Capstone Project - 2

Supervised ML - Regression

Bike Sharing Demand Prediction

Anas Malik
Data Science Enthusiast, Almabetter

Project Flowchart

- Business Problem Statement
- Dependencies Required
- Data Inspection
- Data Wrangling
- Exploratory Data Analysis
- Data Pre-processing
- Model Implementation
- Cross Validation and Hyperparameter Tuning
- Conclusion

1. Business Problem Statement

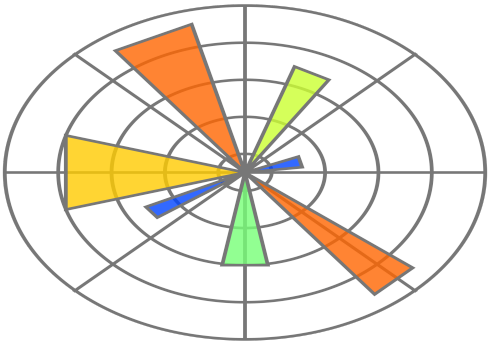
Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes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

The most challenging part for any rental company is to make rental bikes available to the public at the right time and right place. In this project our main aim is to solve this problem and predict the number of bikes demands across different hours and places.

2. Dependencies Required :

The dependencies of libraries for this project are as follows:

1. Numpy
2. Pandas
3. Matplotlib
4. Seaborn
5. Scikit-learn

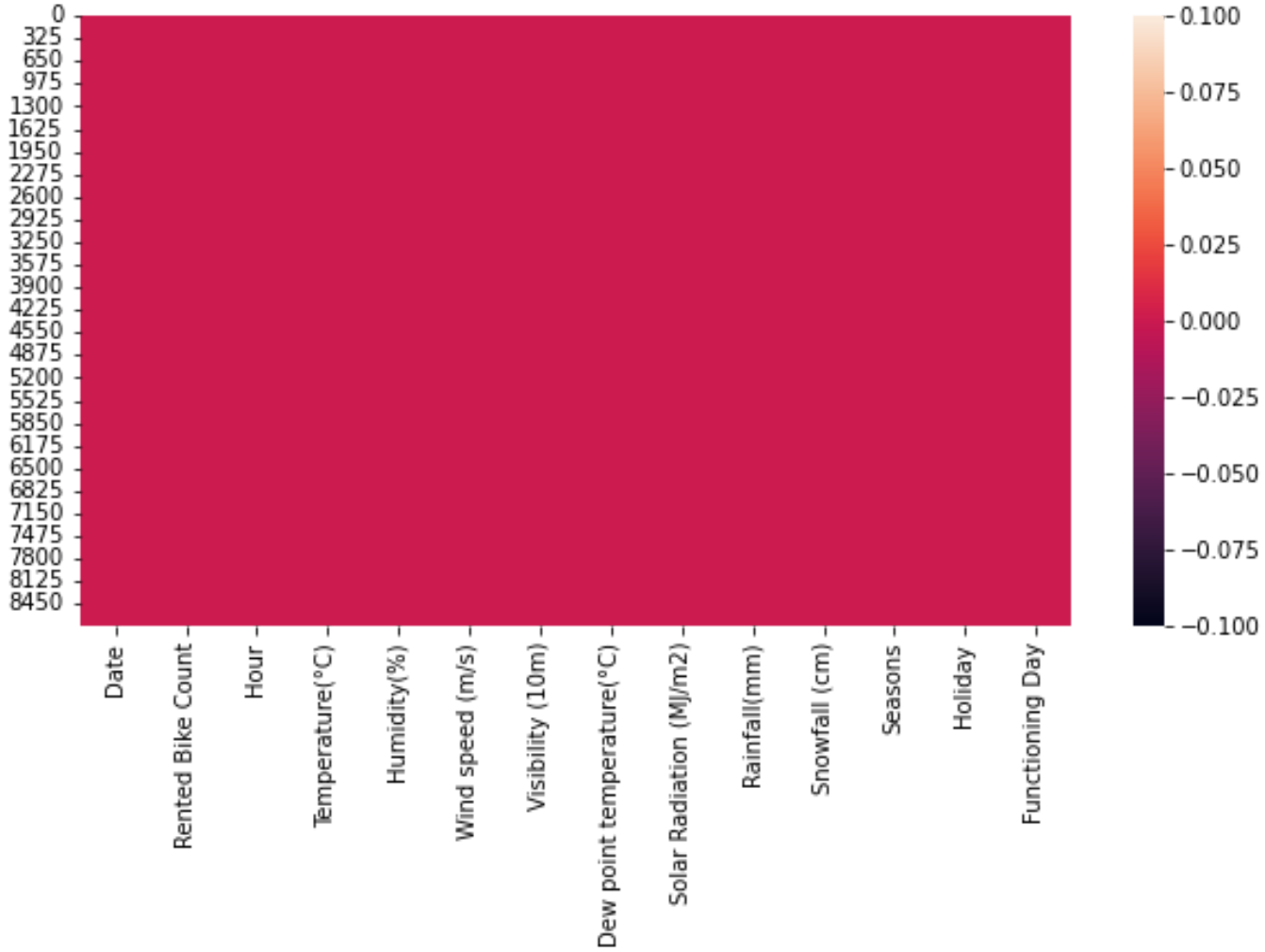


Numpy is famous for its n-dimensional arrays and mathematical operations. Pandas is famous for its Series and DataFrames. Matplotlib and Seaborn are famous for their different charts for visualization. Scikit-learn is famous for its inbuilt functions that help us create a Machine Learning model.

3. Data Inspection:

After inspecting the dataset, I came with some points that are follows:

- The data is of one year in Seoul(Capital of South Korea).
- There were no missing values.
- There were no duplicated entries as well.
- The dataset has 8760 rows and 14 columns.
- The target variable in the dataset was Rented Bike Count.



4. Column Descriptions :

There were total 14 columns which are as follows :

- Date - Date of the day.
- Rented Bike Count - Number of bikes rented.
- Hour - Hour of day when bike rented.
- Temperature - Temperature of the day.
- Humidity - Humidity of the day in percentage.
- Wind Speed - Speed of the wind on that day.
- Visibility - Visibility of objects in meters.
- Dew Point Temperature - Temperature at the beginning of the day in celsius.
- Solar Radiation - Electromagnetic Rays from sum that day.
- Rainfall - Is that a rainy day ?
- Snowfall - Is that a snowy day ?
- Seasons - what was the season ?
- Holiday - Is that a holiday ?
- Functioning Day - Is that a functioning day ?

5. Data Wrangling :

First of all, I created a copy of the main dataset. Then, at first i rename some typical columns names which are as follows:

- Rented Bike Count - Rented_Bike_Count
- Temperature(°C) - Temperature
- Humidity(%) - Humidity
- Wind speed (m/s) - Wind_speed
- Visibility (10m) - Visibility
- Dew point temperature(°C) - Dew_point_temperature
- Solar Radiation (MJ/m2) - Solar_Radiation
- Rainfall(mm) - Rainfall
- Snowfall (cm) - Snowfall
- Functioning Day - Functioning_Day

Then I converted the datatype of the Date column from object to datetime. With the help of this Date column now, I created two new columns, one is Month(month in a year) and another one is day(day in a month). Now, with the help of the day column, I created one more column that is Weekend(whether the day was Saturday, Sunday or not). Then I drop two columns ,i.eDate and day as there is requirement of these two columns according to me.

6. Exploratory Data Analysis :

I Performed the EDA part in four steps which are as follows :

1. **Univariate Analysis** - Check distribution of each column.
2. **Bivariate Analysis** - Visualize each variable with the target variable.
3. **Multivariate Analysis** - Visualize two or more variables with the target variable.
4. **Relationship of Columns.**

- ## Univariate Analysis :

Rented_Bike_Count: More number of records are in between **0-500**. Positive Skewed distribution.

Temperature: More Records are in between **15-30**. Somehow Normally distributed.

Humidity: More records are in between **40-80**. Somehow normally distributed.

Wind_speed: More records are in between **1-2**. Positively Skewed.

Visibility: More records are in between **1800-2200**. Negatively Skewed.

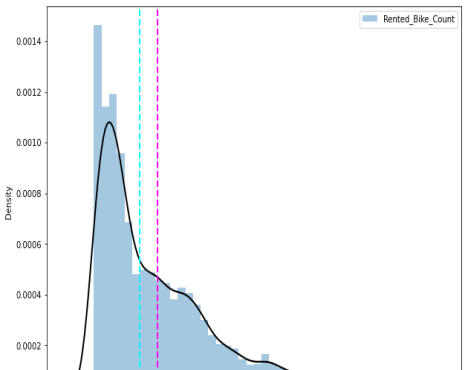
Dew_point_temperature: More records are in between **0-20**. Slightly negatively skewed.

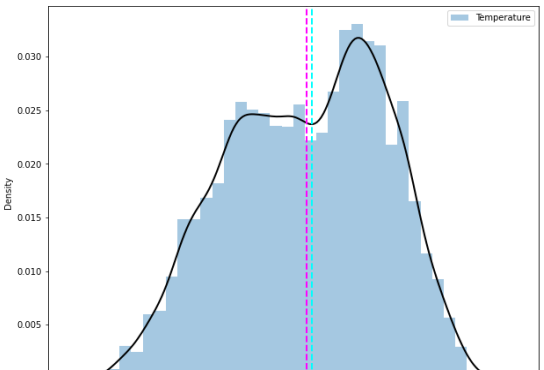
Solar_Radiation: More records are in between **0-0.5**. Positively Skewed.

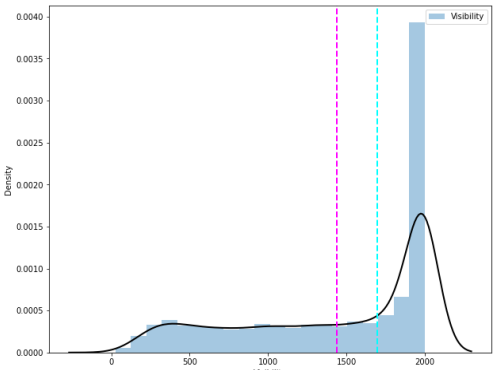
Rainfall: More records are around **0**. Positively Skewed.

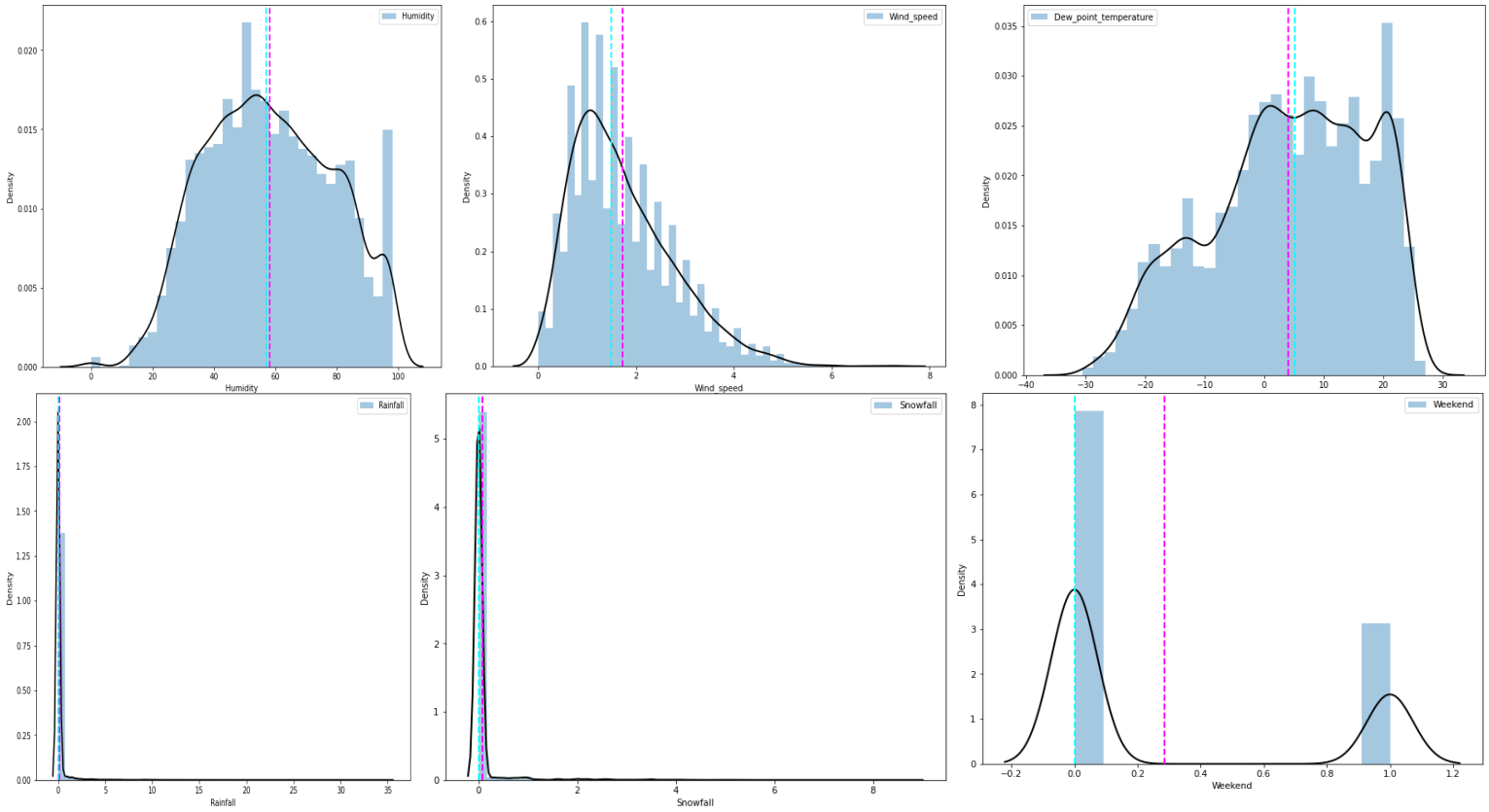
Snowfall: More records are around **0**. Positively Skewed.

Weekend: Only two records 0 and 1. More records are 0.







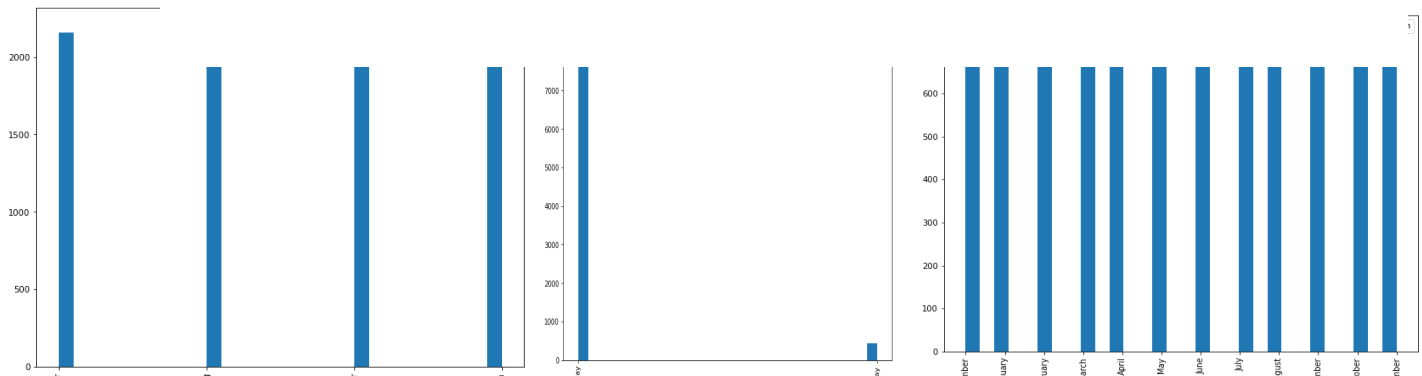


Seasons: There are four unique values ,i.e., Winter, Spring, Summer, Autumn. All of them have an equal number of records. This means people prefer renting bikes in all seasons.

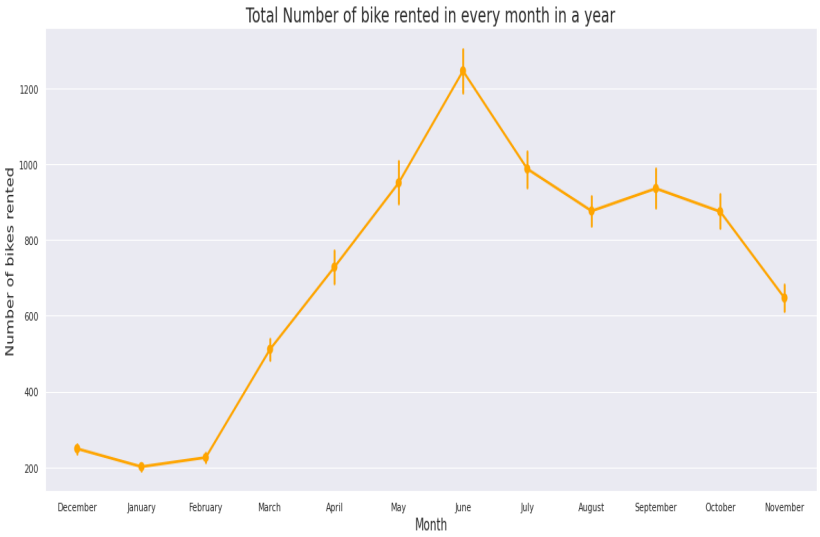
Holidays: There are only two unique values ,i.e., No Holiday, Holiday. No Holiday has a high number of records which means people prefer to rent bikes when they have some work and prefer to not rent bikes when there is a holiday.

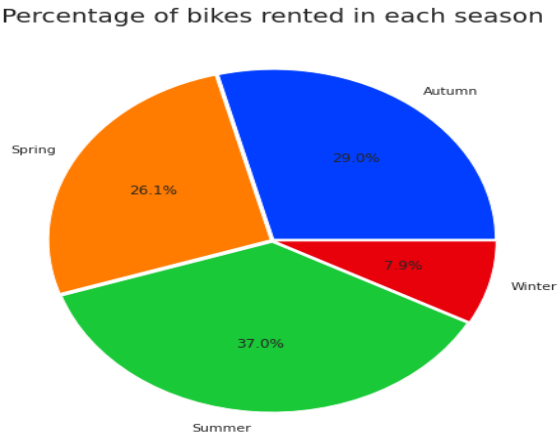
Functioning_Day: This is similar to Holidays columns and the same conclusion can be made for this ,i.e.. people rent bikes when it is a functioning day.

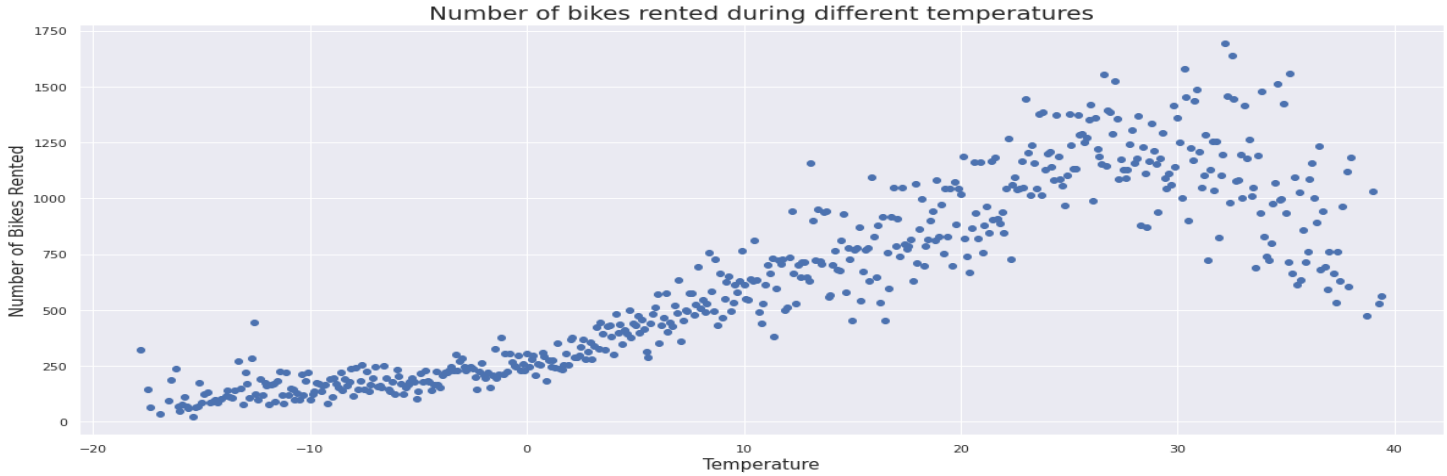
Month: This has 12 unique values and all of them have records. February has the lowest records. This column represents the months in a year.

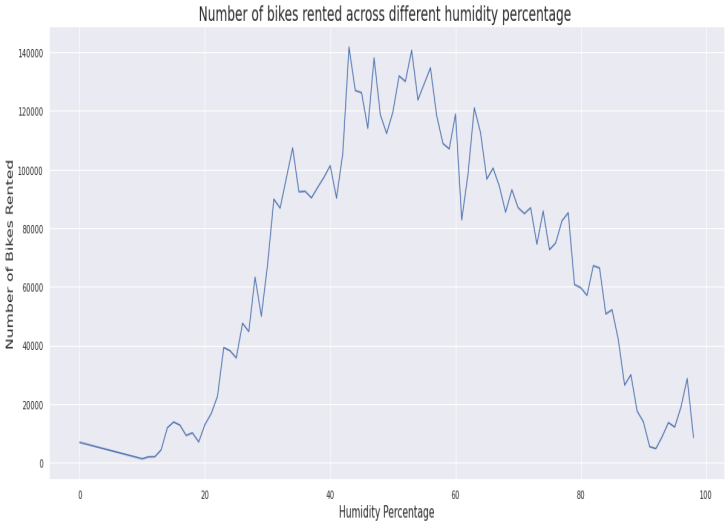


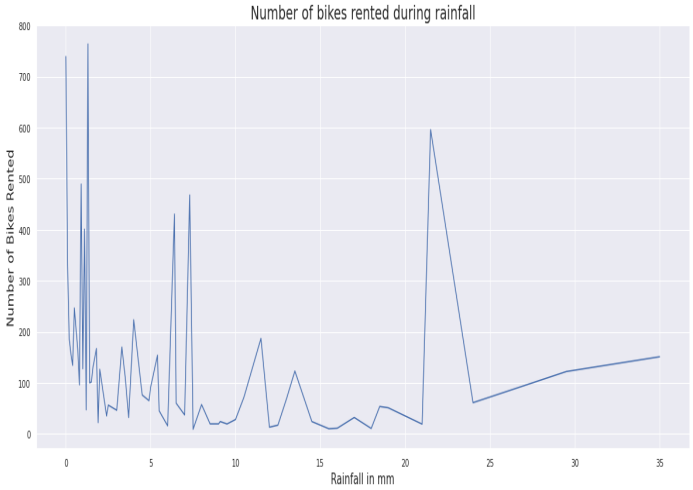
- **Bivariate Analysis :**

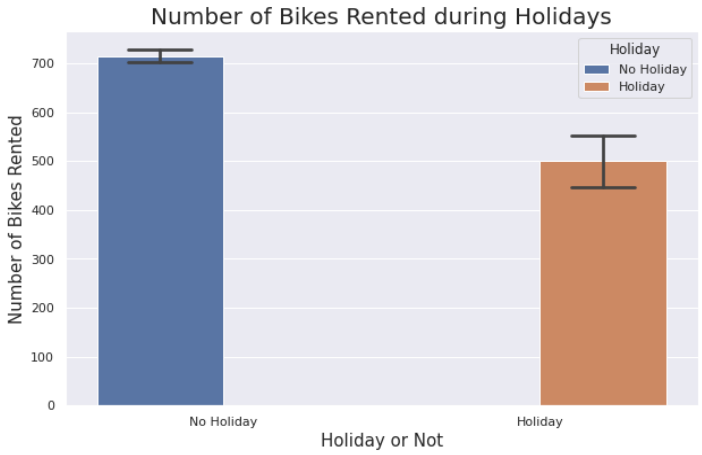


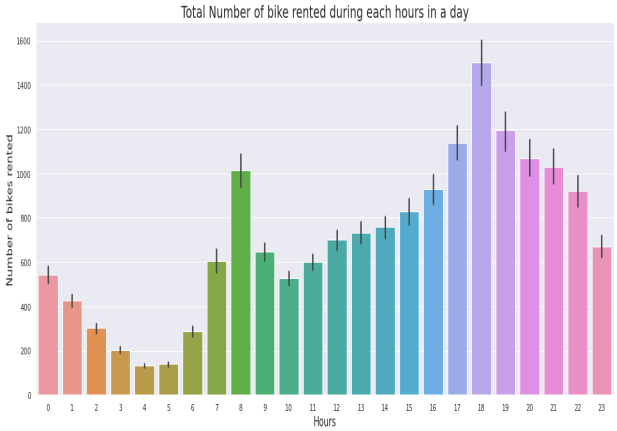




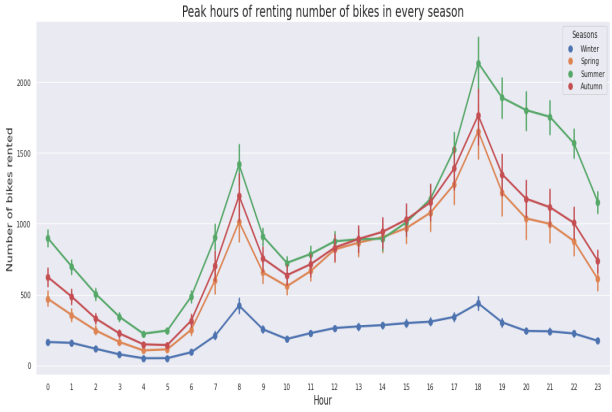


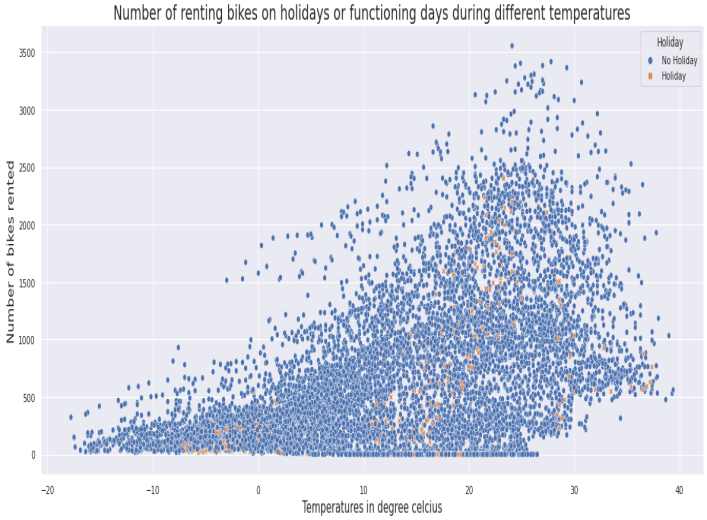


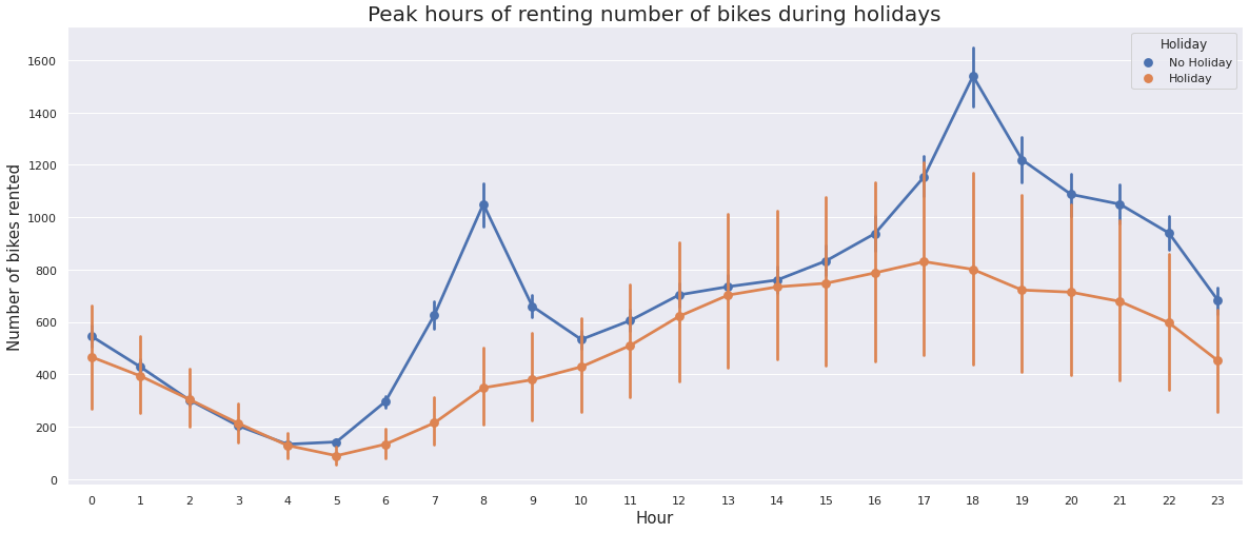


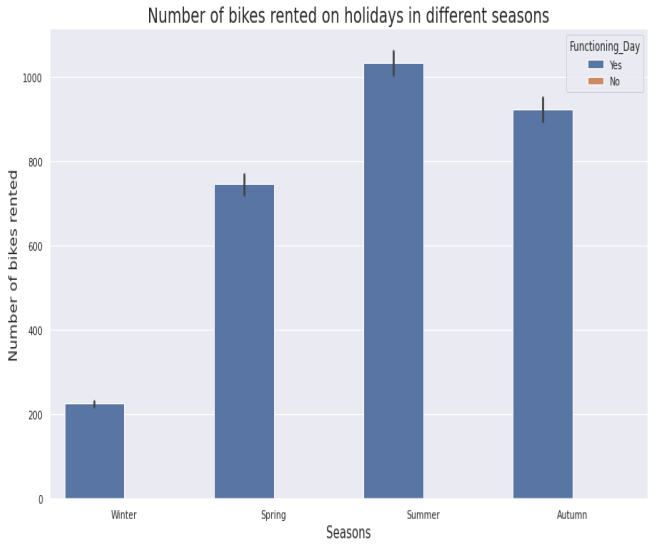


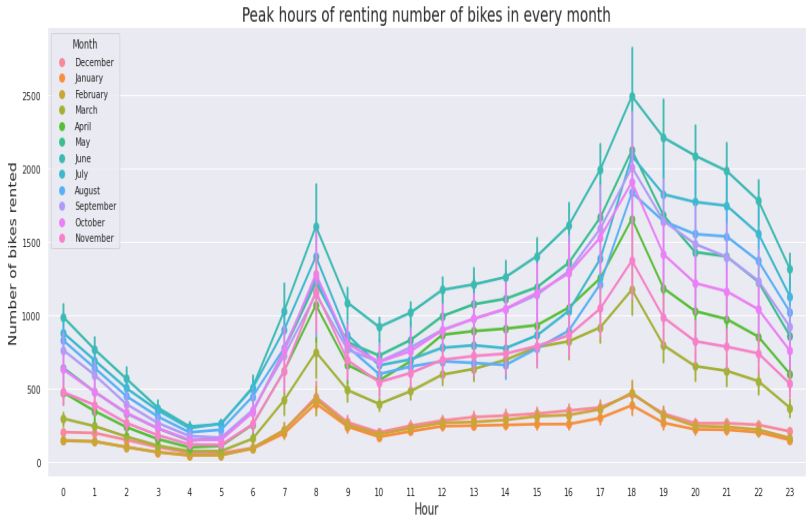
- **Multivariate Analysis :**











- **<u>Correlation Between Variables:</u>**

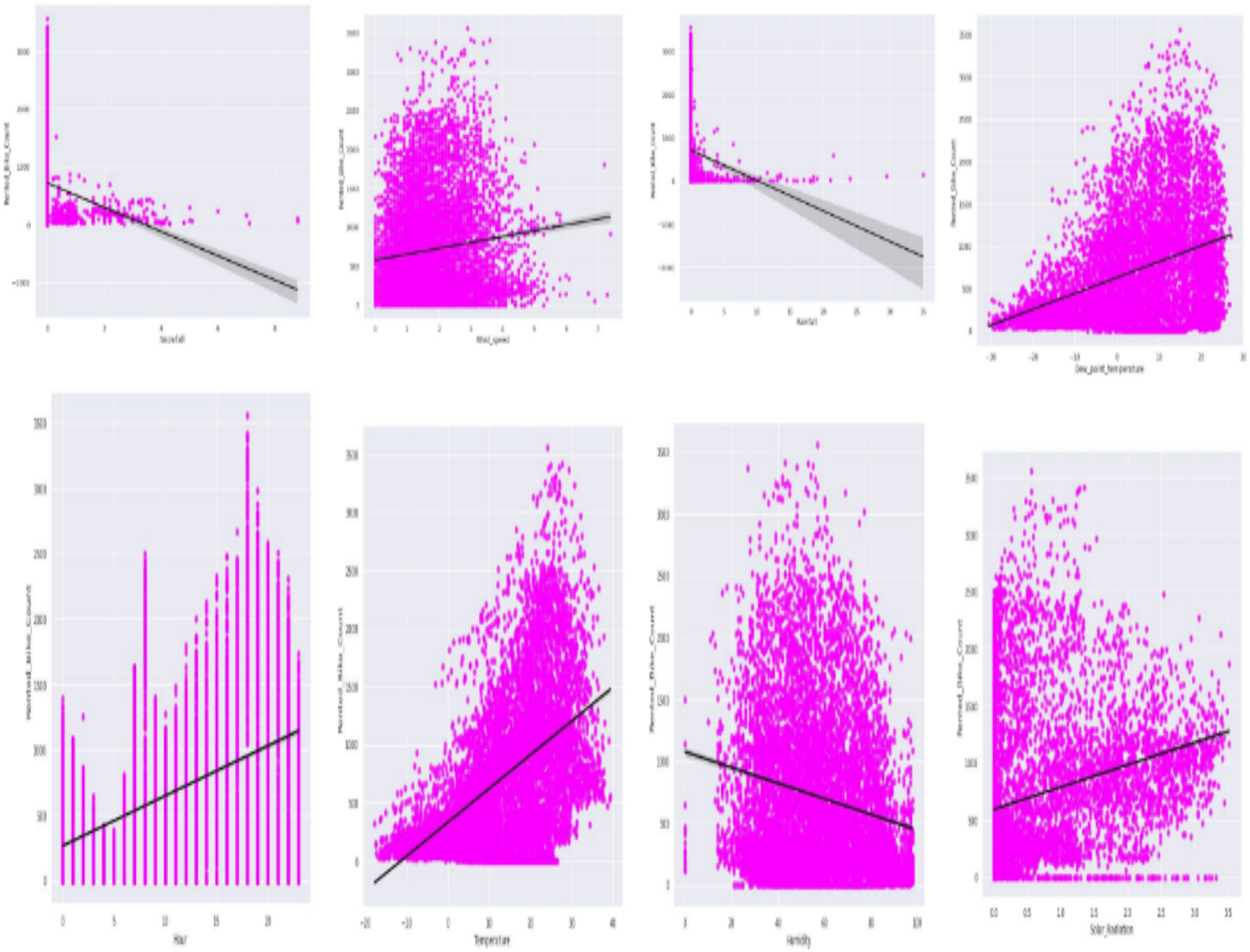
Positively Related to Target Variable :- Target variable increases on increasing the below variables.

- Solar_Radiation
- Hour
- Visibility
- Wind_speed
- Temperature
- Dew_point_temperature

Negatively Related to Target Variable :- Target variable decreases on increasing below variables.

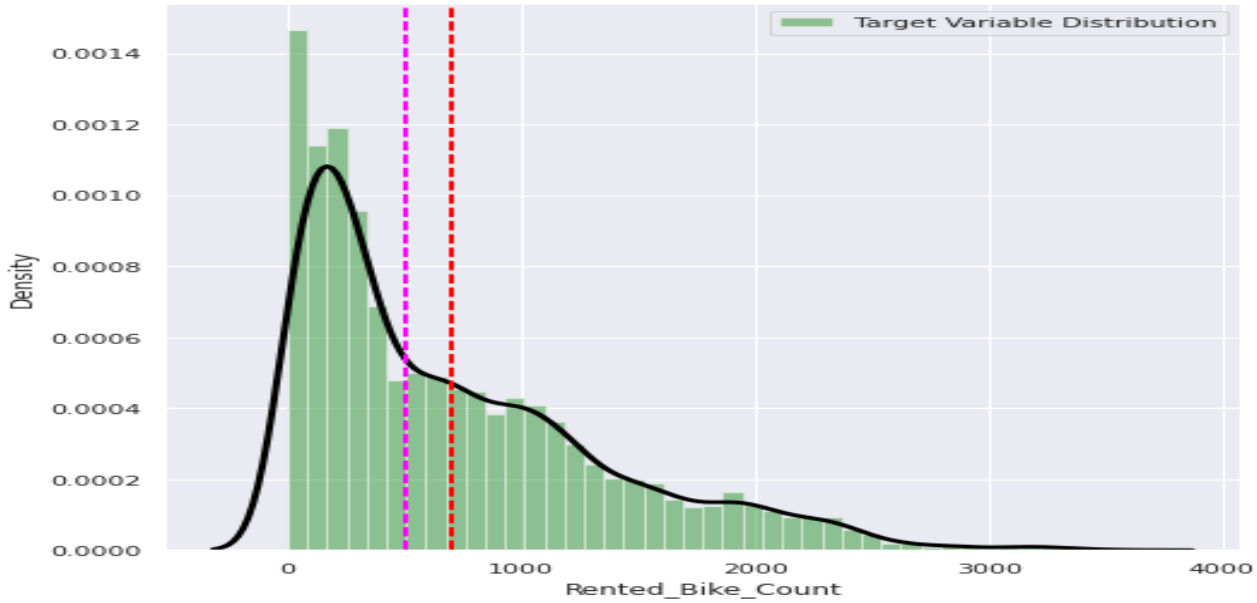
- Rainfall
- Humidity
- Weekend

- Snowfall

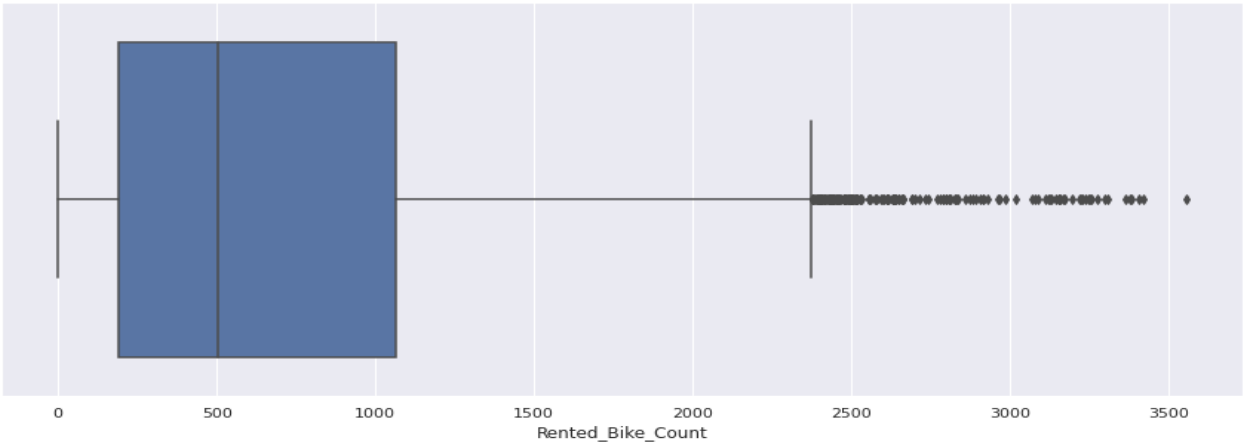


7. Data Pre-processing :

First of all I analyze the distribution of the target variable. To do visualization I used the seaborn library. The distribution of target variable lookalike.



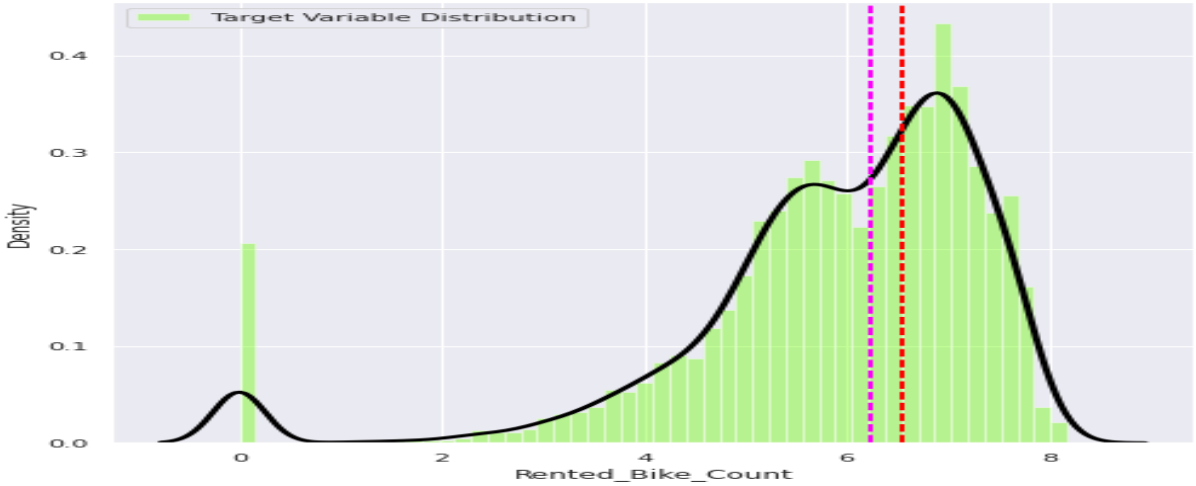
The distribution was positively skewed because there were some outliers present in the distribution as we can see in the below boxplot.



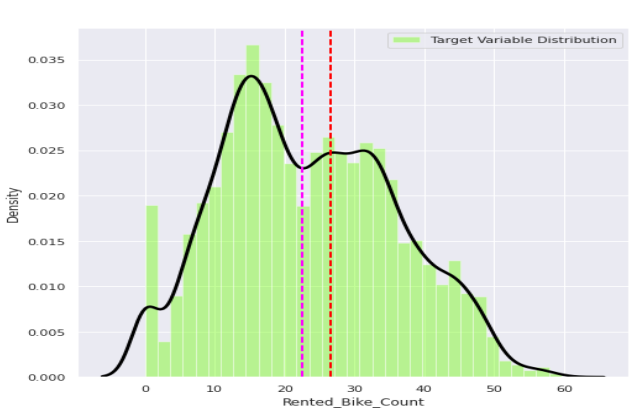
To make the distribution normal, we need to perform some transformation.There are three types of transformation we can apply to make it normal :

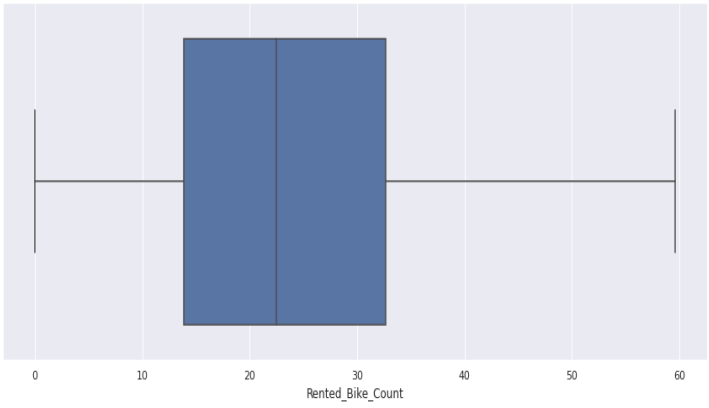
1. Log-Transformation
2. Square Root Transformation
3. Inverse Transformation

First I perform log transformation, and the distribution becomes negatively skewed as in figure.



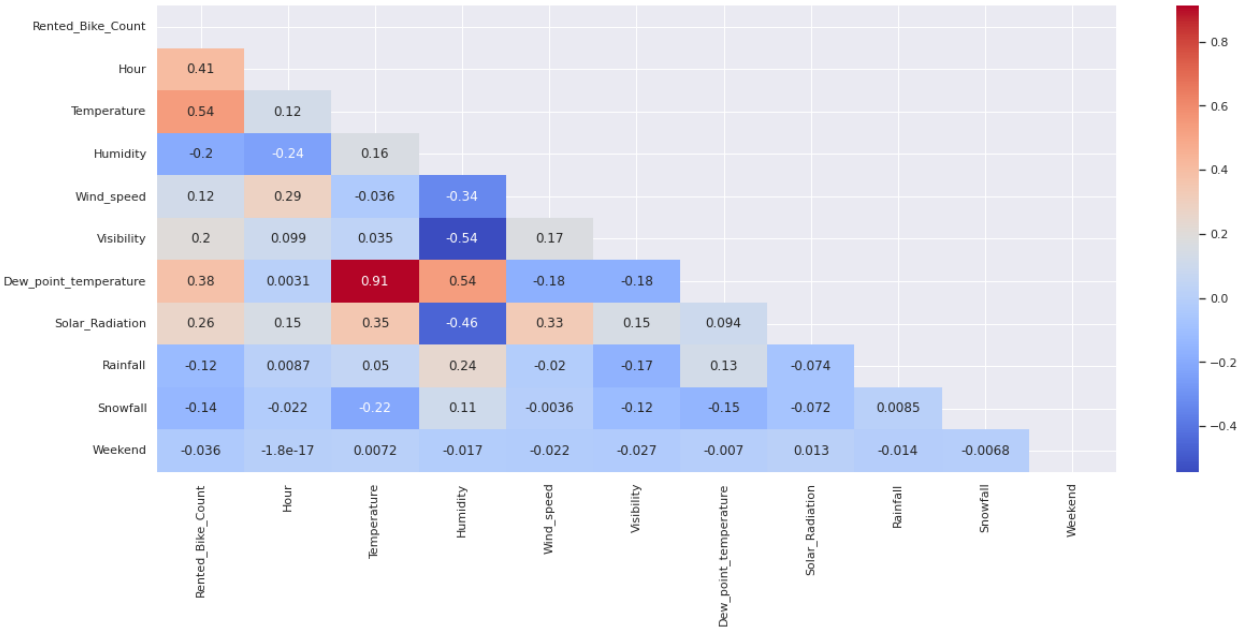
Then I apply Square Root transformation and the distribution becomes normal and there are no outliers present in the distribution.





- **Correlation between variables :**

From the below heatmap we can see that Dew_point_temperature is highly correlated with Temperature. So we can remove either one of them. I decided to remove Dew_point_temperature because I don't think people choose to rent a bike after observing morning temperature.



- **Removing Multicollinearity :**

To remove multicollinearity, I used variance inflation factor. If the value of VIF is greater than 10 then we may consider that there is still multicollinearity between variables.

	Columns	VIF_value
0	Visibility	9.106191
1	Rainfall	1.081868
2	Wind_speed	4.809775
3	Hour	4.418398
4	Weekend	1.409388
5	Solar_Radiation	2.882383
6	Temperature	33.984042
7	Snowfall	1.120882
8	Dew_point_temperature	17.505235
9	Humidity	5.617480

As we can see that the Dew_point_temperature and temperature has a VIF value greater than 10. So I decided to remove Dew_point_temperature as the temperature at the beginning of the day doesn't have much impact on renting a bike.

	Columns	VIF_value
0	Visibility	4.738121
1	Rainfall	1.079752
2	Wind_speed	4.608625
3	Hour	3.930173
4	Weekend	1.378871
5	Temperature	3.230140
6	Snowfall	1.120665
7	Solar_Radiation	2.254781
8	Humidity	5.016930

After removing Dew_point_temperatre, all the columns have VIF value less than 10. Therefore, it's a green sign to proceed with the encoding part.

- **Feature Encoding :**

I separated all of the categorical columns to do feature encoding. Since all of the columns were nominal in nature so I proceed with one hot encoding rather than label encoding. Then I splitted the data into two parts : training data(75 %) and the test data(25%).

The train test split is used to split the data into training set and testing set. The training set is used to train machine learning model and then test set is used to test the accuracy of the model on the unseen test data.

```
[ ] # Assigning the values in two part X and y
    X = data.drop(columns=['Rented_Bike_Count'], axis=1)
    y = np.sqrt(data['Rented_Bike_Count'])
```

```
[ ] X.shape, y.shape
```

```
    ((8760, 47), (8760,))
```

```
[ ] # Splitting the data into training set and test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

```
[ ] print("Shape of train data : ", X_train.shape, y_train.shape)
    print("Shape of test data : ",X_test.shape, y_test.shape)
```

```
    Shape of train data :  (6570, 47) (6570,)
    Shape of test data :  (2190, 47) (2190,)
```

- **Feature Scaling :**

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing.

There are many ways to do feature scaling. Two of them are :

1. Standardization - To standardize the data having mean 0 and standard deviation 1.
2. Normalization - To normalize the data in the range of 0 and 1.

I am going to use the standardization method with the help of a function StandardScaler() of the scikit-learn library.

8. Model Implementation :

Machine learning models are the programs written which help to find patterns or trends within the data and try to predict the prediction on unseen data.

Because we are dealing with a Supervised Machine Learning Regression(When a dependent variable is present and continuous in nature) problem, we will use some below listed regression algorithms.

- LinearRegression
- Lasso(L1 regularization)
- Ridge(L2 regularization)
- ElasticNet Regression
- RandomForest Regression

❖ Linear Regression:

Training Metrics :

Mean Square Error	35.07755090622306
Root Mean Square Error	5.922630404323999
Mean Absolute Error	4.474055591986692
R2 Score	0.7722099078993463

Testing Metrics :

Mean Square Error	33.27390585673638
Root Mean Square Error	5.7683538255499185
Mean Absolute Error	4.410100719860122
R2 Score	0.7722099078993463

After doing cross validation I received the same evaluation metrics which means there is no overfitting.

❖ Lasso Regression :

Evaluation Metrics

Mean Square Error	33.87405315907444
Root Mean Square Error	5.820142022242623
Mean Absolute Error	4.457706017227814
R2 Score	0.785561660949612

CV Evaluation Metrics

Mean Square Error	33.87405315907444
Root Mean Square Error	5.820142022242623
Mean Absolute Error	4.457706017227814
R2 Score	0.785561660949612

As we can compare both metrics of testing data, we can see there is no overfitting.

❖ Ridge Regression :

Evaluation Metrics

Mean Square Error	33.27663692231535
Root Mean Square Error	5.768590549026283
Mean Absolute Error	4.41028503807013
R2 Score	0.78934358054839

CV Evaluation Metrics

Mean Square Error	33.289046682617254
Root Mean Square Error	5.769666080685888
Mean Absolute Error	4.411305418312488
R2 Score	0.7892650210570101

As we can compare both metrics of testing data, we can see there is no overfitting.

❖ ElasticNet Regression :

Evaluation Metrics

Mean Square Error	34.14204279014603
Root Mean Square Error	5.843119268861969
Mean Absolute Error	4.4821264887616445
R2 Score	0.78386516330583

CV Evaluation Metrics

Mean Square Error	33.30551825703942
Root Mean Square Error	5.771093332899704
Mean Absolute Error	4.412070056234412
R2 Score	0.7891607484137558

In ElasticNet regression there is no overfitting as we can compare both of the metrics on testing data.

❖ RandomForest Regression :

Evaluation Metrics

Mean Square Error	12.692243668652257
Root Mean Square Error	3.5626175305036965
Mean Absolute Error	2.2110714544850123
R2 Score	0.919652258962123

CV Evaluation Metrics

Mean Square Error	13.964937998034555
Root Mean Square Error	3.7369690924644474
Mean Absolute Error	2.360394257697525
R2 Score	0.9115955183993694

As we can compare both metrics of testing data, we can see there is no overfitting.

8. Conclusion :

- **EDA Conclusion :**

➢ The number of renting bikes increases as summer starts.
➢ The number of renting bikes decreases as the winter season starts.
➢ In summer, 37% people prefer to rent a bike and in winters, only 7.9% people prefer to rent a bike.
➢ Most people prefer to rent a bike when the temperature of the day is between 15 degree to 30 degree celsius.
➢ When the humidity percentage is in between 30 to 70, people prefer to rent a bike.
➢ Most of the bikes are rented when the sky is shiny or there is no rainfall or snowfall.
➢ Majority of the bikes are rented on a functioning day. On holidays, the majority of people prefer to stay home.
➢ When there is a functioning day, most of the bikes are rented during the daytime. The peak hour of renting a bike is 6 PM in the evening. The second highest peak hour of renting a bike is 8 AM in the morning.
➢ The lowest number of bikes are rented around 4 AM and 5 AM.

- **ML Model Conclusion :**

➢ Random Forest algorithm gave the best performance among all of the models applied.
➢ The Linear and Ridge regression both show 77% accuracy on the training dataset. On the test dataset, Linear Regression shows 77% accuracy and Ridge Regression shows 78% accuracy.

- The Lasso and Elastic Net Regression both show 76% accuracy on the training dataset and 78% accuracy on test dataset.
- After doing cross validation on Lasso, Ridge and Elastic Net regression, the accuracy for all of them remains the same on the test dataset. This shows no overfitting.


THANK
you!