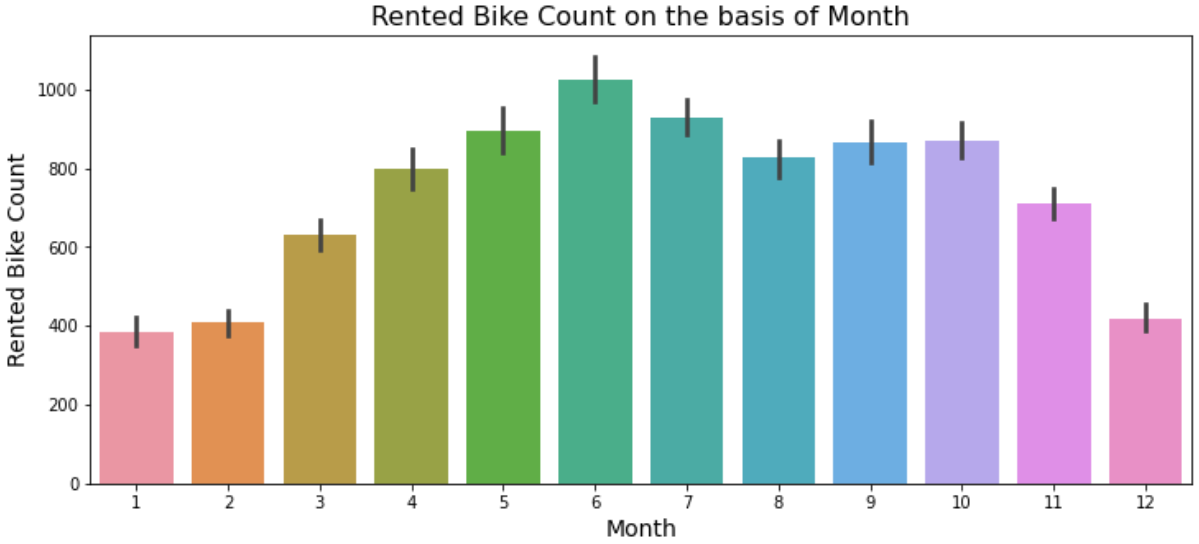
The above boxplot is drawn for the purpose of seeing the data distribution and for binary encoding of some features like Visibility, Solar radiation, Rainfall, Snowfall, Functioning day, Holiday.

Distribution data among the Year



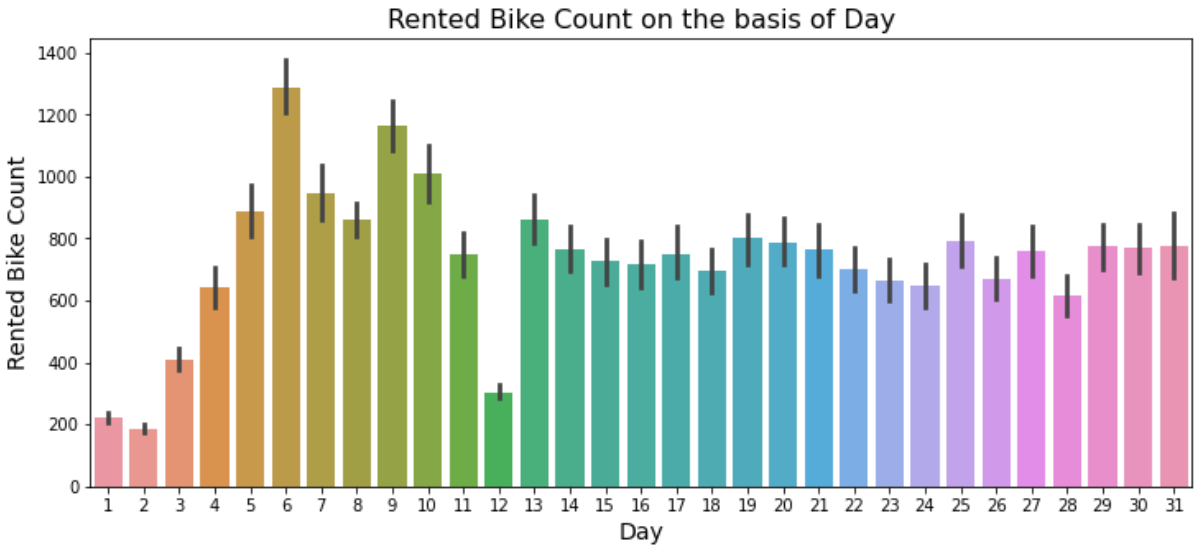
Most of the data is related to 2018.

Distribution data among the Months



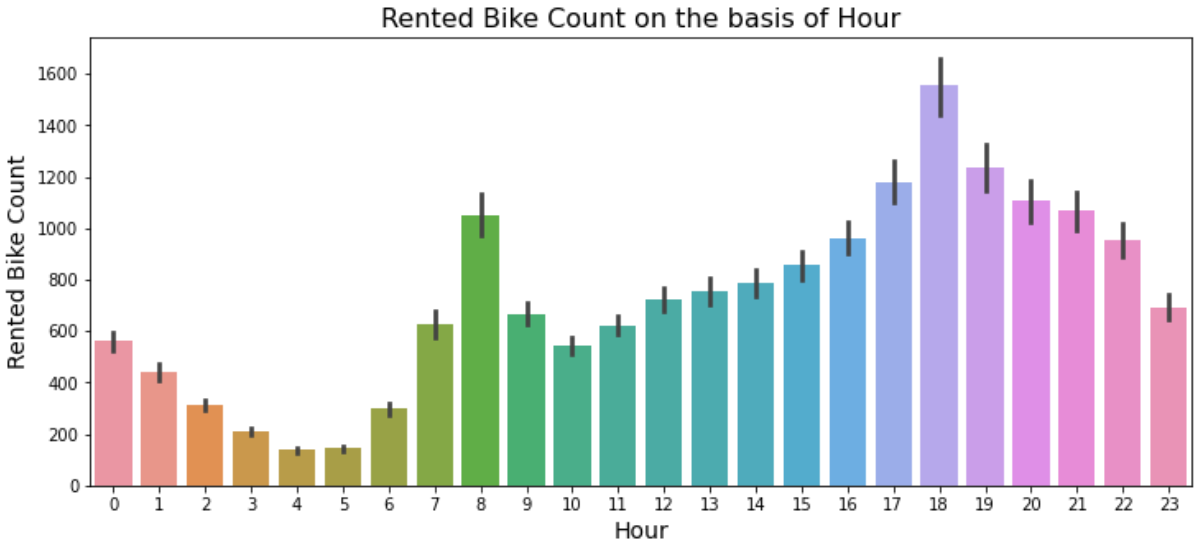
The above value counts of month shows that the demand is more in the month of June, May and July it seems that during the summer season. Less demand in the months of January, February and December.

Distribution data among the Days



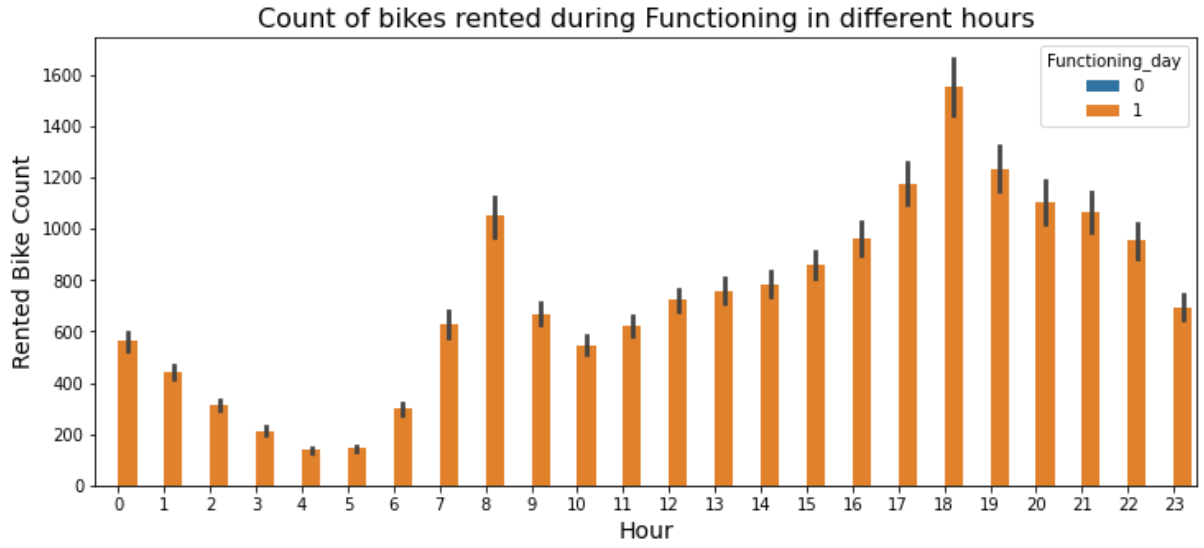
The demand is more during 5th, 9th and 10th dates. Low demand during 1st, 2nd and 12th dates.

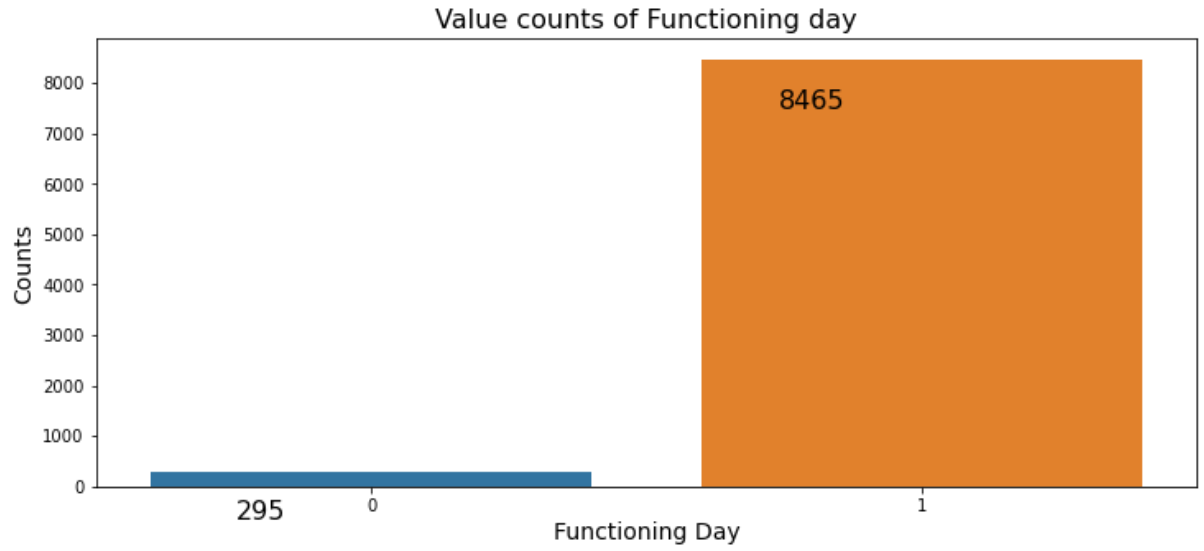
Distribution data among the Hours



The demand is more at 8 in the morning and 6 at evening time. It seems to be the starting and ending time of the office hours. Also the demand is more after office hours at evening time.

Functioning day vs. Rented Bike Count vs. Hours

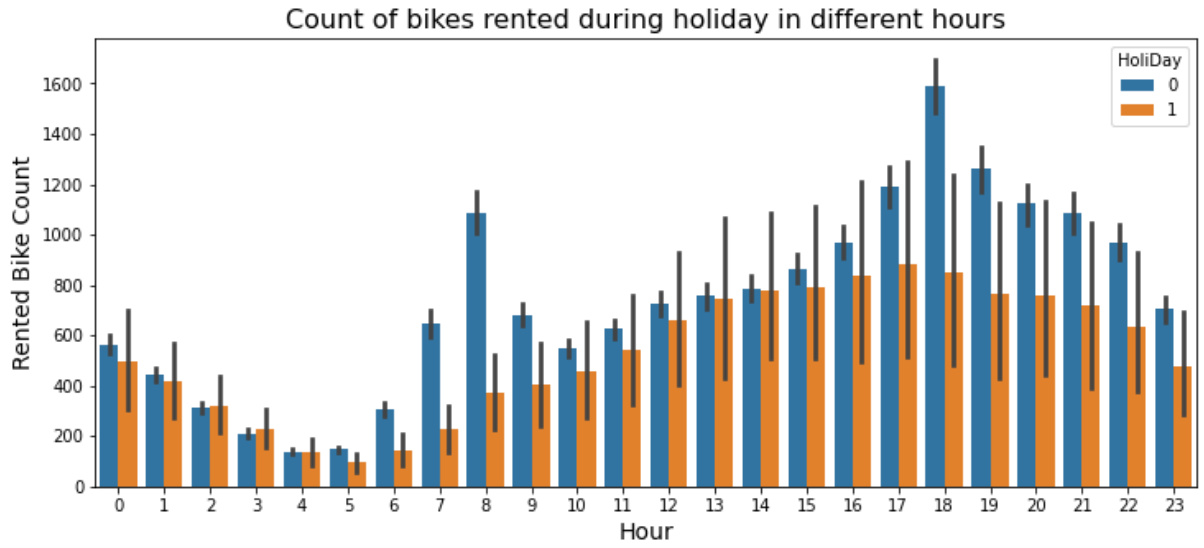


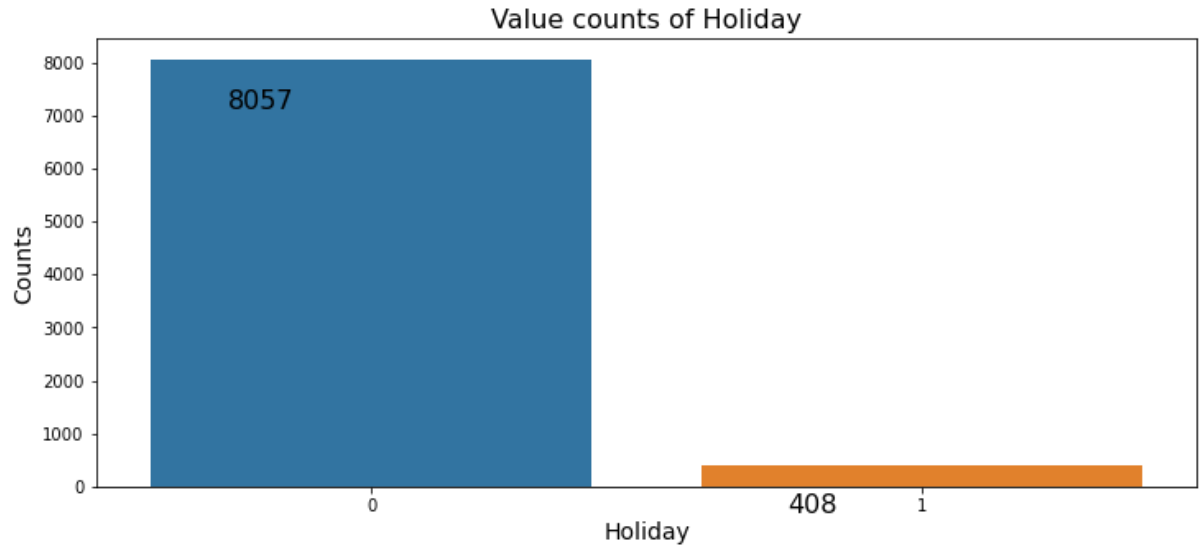


The above figure shows that the bike rentals are available only on the functioning day. The demand is more at 8 and 6 of the hours. It means the starting and ending time of the office hours. And demand is more also after office hours at evening time.

From the above analysis it's better to drop the non-functioning day data which is around 295 rows of data.

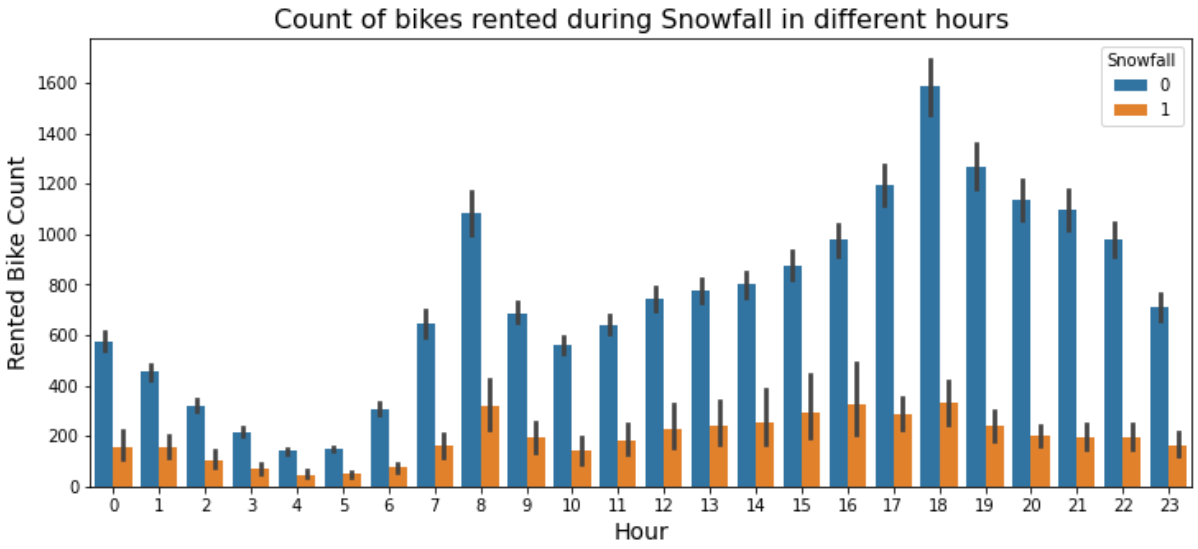
Holiday vs. Rented Bike Count vs. Hours

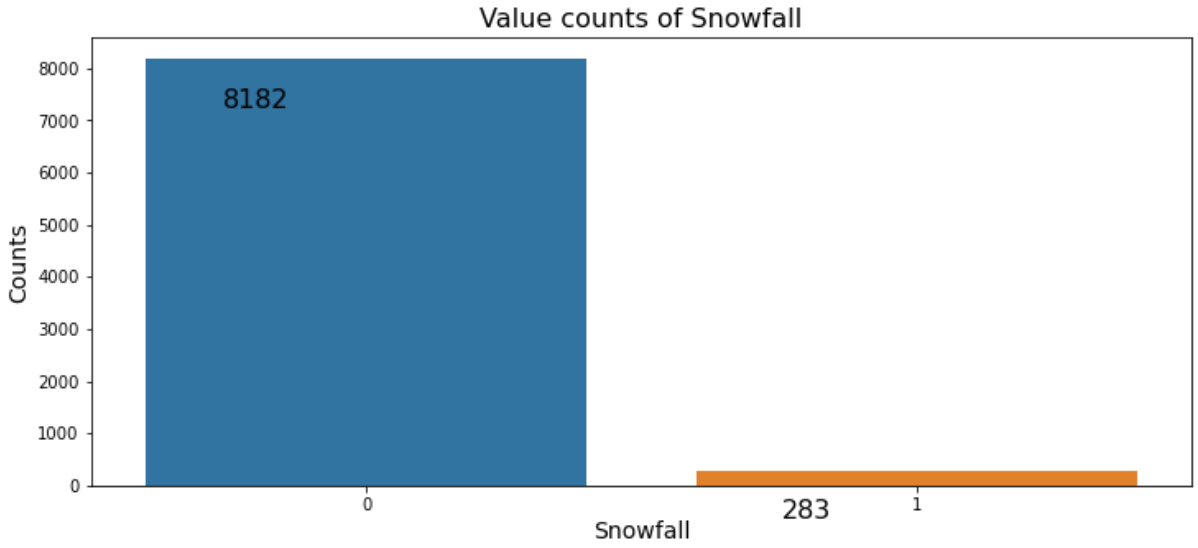




From the above figure we can say that there is more demand for bike rentals during non-holiday days when it is compared with the holidays. On holidays also there is more demand during evening hours.

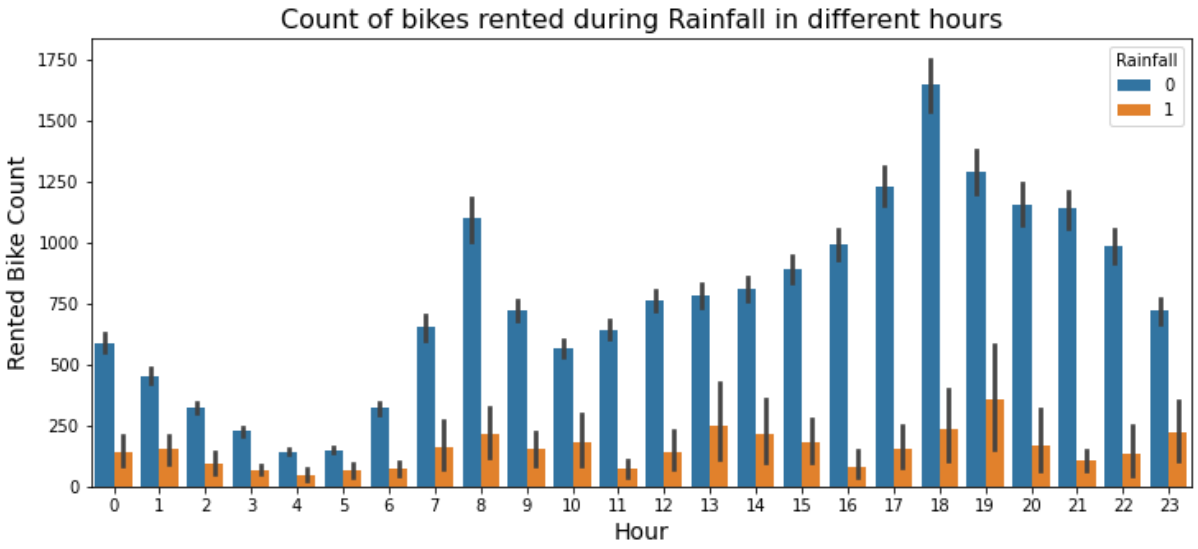
Snowfall vs. Rented Bike Count vs. Hours

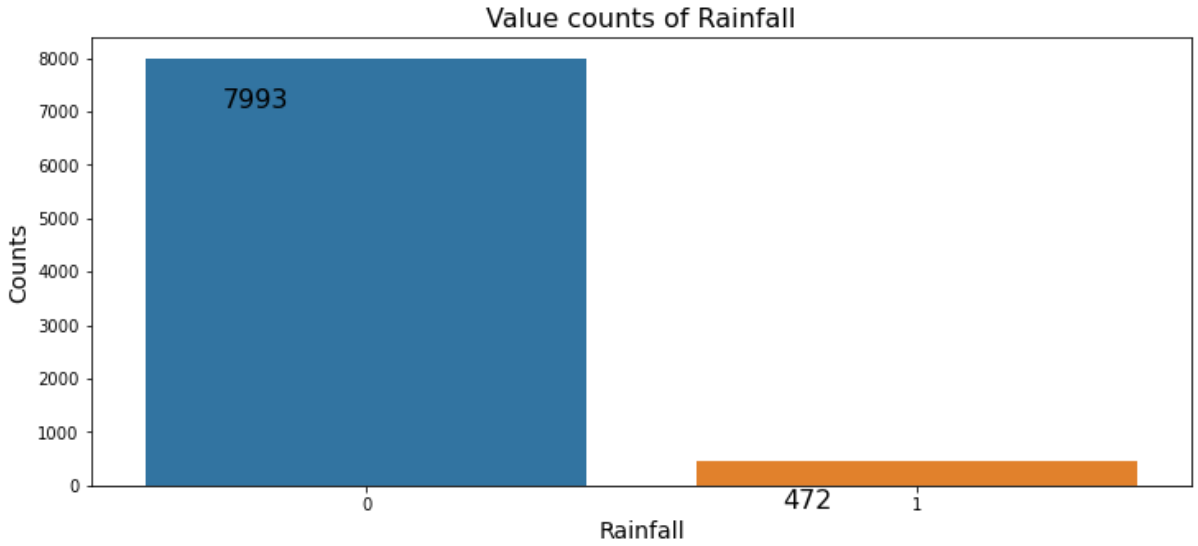




The demand is less during the snowfall time. And there is less demand during early morning hours in both the time.

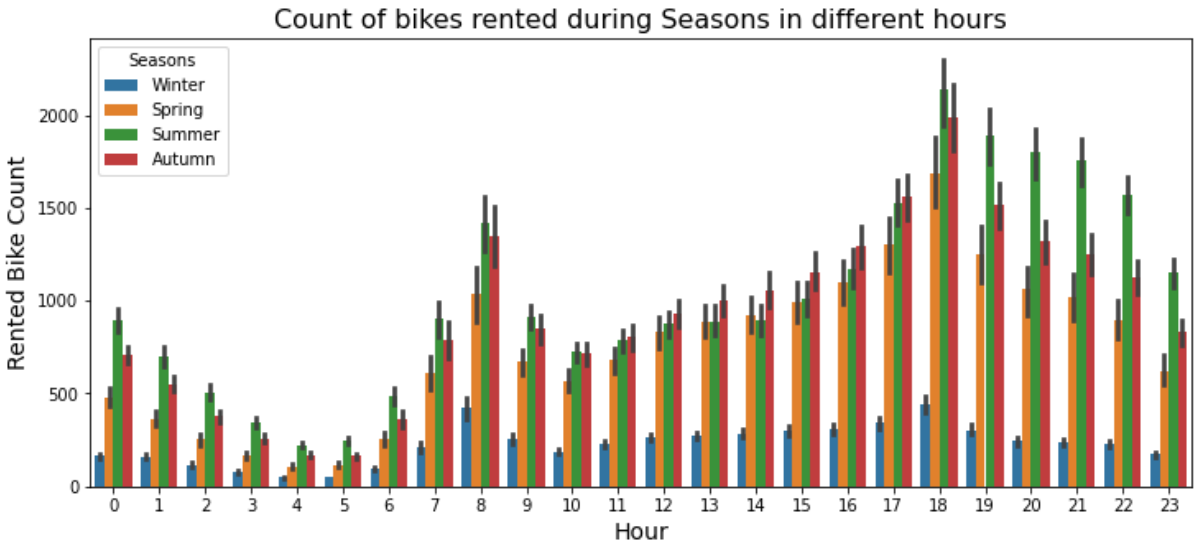
Rainfall vs. Rented Bike Count vs. Hours

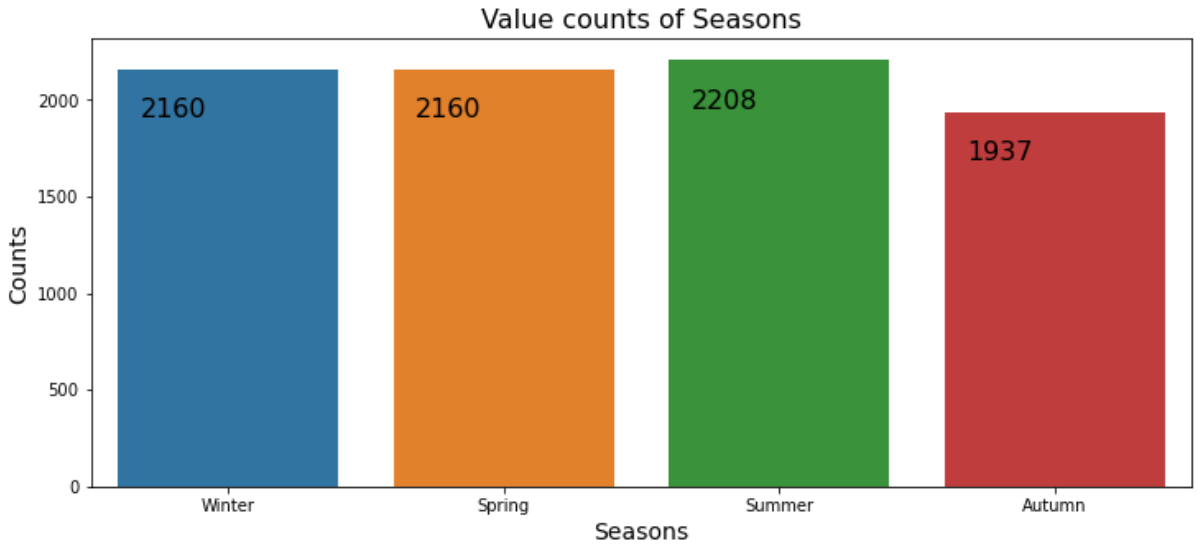




The demand is less during the rainfall time. Maybe only the emergency work people are using the services during the rainfall times.

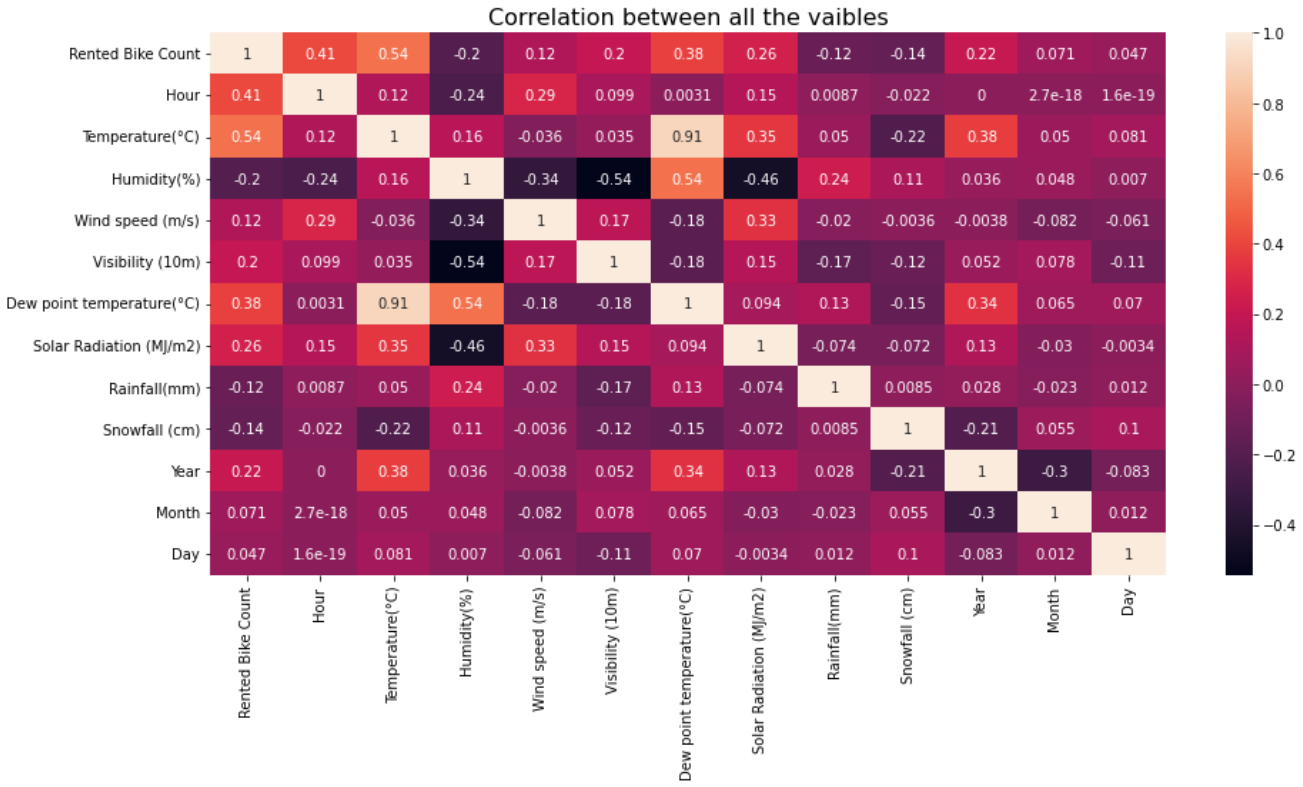
Seasons vs. Rented Bike Count vs. Hours





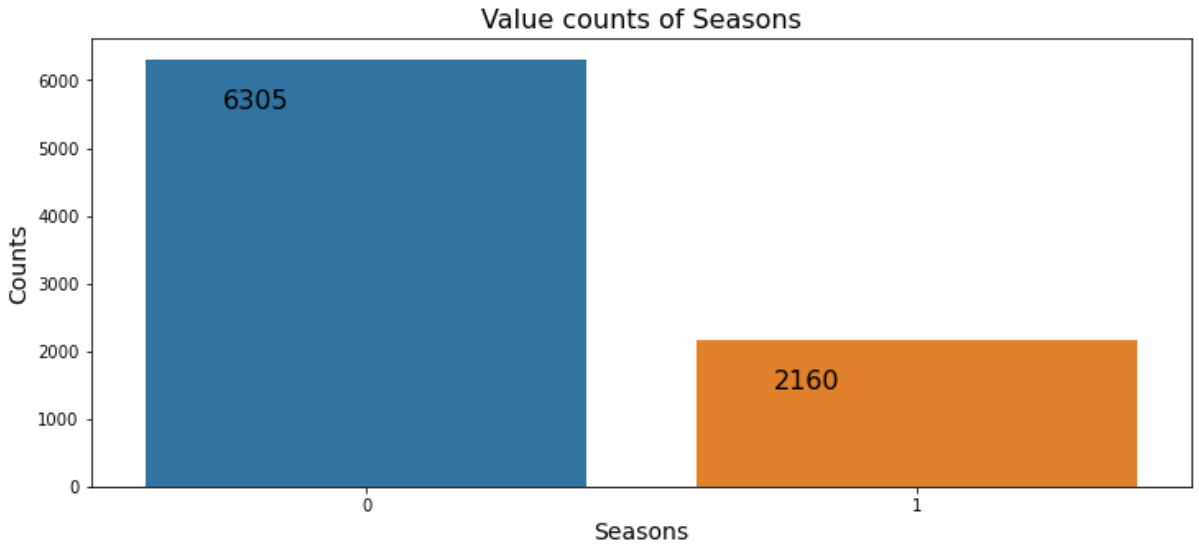
The value counts figure says that the demand is mostly equal in all the seasons and during the autumn season the demand is somewhat less and more demand during summer.

Visualization of correlation



The above picture shows the correlation between the different variables in the dataset. The colour in it is the scaling part of the relationship among them.

OneHot Encoding



By using the get_dummies option from pandas we are converting the seasons into binary format.

Variance Inflation Factor

The variance Inflation Factor is used when we have the multi-collinearity between the features. The factor helps in reduce the inflation between the features by dropping some of the features which are having the high correlation among them.

In the given dataset the correlation between the features is very high and some of the features are dropped to reduce the correlation among them.

Data Modelling

After the data preparation is completed it is ready for the purpose of analysis. Only numerical valued features are taken into consideration. The data was combined and labelled as X and y as independent and dependent variables respectively. The open, high and low columns are taken as independent variables (X) and the closing price is taken as dependent variable (y).

Splitting the data

The train_test_split was imported from the sklearn.model_selection. The data is now divided into 80% and 20% as train and test splits respectively. 80% of the data is taken for training the model and 20% is for the test and the random state was taken as 0.

Scaling the data

To normalise the data minmaxscaler was used from sklearn.preprocessing. It scales the data in the form of standard deviation of the feature multiplied with the difference of maximum and minimum, again it was added to minimum. At first the training data was made fit into the scaling function and test data is transformed now. The output we get are X_train, X_test, y_train, y_test.

```
#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, 12)
Size of X_test is: (1693, 12)
Size of y_train is: (6772,)
Size of y_test is: (1693,)
```

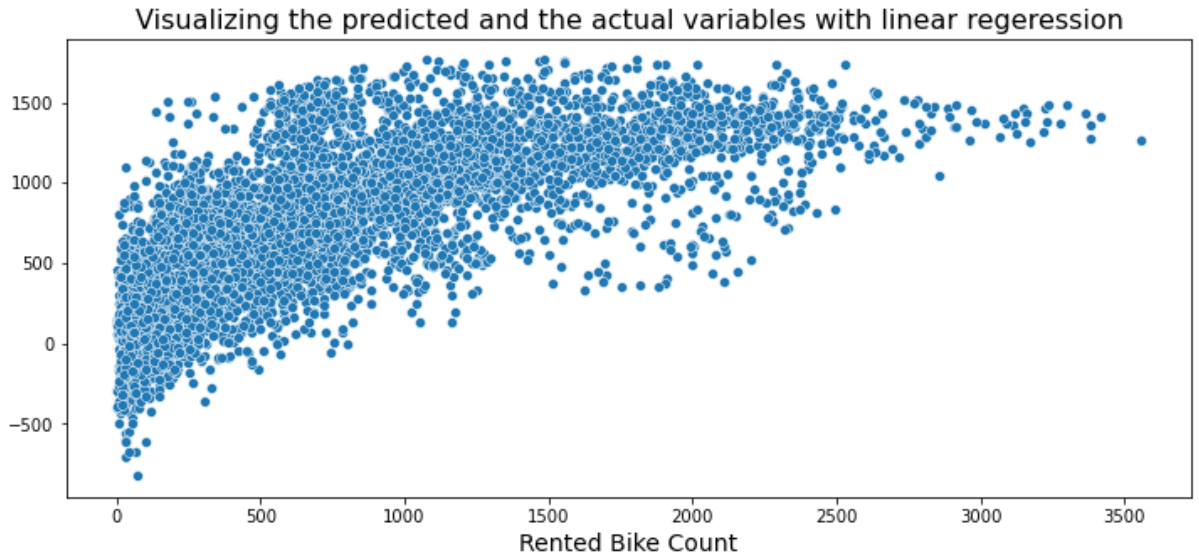
Linear Regression

The next step is implementing the algorithm and training the model. A linear regression is a type of statistical procedure that involves finding a relationship between a linearly-related variable and a prediction. Mathematically it solves problem in the form of:

$$\min_{w} ||Xw - y||_2^2$$

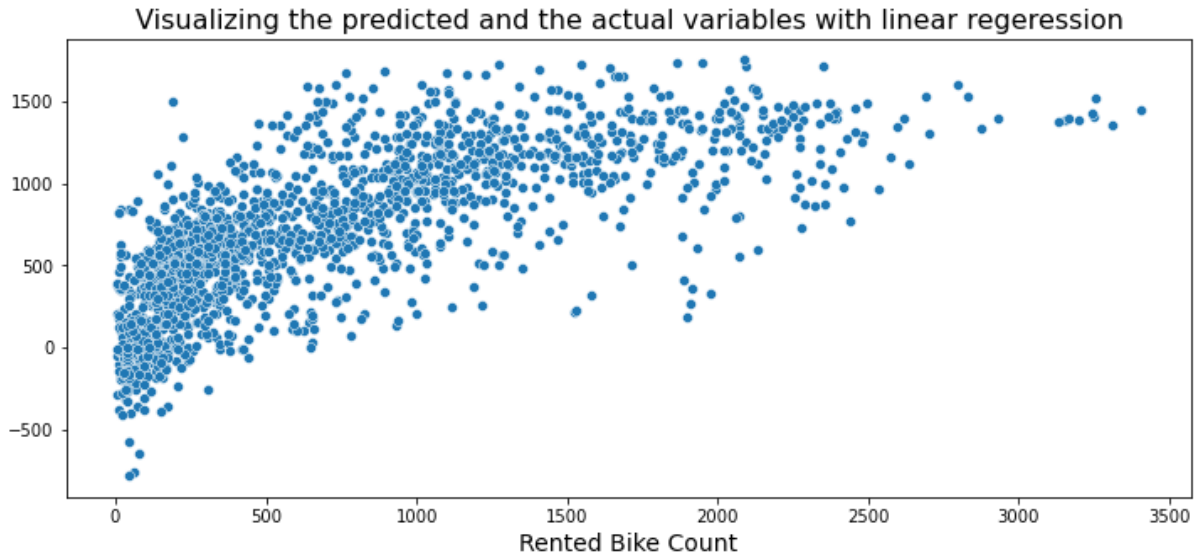
The first thing we do is to fit the training set and train the model. The score we obtained in training the model is 53.634%. To get the predicted data we fit the independent variables in the regression model as predicted. The available variables in hand at last are the actual y values and the predicted values. With these we can make the visualizations as follows.

Training data



The above figure shows how the model was trained and fit with the data. It consists of both predicted and actual training dataset in it. With the accuracy of 53.63% of the model can predict correctly what the demand for the bike rentals is.

Test dataset



The above figure shows how the model was trained and fit with the data. It consists of both predicted and actual data from the test dataset in it. With the accuracy of 51.34% of the model was trained to predict correctly what the demand for the bike rentals is.

Metrics

The metrics are tools used for evaluating the performance of a regression model. The following are the metrics that have been used in the analysis of the data.

- Mean Squared Error

  The mean squared error (MSE) is a commonly used metric for estimating errors in regression models. It provides a positive value as the error gets closer to zero. It is simply the average of the squared difference between the target value and the value predicted by the regression model.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

MSE score on training and test sets is 193487.67 and 191262.50 respectively.

- Root Mean Squared Error

  The root-mean-Square error or RMSE is a frequently used measure to evaluate the differences between the values that a model has predicted and the values that were observed. It is computed by taking the second sample moment and dividing it by the quadratic mean of the differences.

  RMSD is a non-zero measure that shows a perfect fit to the data, and it is generally better than a higher one. It is not used to evaluate the relationships among different types of data.

  RMSD is the sum of the average of all errors. It is sensitive to outliers.

$$\mathrm{RMSD}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})} = \sqrt{\mathrm{E}((\hat{\theta} - \theta)^2)}.$$

  RMSE score on training and test sets is 439.87 and 437.34 respectively.

- R2 score

  The goodness-of-fit evaluation should not be performed on the R2 linear regression because it quantifies the degree of linear correlation between the two values. Instead, the linear correlation should only be taken into account when evaluating the Ypred-Yobs relationship.

  The metric helps us to compare our current model with a constant baseline and tells us how much our model is better. The constant baseline is chosen by taking the mean of the data and drawing a line at the mean. R^2 is a scale-free score that implies it doesn't matter whether the values are too large or too small, the R^2 will always be less than or equal to 1.

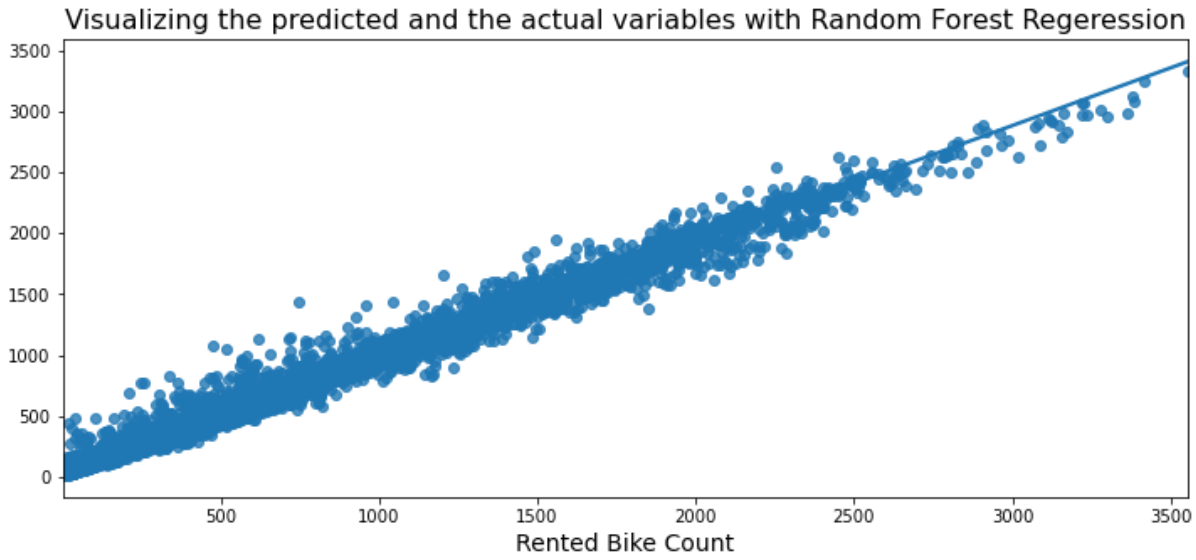
  R2 score on training and test sets is 0.5363 and 0.5133 respectively.

Random Forest Regression

Random Forest is a supervised learning algorithm which uses ensemble learning methods for statistical regression. Ensemble learning method is a technique that combines the predictions from multiple machine learning algorithms to make a more accurate prediction. A random forest operates by constructing several decision trees during training time and outputs the mean of the classes at the prediction of all the trees.

The model was imported and trained with the data available in the training dataset. Defined the predicted variable and checked the score of the model.

Training data



From the above figure we can see that the best fit line passes through the points and shows how good the model is trained and fits with the data. The model with the random forest regression has an accuracy of 98.22% on the training dataset. The below figure shows the result of the evaluation of the model.

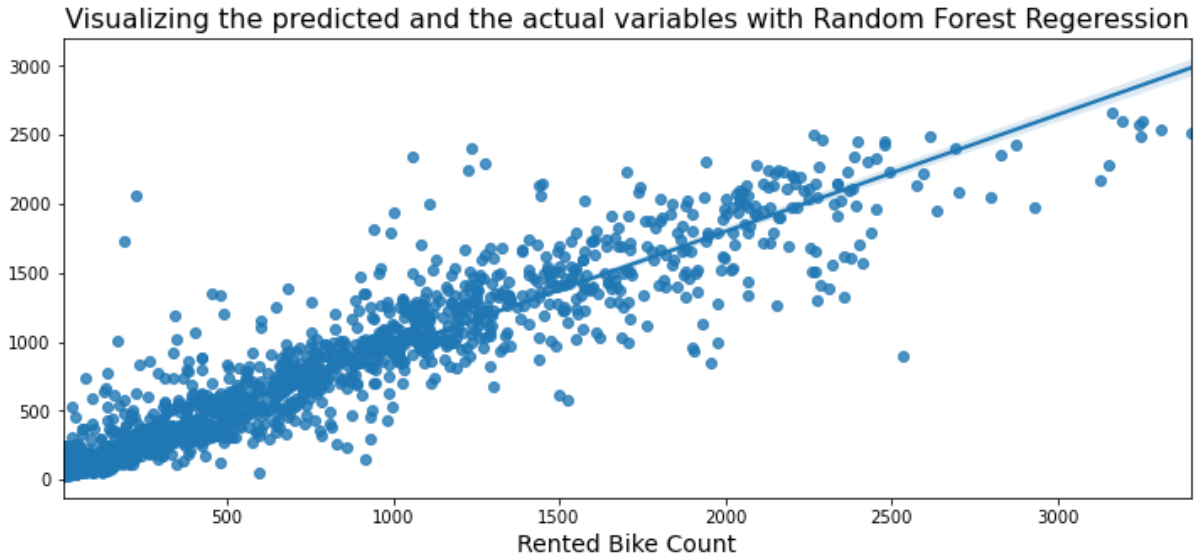
```
MSE_train = mean_squared_error(y_train, pred_train)
print(f'MSE= {MSE_train}')

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
```

Test dataset



The above figure shows how good the model has predicted and performed on the test data. It has an accuracy of 98.22% on the training data set and 86.79% on the test dataset.

Conclusion

- Linear Regression, Lasso Regression, Ridge Regression and Random Forest Regression are used to train the model.
- As per the evaluation it is better to implement the Random Forest Regression rather than going for Linear Regression, Lasso, Ridge Regressions.
- When it comes to the accuracy the Random Forest Regression is performing well on the test dataset with the accuracy of 86.79%. As it can predict what the demand is for bike rental with the same accuracy.