Bike Renting

Submitted By

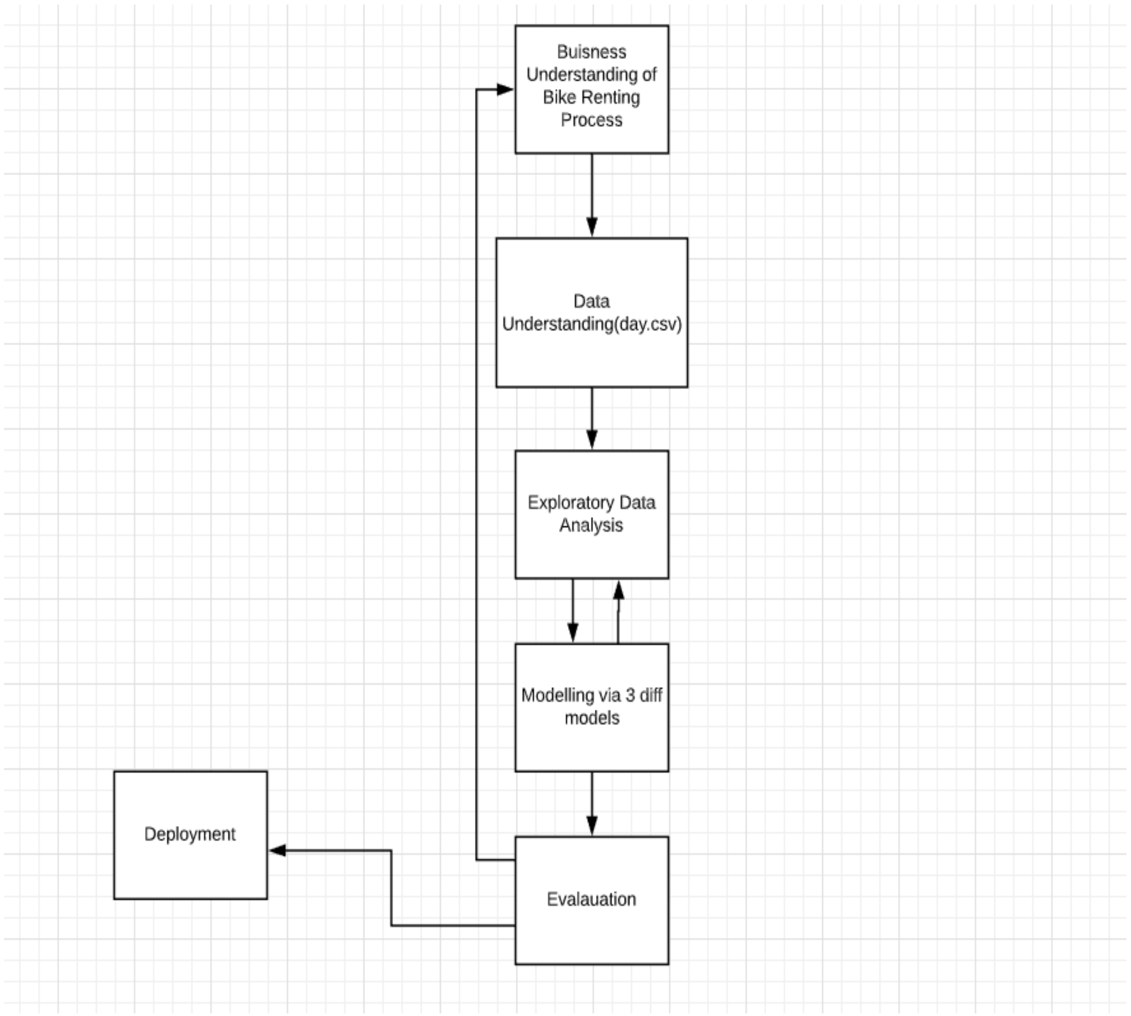
Abhiyu

Contents

Chapter 1

1. Introduction

1.1 **Problem Statement:** The objective of this Project is to predict the count of the bikes to be rented on daily basis. This count will take environmental and seasonal settings from the historical data in account while forecasting the daily demand.


Buisness Understanding of Bike Renting Process
Data Understanding(day.csv)
Exploratory Data Analysis
Modelling via 3 diff models
Deployment
Evalauation

Process Flow Representation via Lucid Chart Tool

1.2 Data: As the dataset given has dependent and independent values, it will come under supervised Machine learning. Our task is to build a Regression models which will help us predicting the count of bikes rented depending on the factors provided.

Given below is a sample of the data set that we are using for our prediction.

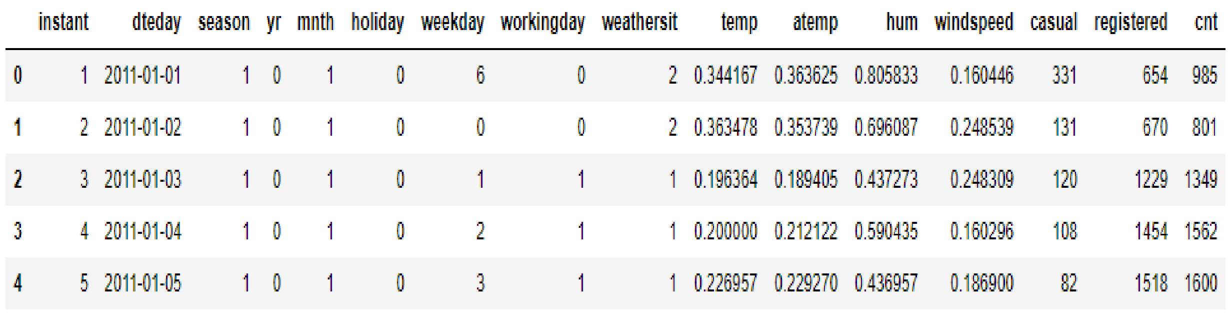
This dataset contains the rental count in between year 2011 and 2012 based on seasonal and environment etc.
The data given is having 731 rows and 16 columns/variables.

Out of these 16 variables two variables which is **casual**, **registered** are the target variables while their sum constitutes to be another target variable named as **cnt.**

The casual target variable contains total bikes acquired by the customers who are not already registered means at random they hired the bike.

While registered variable represents the hired number of bikes only by the persons who are already registered, and they are historical customers.

Thus, combining both the counts that is casual and registered builds "cnt" and it gives the total count of rented bikes on each particular date of given month and year.



	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

Variable	**Explanation**
Instant(rec_id)	Record id
Dteday(datetime)	Date time
season	Season (1:spring, 2:summer, 3:fall, 4:winter)
Yr(Year)	Year (0: 2011, 1:2012)
Mnth (Month)	Month (1 to 12)
holiday	weather day is holiday or not (extracted from Holiday Schedule)
weekday	Day of the week
workingday	If day is neither weekend nor holiday is 1, otherwise is 0.
weathersit	(extracted from Freemeteo) 1: Clear, Few clouds, Partly cloudy 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp	Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
atemp	Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_maxt_min), t_min=-16, t_max=+50 (only in hourly scale)
hum (Humidity)	Normalized humidity. The values are divided to 100 (max)
windspeed	Normalized wind speed. The values are divided to 67 (max)
casual	count of casual users
registered	The number of registered users at a given day
cnt (Total Count)	Total Rentals with both casual and registered users

2. Methodology

2.1 Pre-Processing: Before we proceed to create our model on top of the provided data. It is necessary to do Exploratory Data Analysis. Exploratory Data Analysis (EDA) is an approach to analyse the data sets to summarize their main characteristics. As the result depends on the data, EDA makes sure the quality of input data is high which will lead to high quality results. We can perform EDA as follows:

2.1.1 **Variable Identification:** In Order to understand the data, we need to first, Identifying Predictor (Input) and Target (output) variables. Then, Identifying the data type and category of the variables.

Renamed the variables for better understanding: -

```
df_bike.rename(columns={'instant':'rec_id','dteday':'datetime','yr':'year','mnth':'month','
weathersit':'weather_condition','hum':'humidity','cnt':'total_count'},inplace=True)
```

a) Types of Variable: Our Target Variable is 'total_count', and Predictor variables are (datetime,season,year,month,holiday,weekday,workingday,weathersit,temp,atemp, humidity, windspeed, casual, registered) .

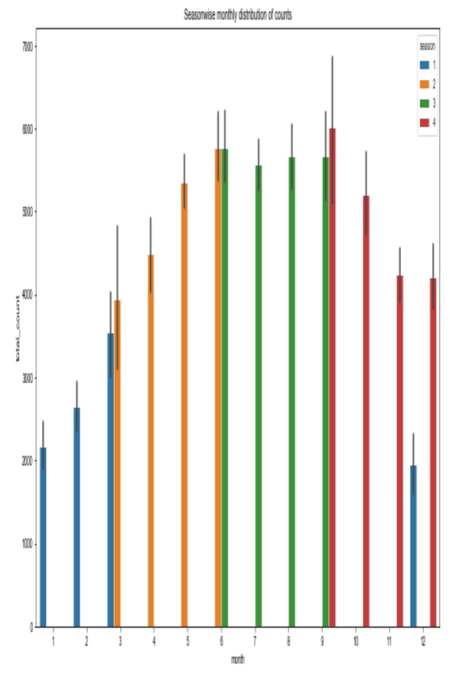
b) DataTypes:
Character(datetime),Numeric(rec_id,season,year,month,holiday,weekday,working day,weathersit,casual,registered,total_count) ,factor(temp,atemp,windspeed).

We need to change the datatype of our variables before starting anything as there are variables that are of category type but present in numeric type.

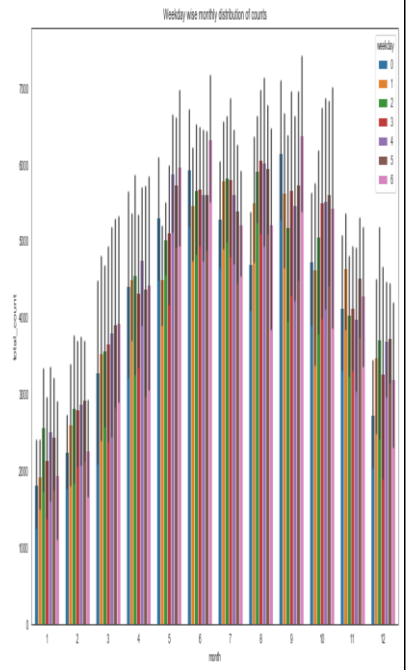
```
df_bike['datetime']=pd.to_datetime(df_bike.datetime)
df_bike['season']=df_bike.season.astype('category')
df_bike['year']=df_bike.year.astype('category')
df_bike['month']=df_bike.month.astype('category')
df_bike['holiday']=df_bike.holiday.astype('category')
df_bike['weekday']=df_bike.weekday.astype('category')
df_bike['workingday']=df_bike.workingday.astype('category')
df_bike['weather_condition']=df_bike.weather_condition.astype('category')
```

c) Variable Categories: Categorical (season, year, month, holiday, weekday, workingday, weathersit), Continuous (temp, atemp, humidity, windspeed, casual, registered)

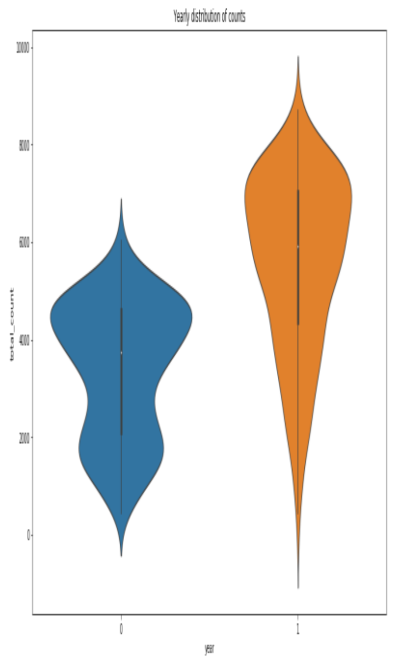
2.2 **Visualization:** Exploring the variable Distributions & Trends.



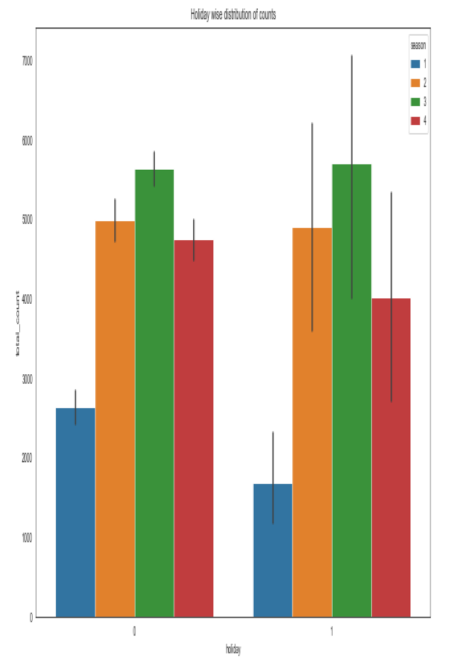
Season wise monthly distribution of Counts.



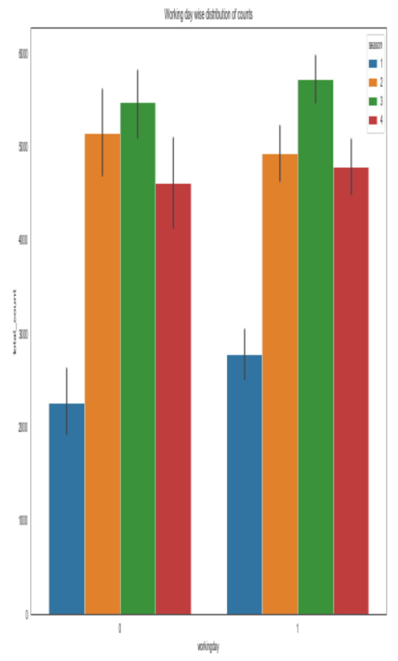
Weekday wise monthly distribution of counts.



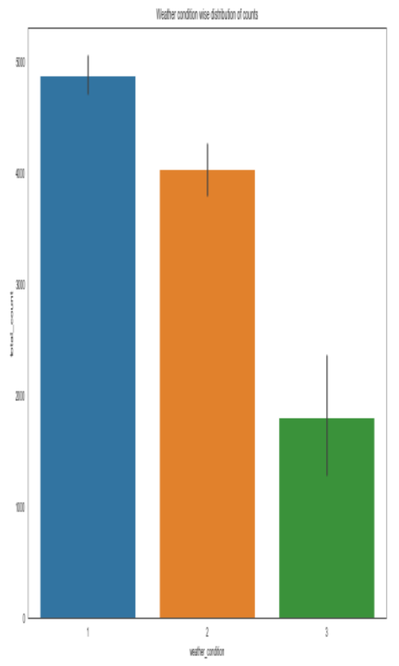
Yearly distribution of counts.



Holiday wise distribution of counts.



Working day wise distribution of counts.

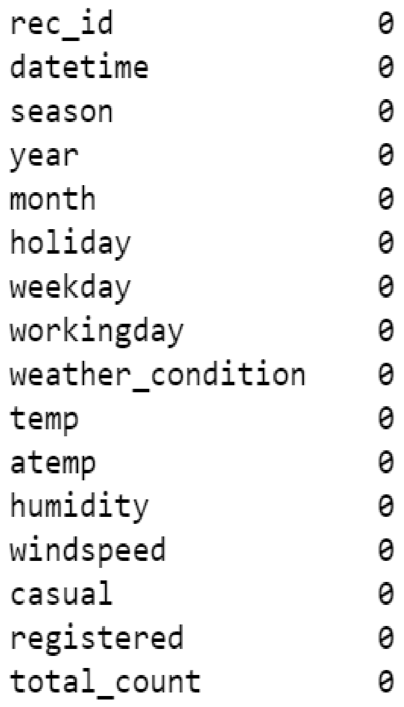


Weather condition wise distribution of counts.

2.3 <u>Missing values treatment:</u>

In statistics, *missing data*, or *missing values*, occurs when there is no *data value* stored for the variable in an observation. *Missing data* are a common occurrence and can have a significant effect on the conclusions that can be drawn from the *data*. If a column has more than 30% of data as missing value then either we can ignore the entire column or we ignore those observations.

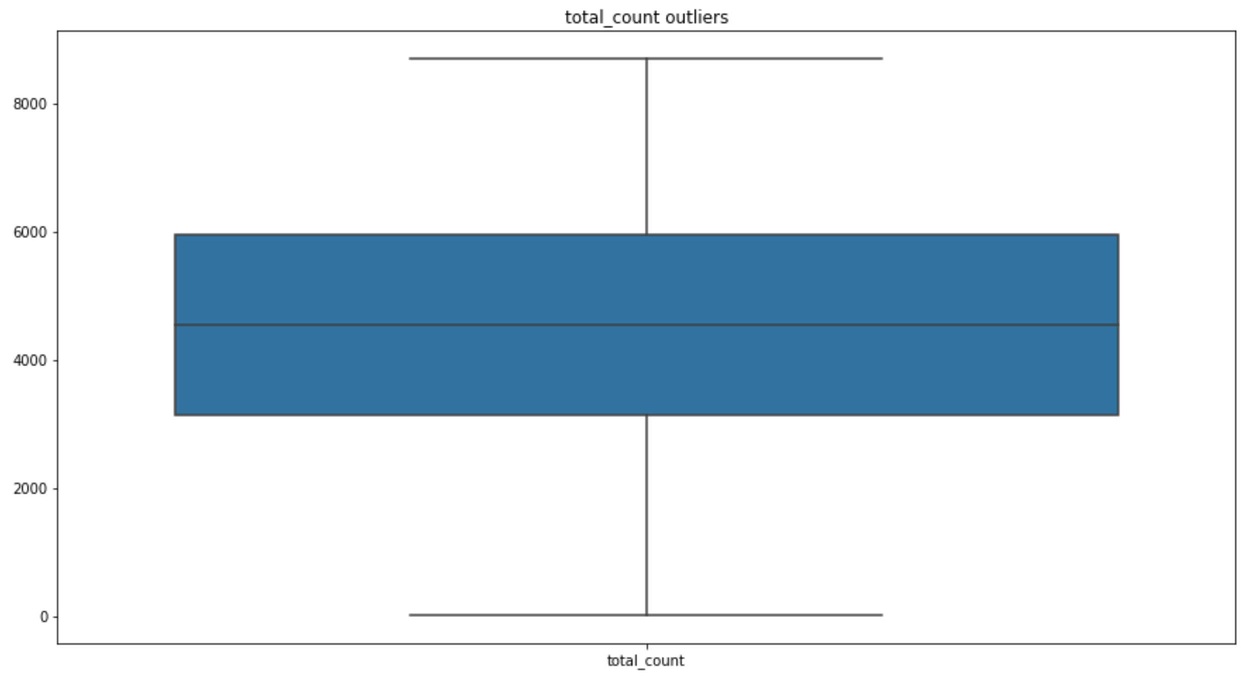
Missing values are a common occurrence, and you need to have a strategy for treating them. A missing value can signify a number of different things in your data. Perhaps the data was not available or not applicable or the event did not happen. It could be that the person who entered the data did not know the right value, or missed filling in. Typically, ignore the missing values, or exclude any records containing missing values, or replace missing values with the mean, or infer missing values from existing values. <u>We checked for missing values in our data and came to know that there are no missing values</u>

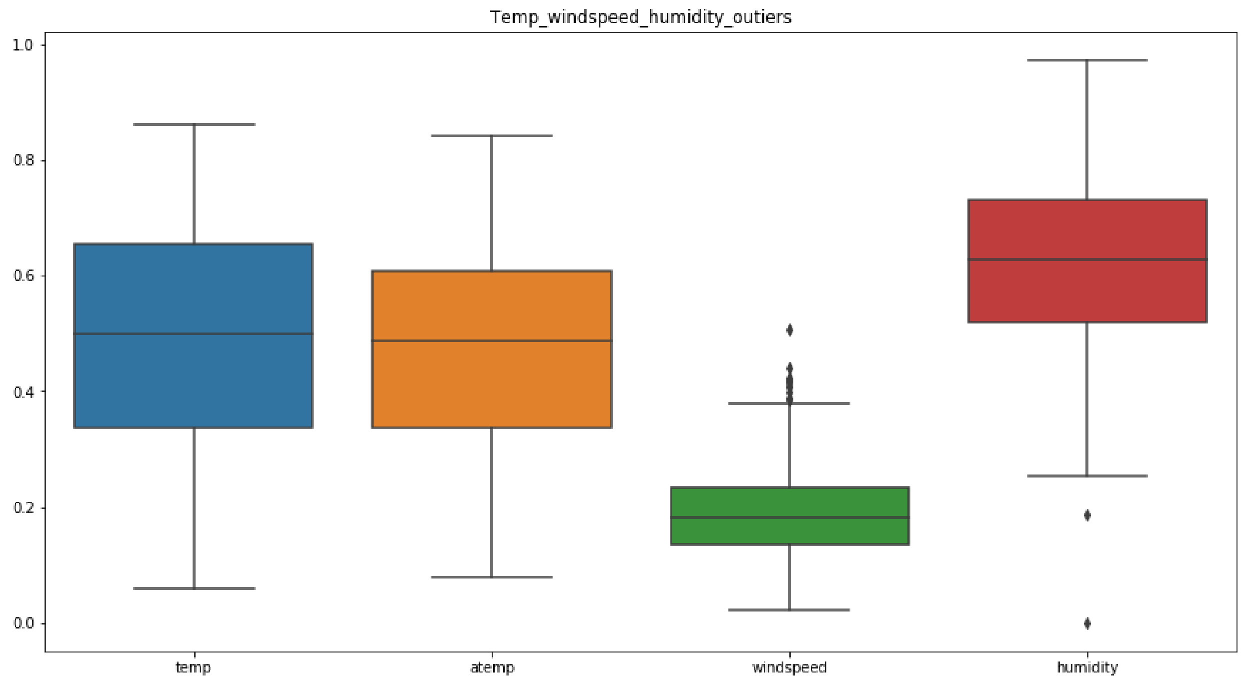


```
rec_id               0
datetime             0
season               0
year                 0
month                0
holiday              0
weekday              0
workingday           0
weather_condition    0
temp                 0
atemp                0
humidity             0
windspeed            0
casual               0
registered           0
total_count          0
```

2.4 <u>Outlier treatment:</u> An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. Outliers can drastically change the results of the data analysis and statistical modelling. Outliers are the unwanted abnormal values that may get generated due to rough handling of data or few values emerging as out of the range value in which most of the data lies.

The box plot for our data could be seen as follows:





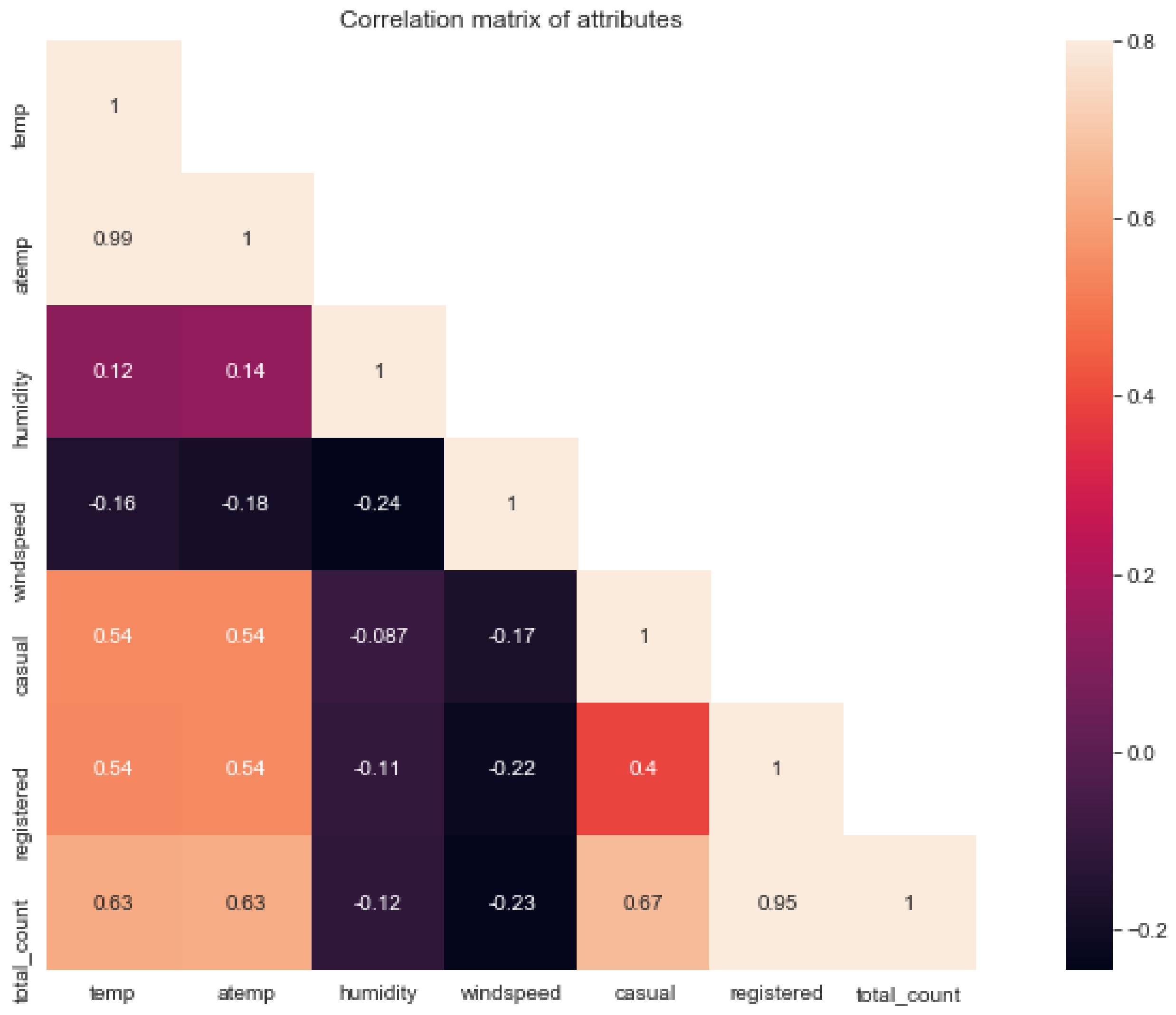
Observation:- From the box plot, we can observe that no outliers are present in temp & normalized temp but few outliers are present in windspeed and humidity variable.

In order to remove the outliers, we can either make these outliers as NA and can impute as missing value using methods as KNN or median or mean or mode or we can delete the entire row which contains outliers.

2.5 <u>Feature Selection</u>: Variable selection is an important aspect of model building. It helps in building predictive models free from correlated variables, biases and unwanted noise. It helps in selecting a subset of relevant features (variables, predictors) for use in model construction and subset of a learning algorithm's input variables upon which it should focus attention, while ignoring the rest.

Correlation Analysis

A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors. Here each numerical variable's correlation is mapped with each other's in a matrix which has been plotted in the following heat map.

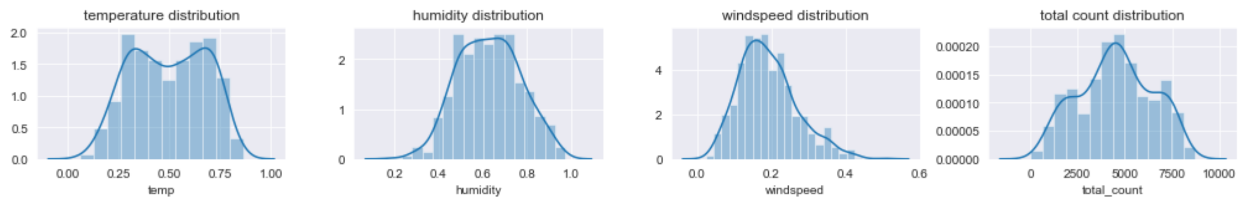


Observation:- From correlation plot, we can observe that some features are positively correlated or some are negatively correlated to each other.

The temp and atemp are highly positively correlated to each other, it means that both are carrying same information. The total_count, casual and registered are highly positively correlated to each other.

So, we are going to ignore atemp,casual and registered variable for further analysis.

2.6 <u>Feature Scaling</u>: Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data pre-processing step.



Observation: - Till here, our features are selected and they are already normalized.
So, we don't need to perform feature scaling.

2.7 Modelling

2.7.1 **<u>Model Selection:</u>** For modelling, we are going to use some famous models to our data-set and will conclude the result according to it.

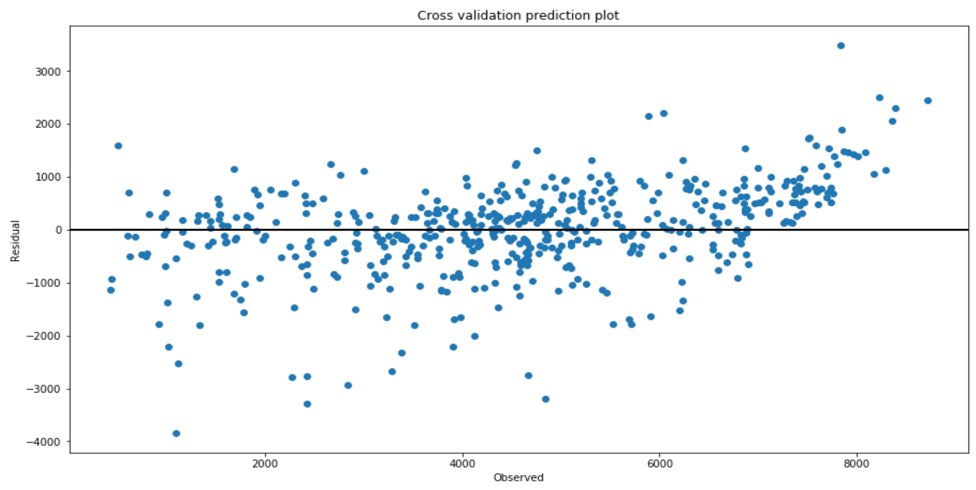
a)- **<u>Linear Regression:</u>** Linear regression is the most basic type of regression and commonly used predictive analysis. Linear regression is an approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.

```
#Training model:-
lr_model=linear_model.LinearRegression()
#fit the trained model:-
lr_model.fit(X_train,y_train)

#Accuracy of the model:-
lr=lr_model.score(X_train,y_train)
print('Accuracy of the model :',lr)
```

On evaluating the model, we received accuracy of 81.6 %.

a.1)- Cross Validation Prediction For Linear Regression: - Cross-validation is a procedure for estimating the test performance of a *method* of producing a model, rather than of the model itself using k-fold cross-validation .

```
predict=cross_val_predict(lr_model,X_train,y_train,cv=3)
```



Cross validation Plot

Observation:- Cross validation prediction plot tells us about finite variance between actual & target values. In above plot., some data points have same finite variance between them and some are not having it.

a.2)- **R squared score**:-

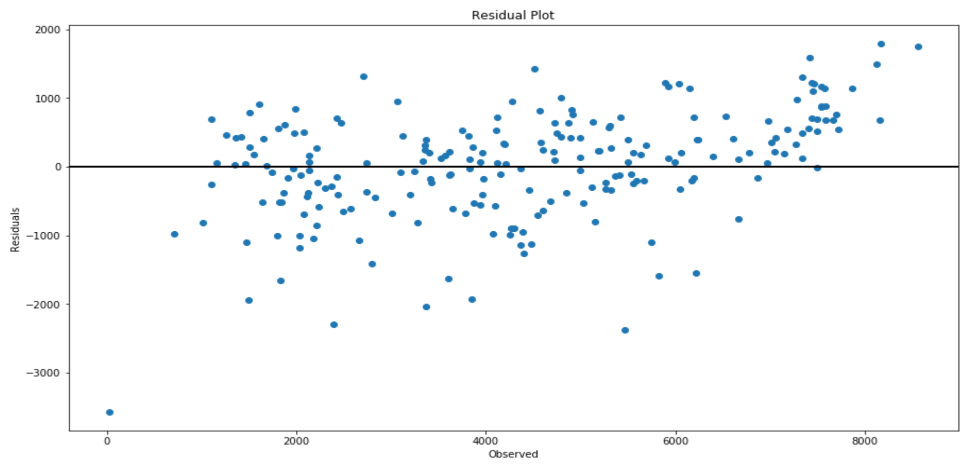
```
r2_scores=cross_val_score(lr_model,X_train,y_train,cv=3)

print('R squared score:',np.average(r2_scores))
```

Observation:- The R-squared or coefficient of determination is 0.80 for 3 fold cross validation,it means predictor is only able to predict 80% of the variance in the target varibale which is contributed by independednt variables.

a.3)- Model performance on test data set:-

```
Predicting the model:-
lr_pred=lr_model.predict(X_test)
lr_pred
```

a.4)- Residual plot-

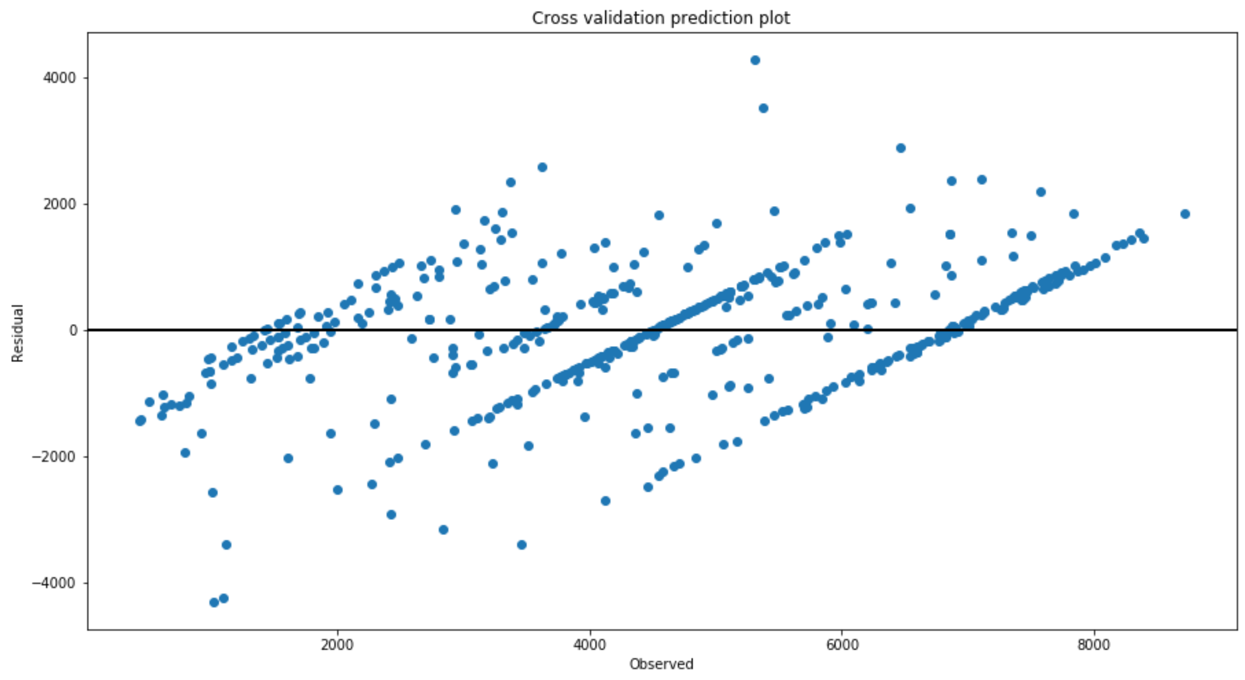


Observation:- Residual plot tells about finite variance between actual target value and predicted target value.In this plot,very less data points are have same finite variance between them and for most are not have it.

b)- **Decision tree regressor:-** Decision tree is a rule. Each branch connects nodes with “and” and multiple branches are connected by “or”. It can be used for classification and regression. It is a supervised machine learning algorithm. Accept continuous and categorical variables as independent variables. Extremely easy to understand by the business users.

Accuracy of model: 0.8082151777539757

b.1)- Cross Validation Prediction For Linear Regression: -



b.2)- R squared score:-

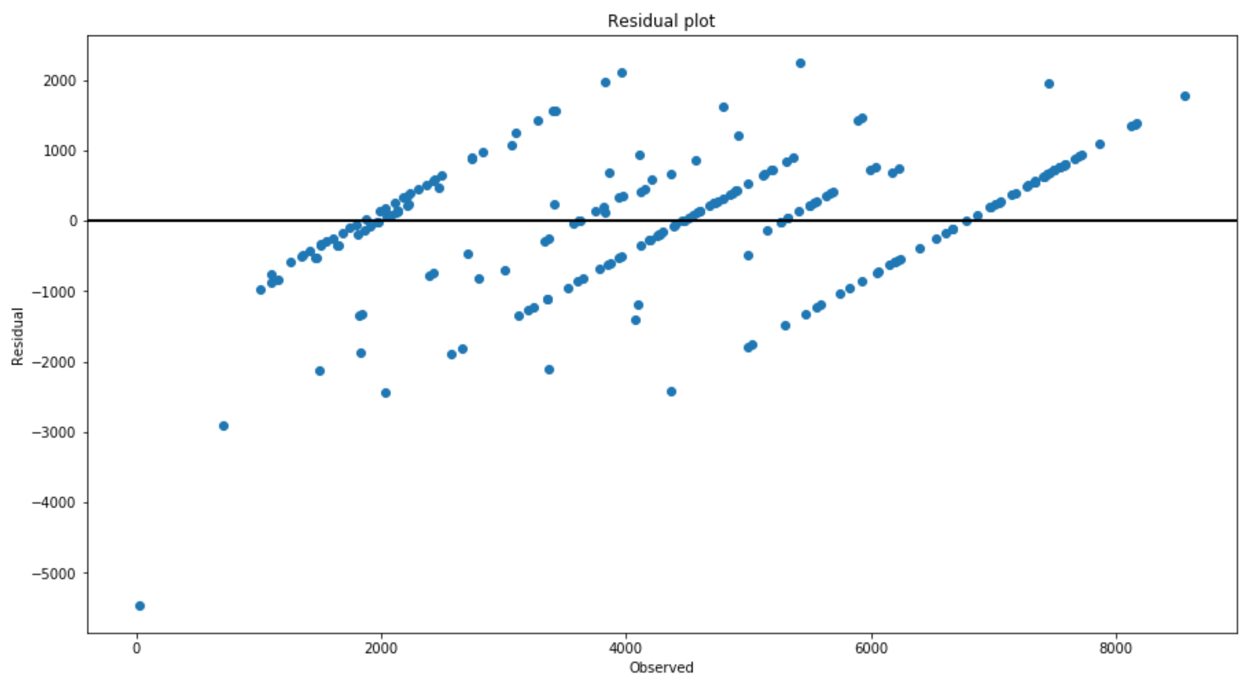
```
#R-squared scores:-
r2_scores=cross_val_score(dtr,X_train,y_train,cv=3)
print('R squared scores :',np.average(r2_scores))
```

Observation:- The R sqaured or coefficent of determination is 0.72 on avergae 3-fold cross validation, it means predictor is only able to predict 72% of the variance in the target variable which is contributed by independednt variable.

b.3)- Model performance on test data set:-

```
#Model performance on test dataset:-
dtr_pred=dtr.predict(X_test)
dtr_pred
```

b.4)- Residual plot-



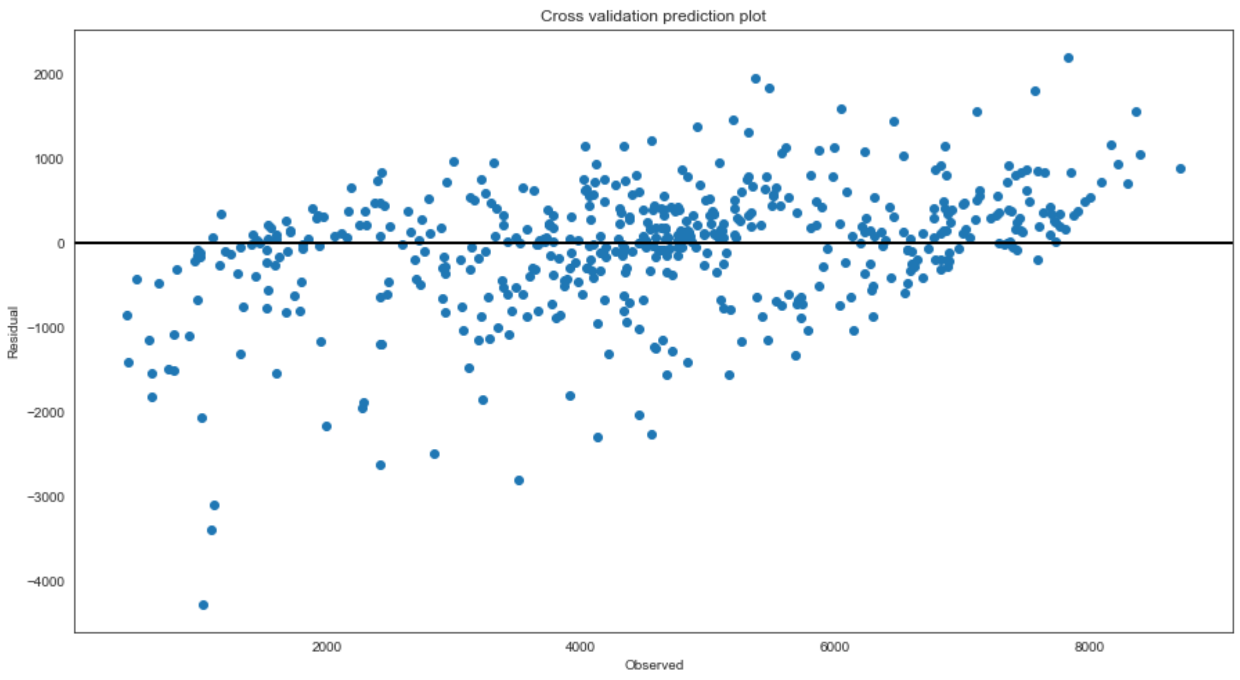
Observation:- Residual plot tells us about finite variance between actual target value and predicted target value. In this plot, some data points are having same finite variance between them and some are not having it.

3) **Random Forest:**Random Forest or decision tree forests are an ensemble learning method for classification, regression and other tasks. It consists of an arbitrary number of simple trees, which are used to determine the final outcome. In the regression problem, their responses are averaged to obtain an estimate of the dependent variable. Using tree ensembles can lead to significant improvement in prediction accuracy (i.e., better ability to predict new data cases). The goal of using a large number of trees is to train enough that each feature has a chance to appear in several model.

As we increase the number of trees the error count decrease until a point and then becomes constant.

Accuracy we get by Random Forest is 0.9806.

c.1)- Cross Validation Prediction For Linear Regression: -



c.2)- R squared score:-

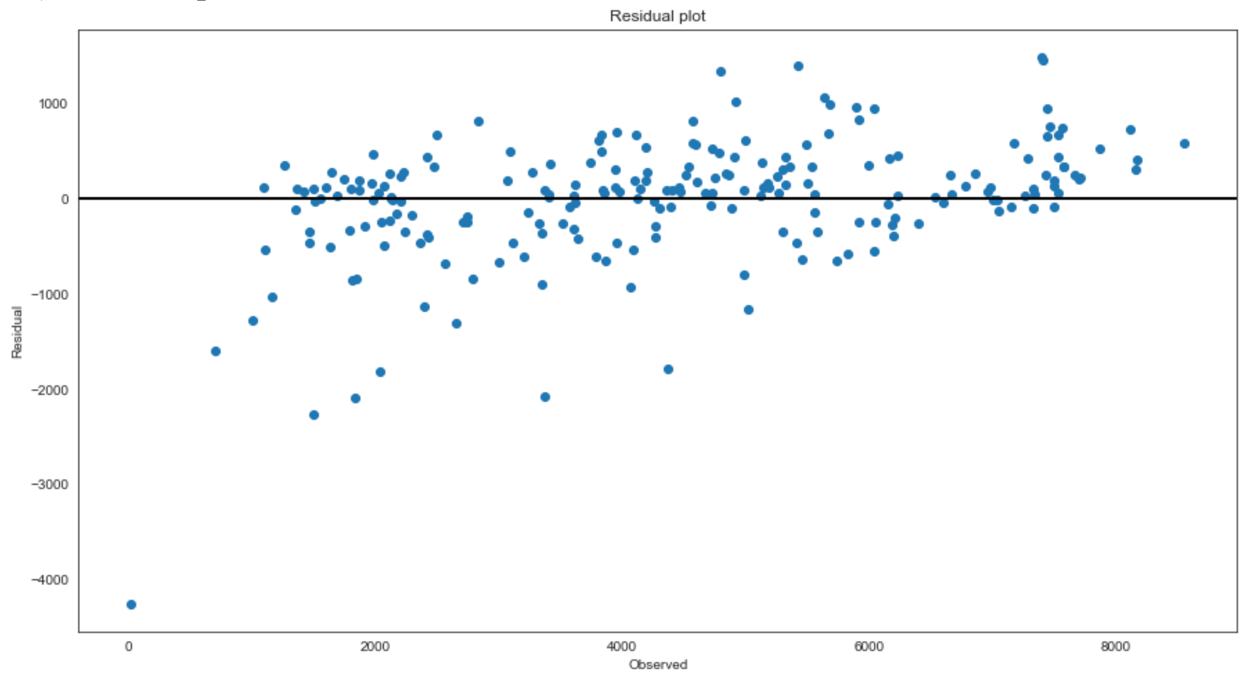
```
R-squared scores:-
r2_scores=cross_val_score(rf,X_train,y_train,cv=3)
print('R-squared scores :',np.average(r2_scores))
```

Observation:- The R-squared or coefficient of determination is 0.85 on avaerage 3 fold cross validation, it means that predictor is able to predict 85% of variance in the target varaible which is contributed by independednt variables.

c.3)- Model performance on test data set:-

```
#Model performance on test dataset:-
X_test=test_encoded_attributes
rf_pred=rf.predict(X_test)
rf_pred
```

c.4)- Residual plot-



Observation:- Residual plot tells about finite variance between actual target value and predicted target value.In this plot,very less data points are have same finite variance between them and for most are not have it.

3. Conclusion

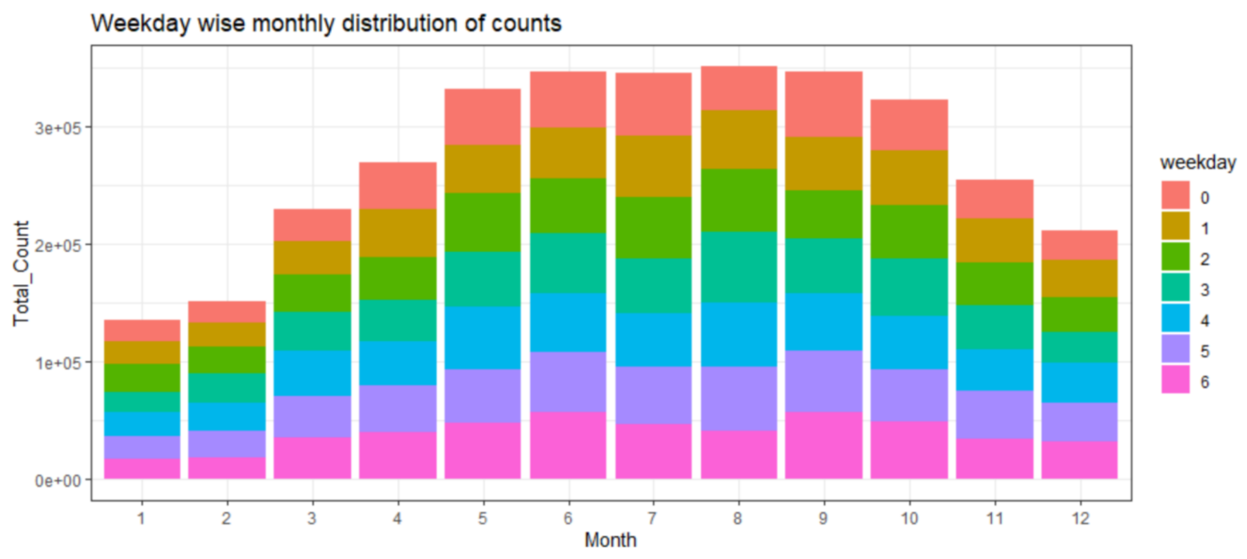
3.1 **Model Evaluation:** Model evaluation is done on basis of evaluation metrics or error metrics. Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results. Simply, building a predictive model is not our motive. But, creating and selecting a model which gives high accuracy on out of sample data. Hence, it is crucial to check accuracy or other metric of the model prior to computing predicted values. In our data as we applied regression models we have error metrics like Root mean square error (RMSE) & Mean absolute error (MAE).

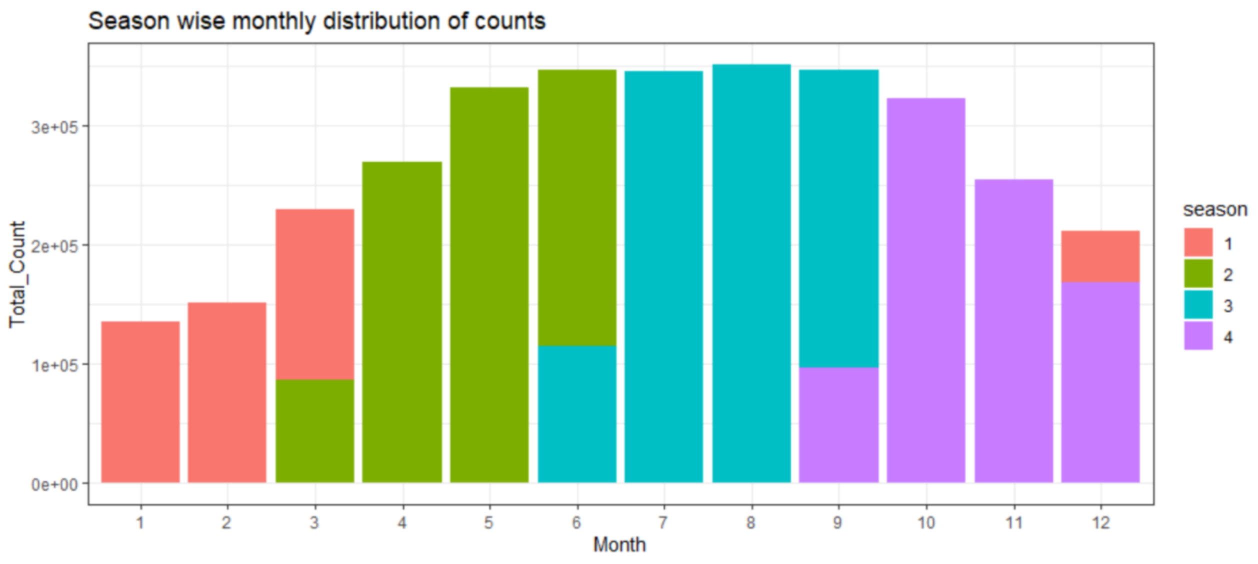
Model	**RMSE (Python)**	**RMSE (R)**	**MAE (Python)**	**MAE (R)**
Linear Regression	783.061	814.9782	594.736	603.6494
Decision Tree Regressor	925.797	952.7179	667.816	694.0827
Random Forest	642.552	683.9306	424.243	487.8721

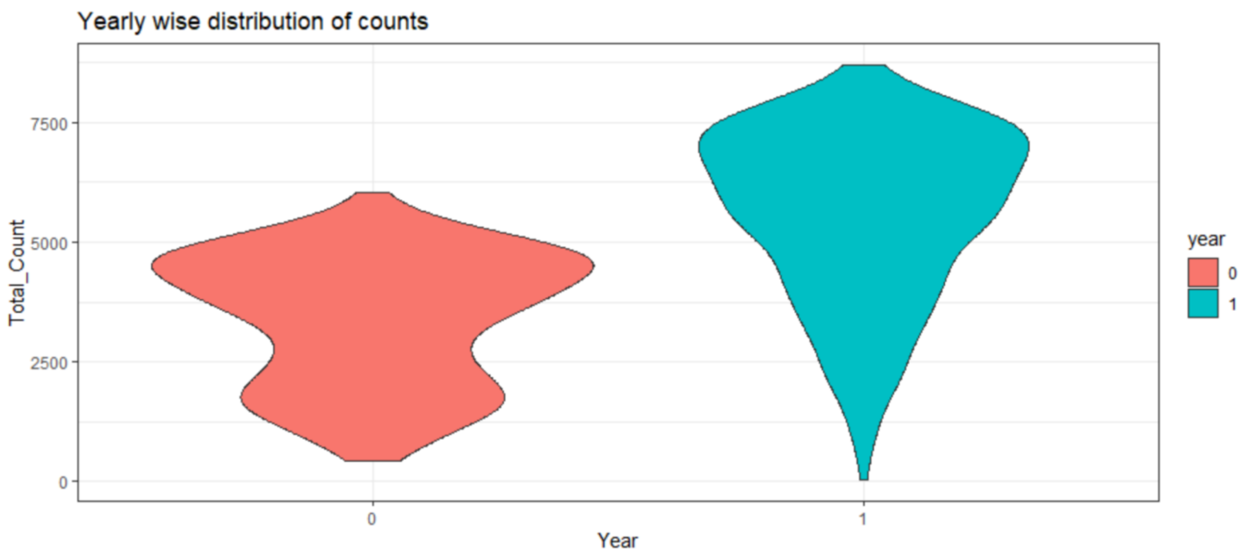
When we compare the root mean squared errors and mean absolute errors of all three models, the random forest model has less RMSE and MAE errors. So, finally random forest model is best for predicting the bike rental count.

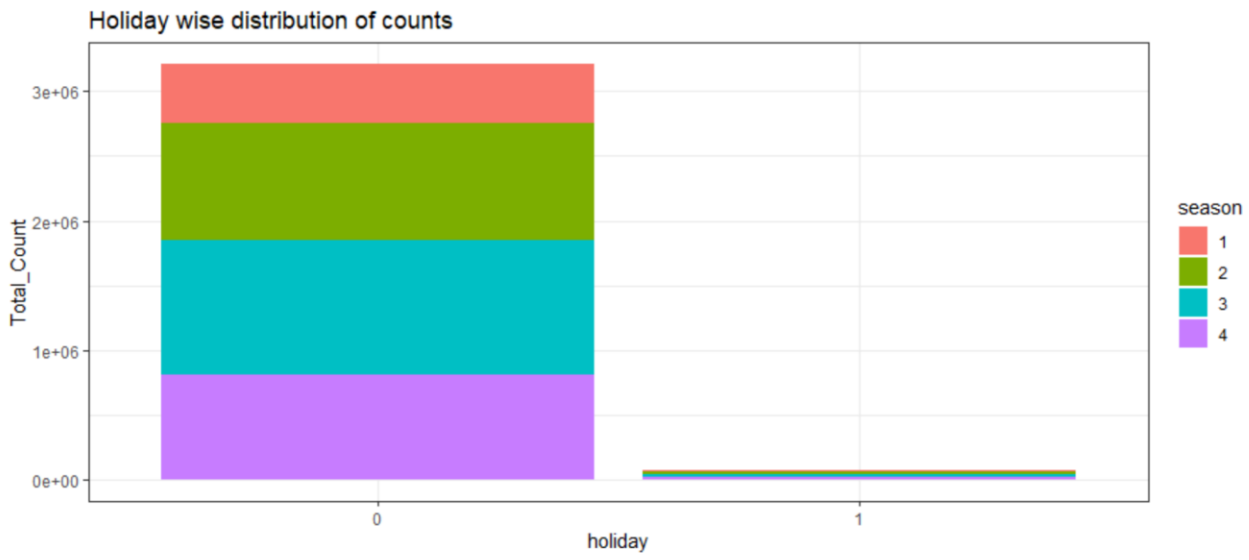
3.2. Model Selection: We can see that all models perform comparatively on average and therefore we select Random forest classifier models for better prediction.

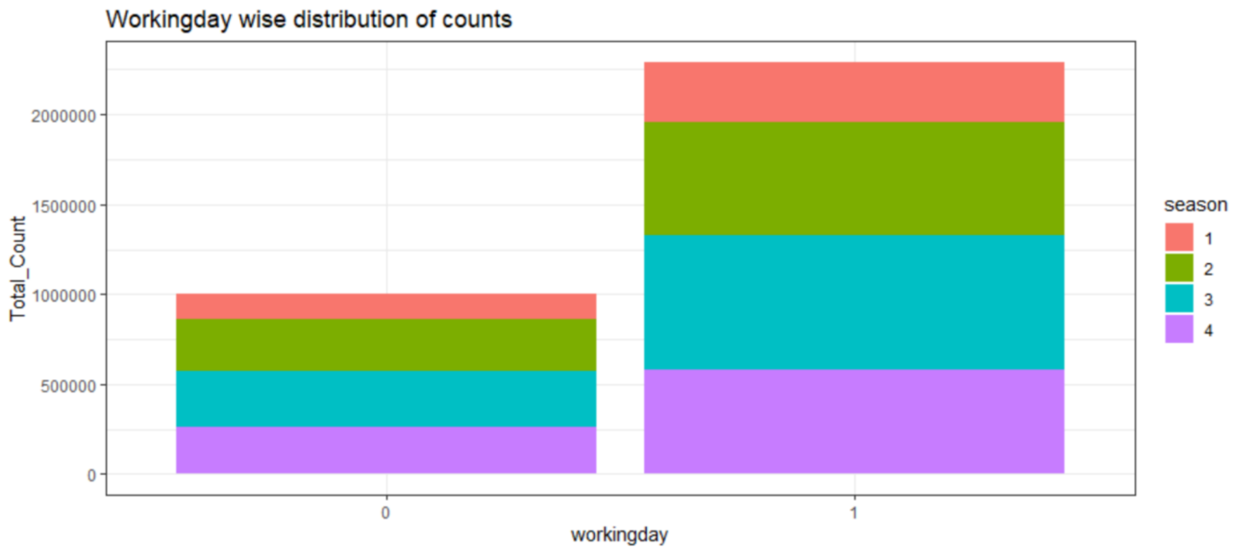
Appendix A -Imp Plots

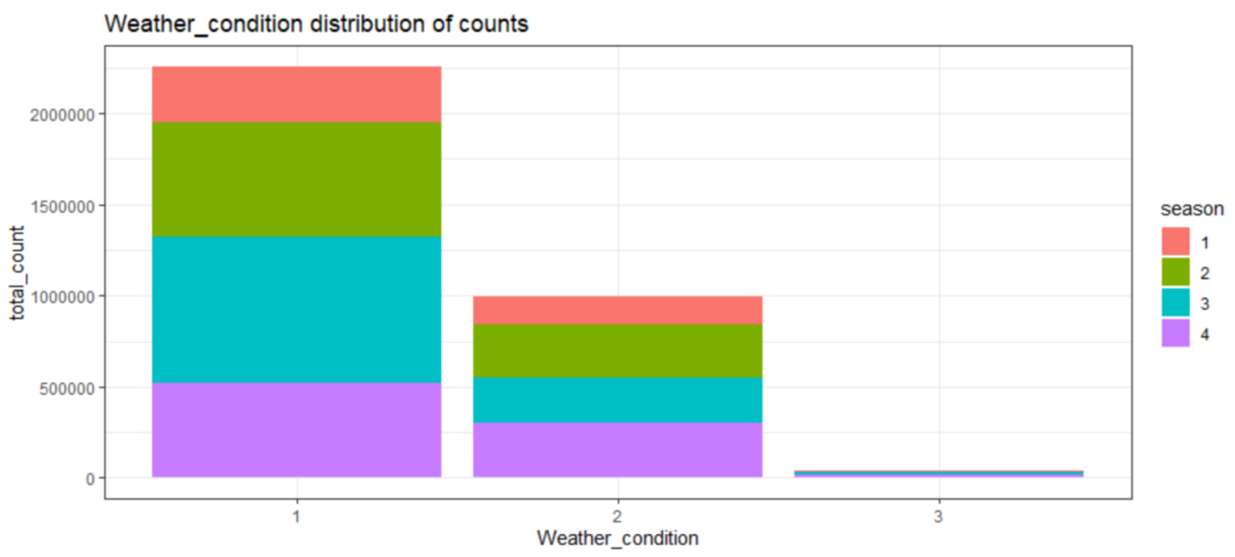


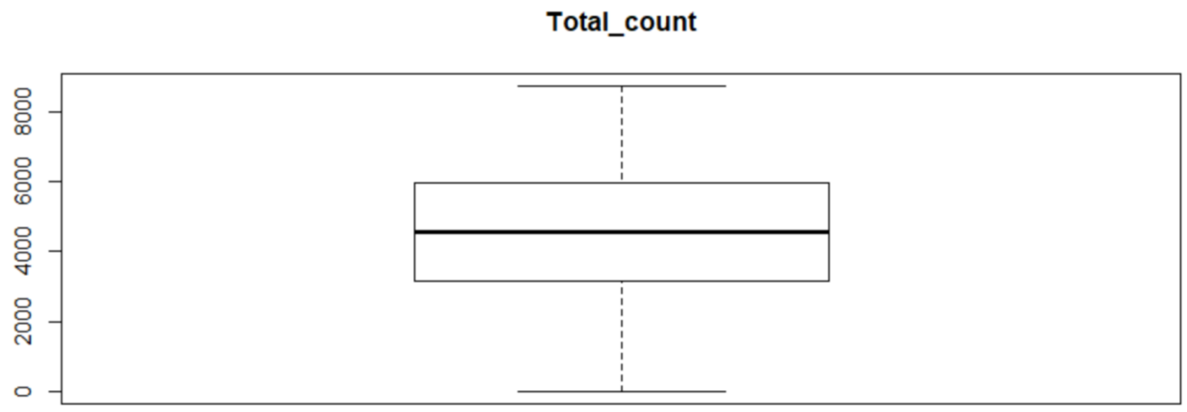


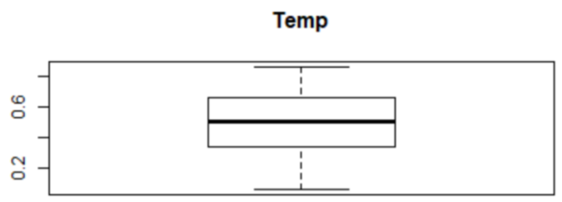

Yearly wise distribution of counts
Total_Count
7500
5000
2500
0
0
1
Year
year
0
1

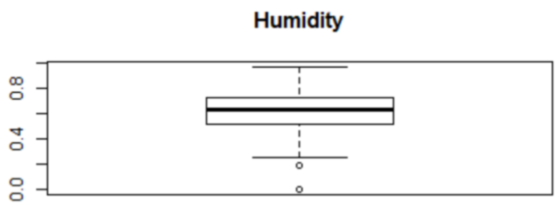

Holiday wise distribution of counts
Total_Count
3e+06
2e+06
1e+06
0e+00
0
1
holiday
season
1
2
3
4


Workingday wise distribution of counts
Total_Count
2000000
1500000
1000000
500000
0
0
1
workingday
season
1
2
3
4

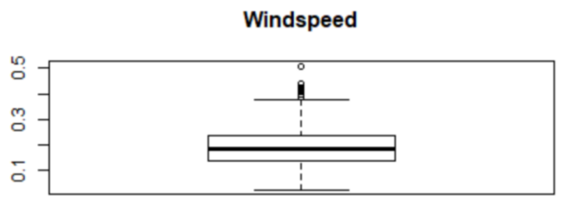

Weather_condition distribution of counts
total_count
2000000
1500000
1000000
500000
0
1
2
3
Weather_condition
season
1
2
3
4

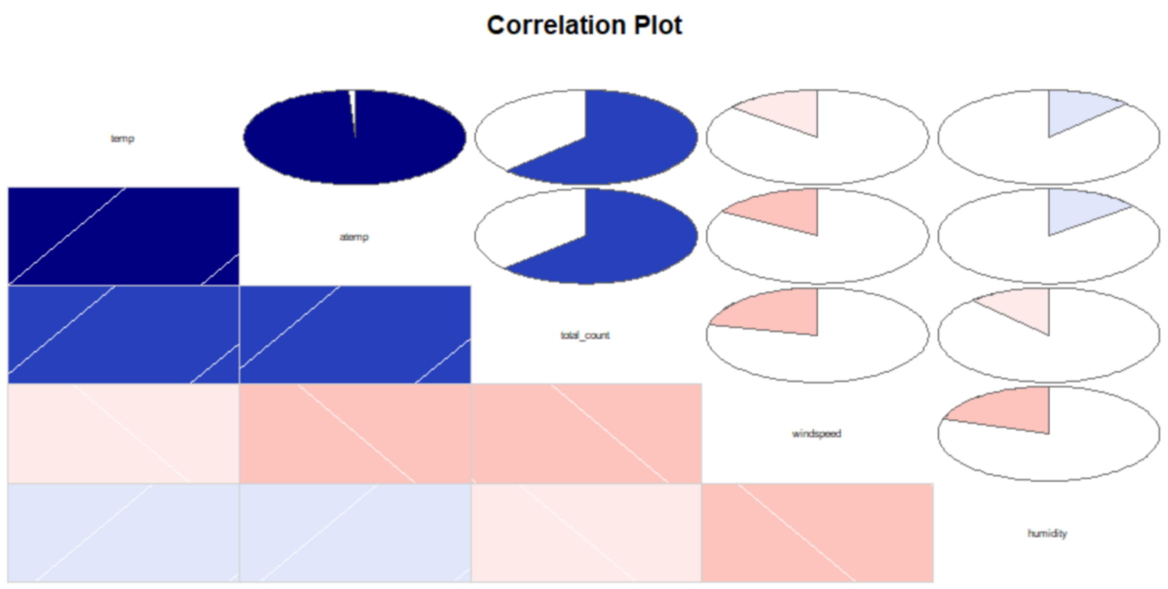

Total_count
0
2000
4000
6000
8000


Temp
0.2
0.6


Humidity
0.0
0.4
0.8

0.180917


Windspeed
0.1
0.3
0.5


Correlation Plot
temp
atemp
total_count
windspeed
humidity

References

- https://stackoverflow.com/
- https://medium.com/
- https://www.youtube.com/user/krishnaik06
- https://www.rdocumentation.org/
- https://www.analyticsvidhya.com/blog
- https://pydata.org/