Predicting daily bike rental counts based on the environmental and seasonal settings

Mohit H. Kothari

Data Science Project

5 August 2019

Contents

1. Introduction

2. Data Pre Processing and Visualizations

3. Model Building and Evaluation

Chapter 1: Introduction

1.1 Problem Statement

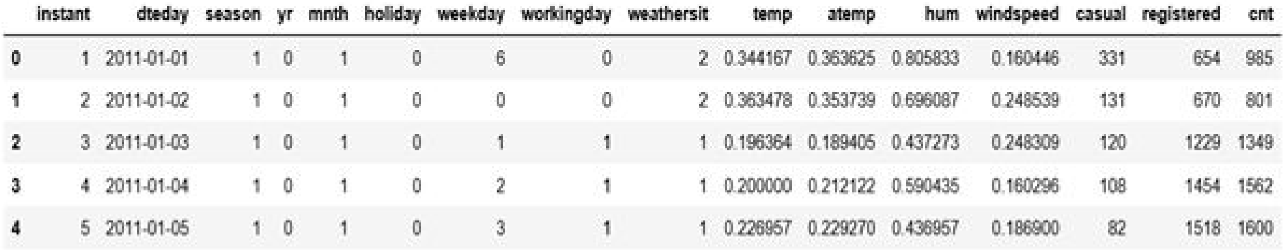
The objective of this project is to predict daily bike rentals count based on environment and seasonal settings.
The aim of the whole project is to predict the bike rental count for each day depending on several factors like season, temperature, weather, etc. parameters, so that company can be prepared for each day and can get an idea about peak demand days and low demand days.

1.2 Data

Data provided with the problem is day.csv.

This is a Regression problem where we have to predict a continous variable (count of bike rentals each day) depending on variables like season, weathersituation, year, month, workingday, weekday, holiday, temperature, feeled temperature, humidity, windspeed, count of casual users renting bike each day, count of registered users renting bike each day and final count of both users in count column.

Table 1.1 - Bike Count dataset of first 5 rows



	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

The details of the variables in the dataset are as follows:

1. instant: Record index
2. dteday: Date
3. season: Season (1:springer, 2:summer, 3:fall, 4:winter)
4. yr: Year (0: 2011, 1:2012)
5. mnth: Month (1 to 12) hr: Hour (0 to 23)
6. holiday: weather day is holiday or not (extracted fromHoliday Schedule)
7. weekday: Day of the week
8. workingday: If day is neither weekend nor holiday is 1, otherwise is 0.
9. weathersit: (extracted fromFreemeteo)
   1: Clear, Few clouds, Partly cloudy, 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
10. 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)
11. 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)
12. hum: Normalized humidity. The values are divided to 100 (max)
13. windspeed: Normalized wind speed. The values are divided to 67 (max)
14. casual: count of casual users
15. registered: count of registered users
16. cnt: count of total rental bikes including both casual and registered

Here we will predict ‘cnt’ variable using all independent variable; or we will predict both casual and registered user individually and then add it to get final count, depending on how data is. So our next step will be EDA (Exploratory data analysis) & visualizations.

Chapter 2: Data Pre Processing and Visualizations

2.1 Exploratory Data Analysis

Here we are given a dataset of 731 rows and 16 Variables. Here ‘dteday’ column was converted into ‘date’ column and then we dropped ‘instant’ and ‘dteday’ column which was of no use.
So finally we are left with 8 category variables and 7 continuous variables in which 3 are our target variables (casual, registered and cnt variables)
So our dataset is having 731 rows and 15 varaibles

2.2 Missing Value Analysis

Our next step will be to predict missing values in our dataset. We applied missing value analysis and found that NO variables are having any missing values and so no need for imputation or any other methods.

2.3 Outlier Analysis

So since there is no missing value, we will now go ahead with analyzing outliers present in our datasets.
We will use Boxplot method to get the visualization of outliers present in our continous variables and then we will treat outliers.

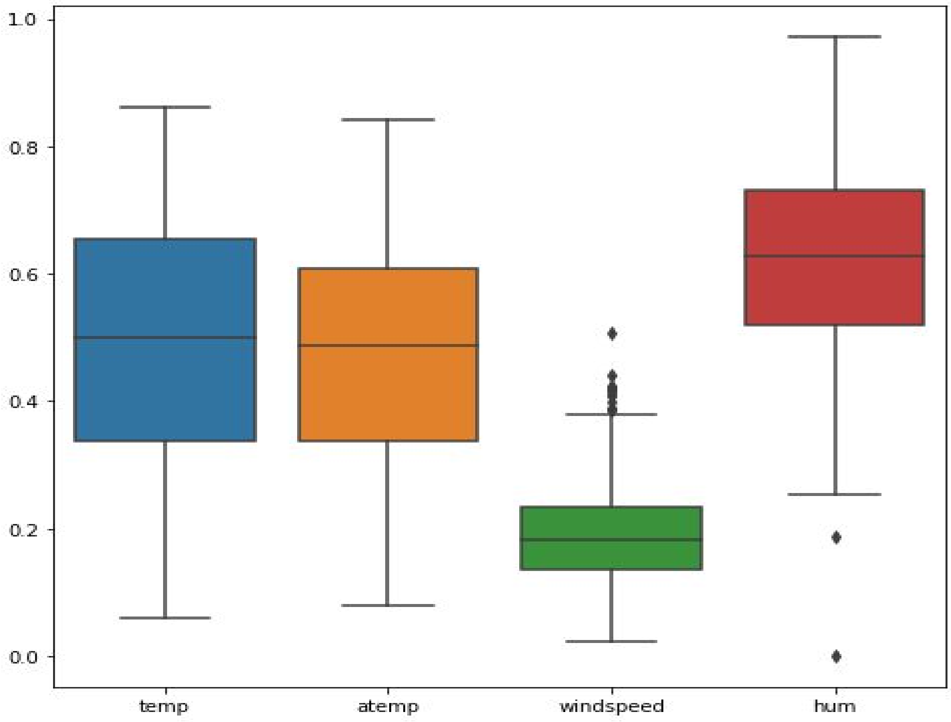


Figure 1: Boxplot of Independent Variables

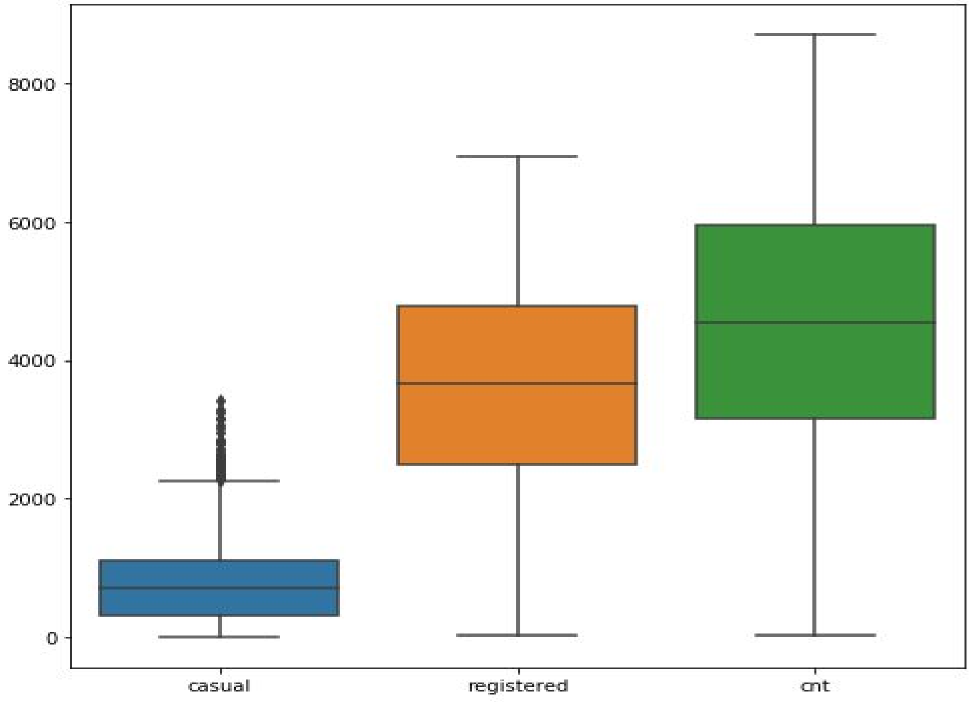


Figure 2 : Boxplot of Dependent Variables

Here there are outliers present in variables ‘humidity’ and ‘windspeed’ variables and also in ‘casual’ variable.

Since outliers are very less in numbers (2 and 13) and that might represent some severe weather conditions we won't delete this rows our impute any value at that place.

So for casual users we wont do anything because here also ther emight be some extreme user count because of soem occassion like world environment day or world health day.

2.4 Distribution of Continous Variables (Histogram)

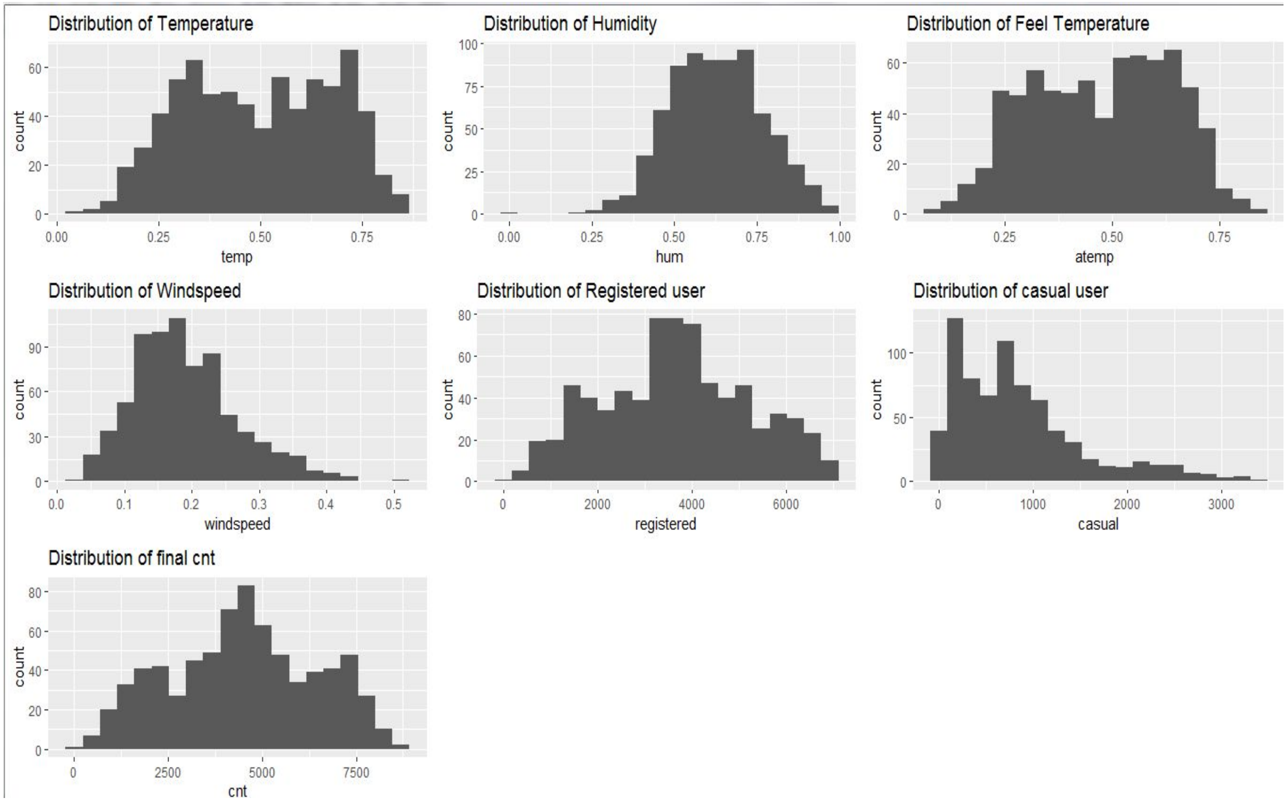


Figure 3 : Histograms of Continous Variables

The output variable is continous so to solve this kind of problem we will require Regression technique to get final count.
In order to apply regression technique our continous variables msut be normally distributed, so to check whether they are normally distributed or not we will plot histogram to know about their distribution.

1. ‘temp’ and ‘atemp’ are normally distributed
2. ‘Humidity’ is following normal distribution
3. ‘Windspeed’ is slightly skewed towards left
4. ‘Count’ variable is normally distributed which is our target variable

2.5 Boxplot to Analyze Category Variables

Now we will plot boxplot to analyze how our categorical variables are distributed with respect to our count variable.

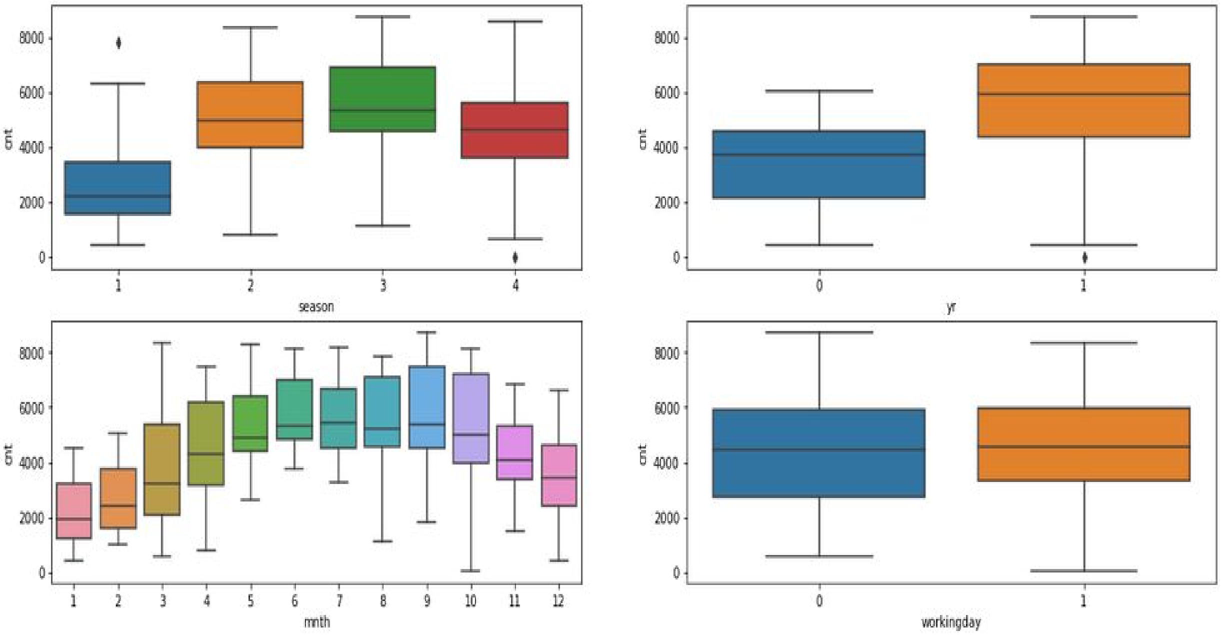


Figure 4: Boxplot of 1. Season, 2. Year, 3. Month and 4. The working day with a count

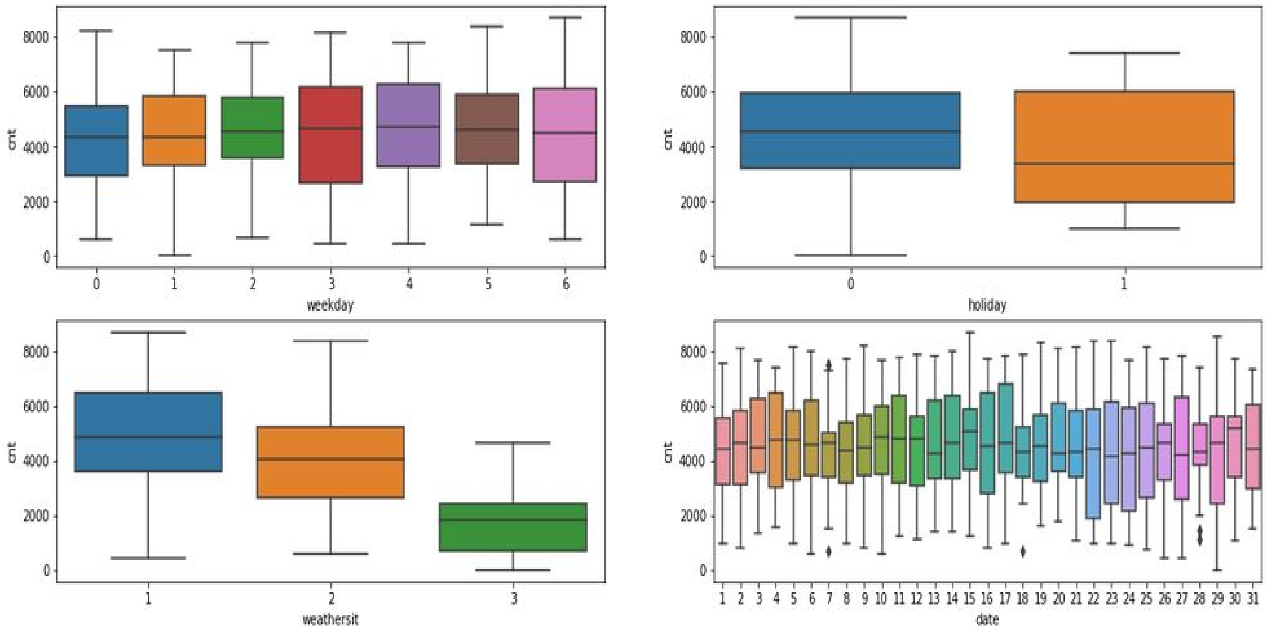


Figure 5: Boxplot of 1. weekday, 2. Holiday, 3. Weathersit and 4. Date with a count

Some of the analysis drawn from these plots are

1. Weather situation 1 has a higher count which reflects people like to ride a bicycle in clear weather
2. In fall and Summer season more people ride a bicycle and it drops in Spring season
3. From month 5 to month 10 there is high demand it fits well with season analysis
4. There no such trend observed in the day variable it is almost constant
5. From the year 2011 to 2012 the average bicycle count has almost doubled, the company is growing
6. We could drop the variable date from the dataset as they are not exhibiting any trend
7. On weekends/holiday the count is low because registered user doesn't go to the office and casual users are using it more as compared to the registered user

2.6 Correlation Analysis

Now we will try to see how the contonous variables are correlated with final cnt variable and also with respect to each other.

By knowing variable controbution to final variable we will how and by which amount it is useful in our dataset.

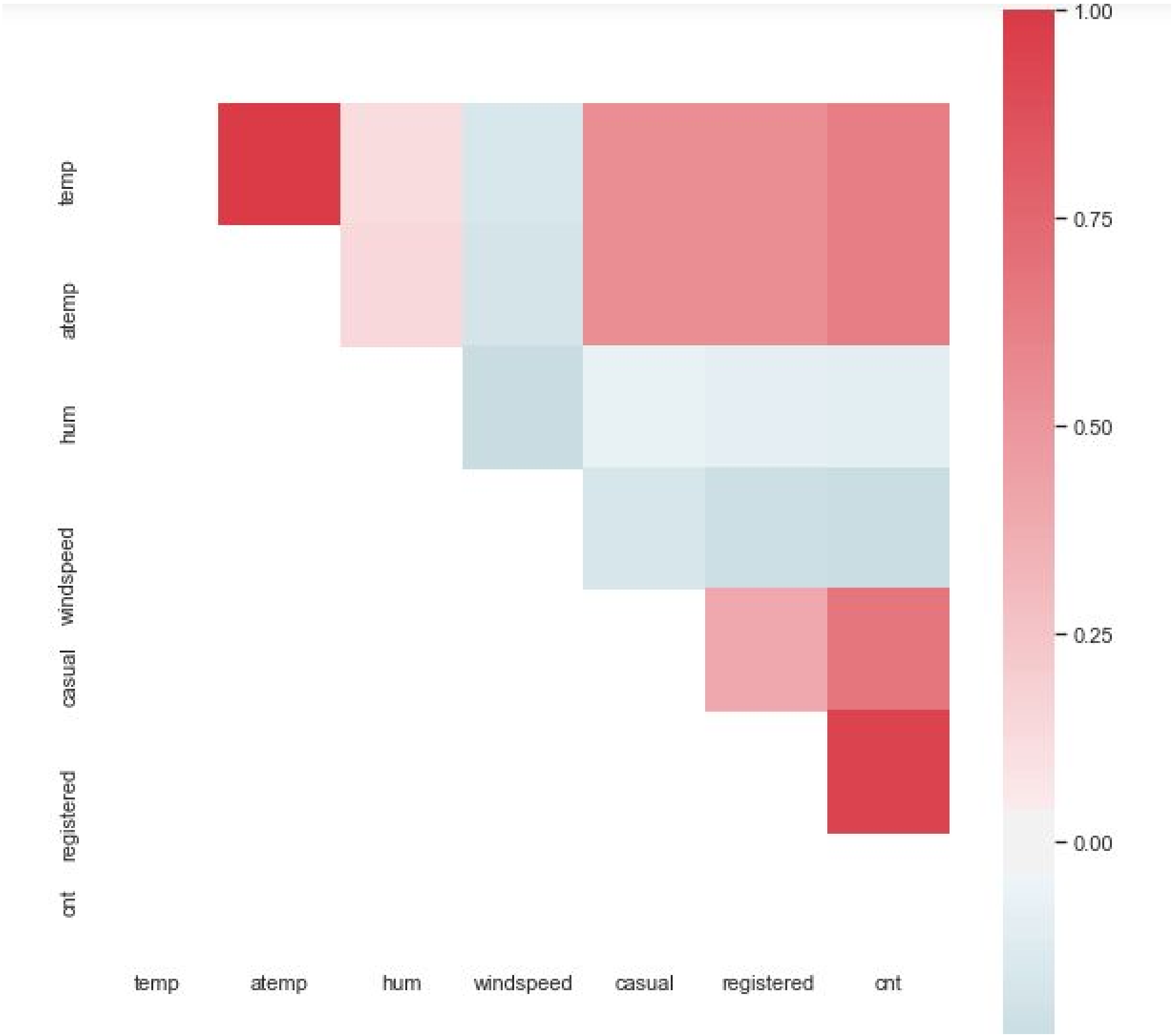


Figure 6 : Correlation Matrix of Continous variable

Here +1 Represnets that variable are highly positively correlated and -1 represents that variables are highly negatively correlated.

1. Here ‘temp’ and ‘atemp’ are highly correlated.
2. Also Registered users are highly correlated with cnt variables
3. Windspeed and humidity are not that much correlated

2.7 Feature Selection

1. We will remove "temp" variable as it is highly correlated with "atemp"
2. Becasue feeled temperature is more important as comapred to temperature
3. We will also remove "date" variable as it is not exhibiting any trend and is of no use
4. We will also remove Registered and casual user because we are interested in final count and registered user base will increase over time
5. So we will remove 1. Date 2. Temp 3. Casual and 4. Registered Column

Chapter 3: Model Building and Evaluation

3.1 Model Selection

Since our is a regression problem we will use regression algorithm to predict the final count. For regression we have algorithms like:

1. Decision tree
2. Random Forest Regressor
3. Linear Regression
4. Gradient Boosting

Out of this Decision tree is a algorithm of weak learners so it wont get us higher accuracy

Linear regression has its own limitations.

So we will use ensemble method of model building and predicting.

Ensemble learning, in general, is a model that makes predictions based on a number of different models. By combining individual models, the ensemble model tends to be more flexible (less bias) and less data-sensitive (less variance).

Two most popular ensemble methods are bagging and boosting.

1. Bagging: Training a bunch of individual models in a parallel way. Each model is trained by a random subset of the data
2. Boosting: Training a bunch of individual models in a sequential way. Each individual model learns from mistakes made by the previous model.

Random forest is an ensemble model using bagging as the ensemble method and decision tree as the individual model.

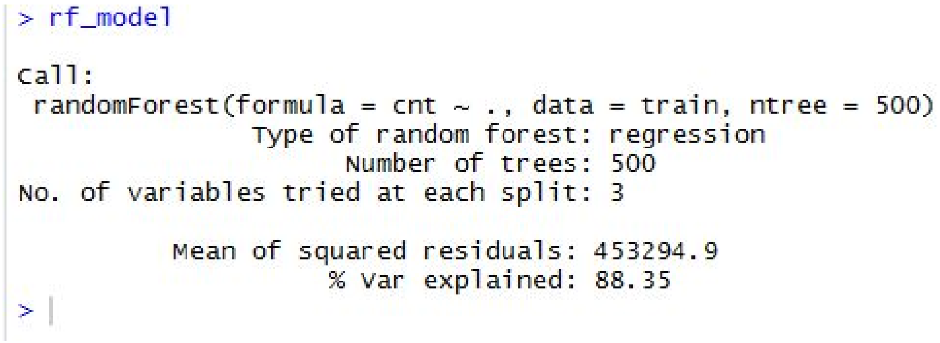
3.2 Random Forest Regressor

Here we will be using random Forest regressor to predict our count variable.

First we have split the data into 80% - 20% (In R) and 70% - 30% (in Python) as train and test data.

Train data will be used to train our model and then we will check model on test data removing target variable from test data and then checking the differences between actual predicted value and original count vlaue that existed in test data.

In Random forest we have used 500 trees for building the model



```
> rf_model

Call:
 randomForest(formula = cnt ~ ., data = train, ntree = 500) 
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 453294.9
                    % Var explained: 88.35
> 
```

Figure 7 : Random Forest Parameters

By changing some parameters we could do fine tuning of the model but the change in accuracy was not that much.

By increasing no. of trees from 500 to 10000 the change in accuracy was only 0.5% improvement.

3.3 Model Evaluation - MAPE

The relative error preference can also be expressed with Mean Absolute Percentage Error, MAPE.For each object, the absolute error is divided by the target value, giving relative error. MAPE can also be thought as weighted versions of MAE.

Whenever we talk in %Error rate or Accuracy then MAPE is best parameter to judge.

$$\mathrm{MAPE} = \frac{100\%}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

While running our model in:

1. Python with 70-30% data split we got **MAPE as 15%(approx.)** which means that our Model can Accurately predict for **85% of test cases.**
2. R with 80-20% data split our **MAPE comes out to be 14%(approx.)** which means our model can accurately predict for **86% of test cases**.

3.4 Future Scope

Here we have only used Bagging algorithm for predicting test cases but we can do the following steps to fine-tune model:

1. Use Boosting Algorithm like gradient boost, adaboost, etc.
2. Use Neural Networks model to accurately predict cases
3. Use some more feature engineering to make data pre-processing step finer

Appendix

List of Figures Used:

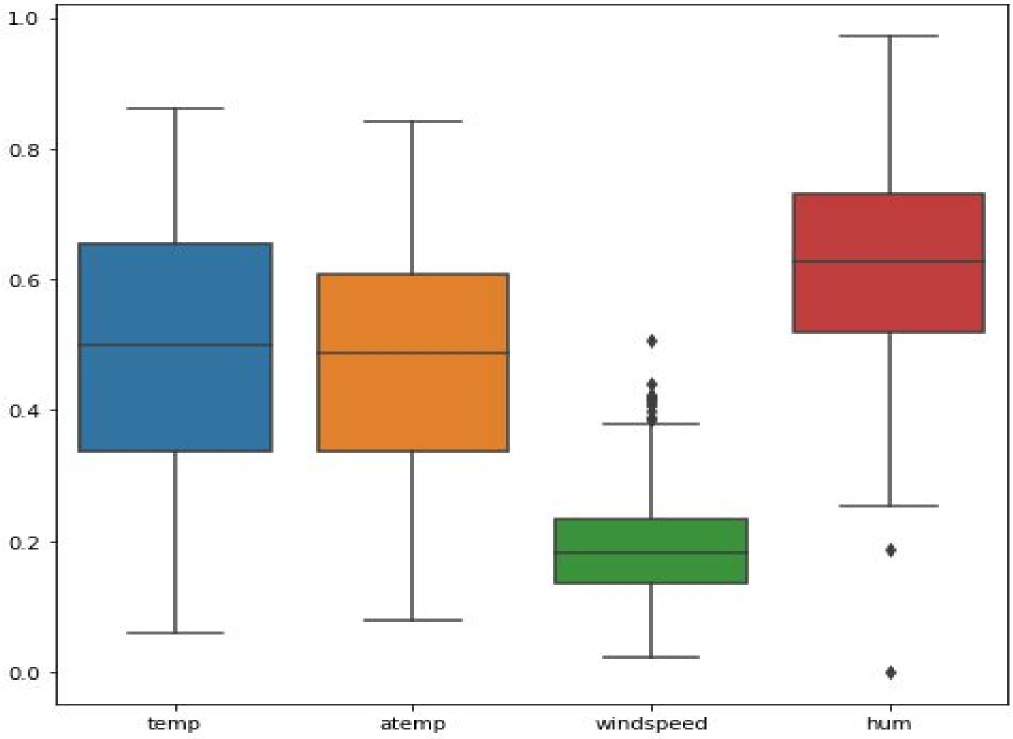


Figure 1: Boxplot of Independent Continous variables

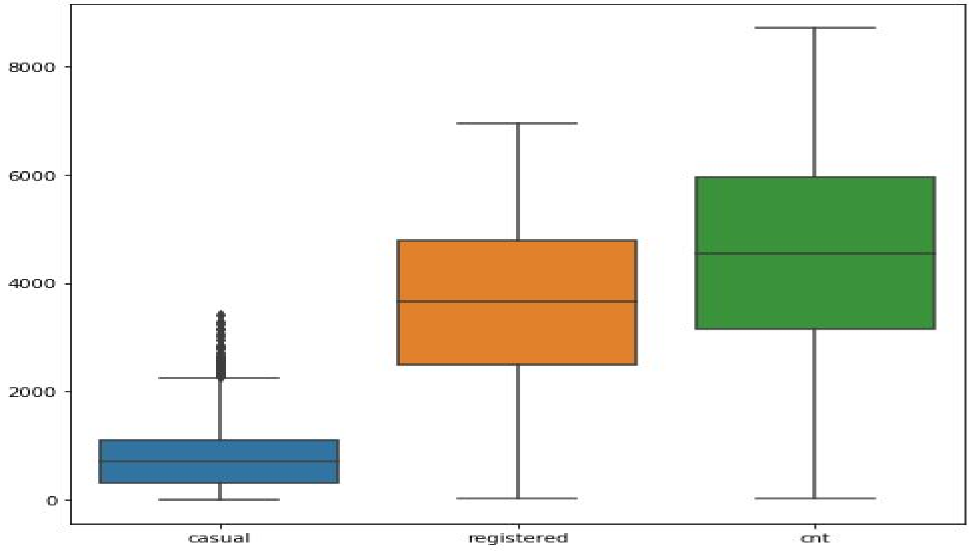


Figure 2: Boxplot of Dependent Continous Variable

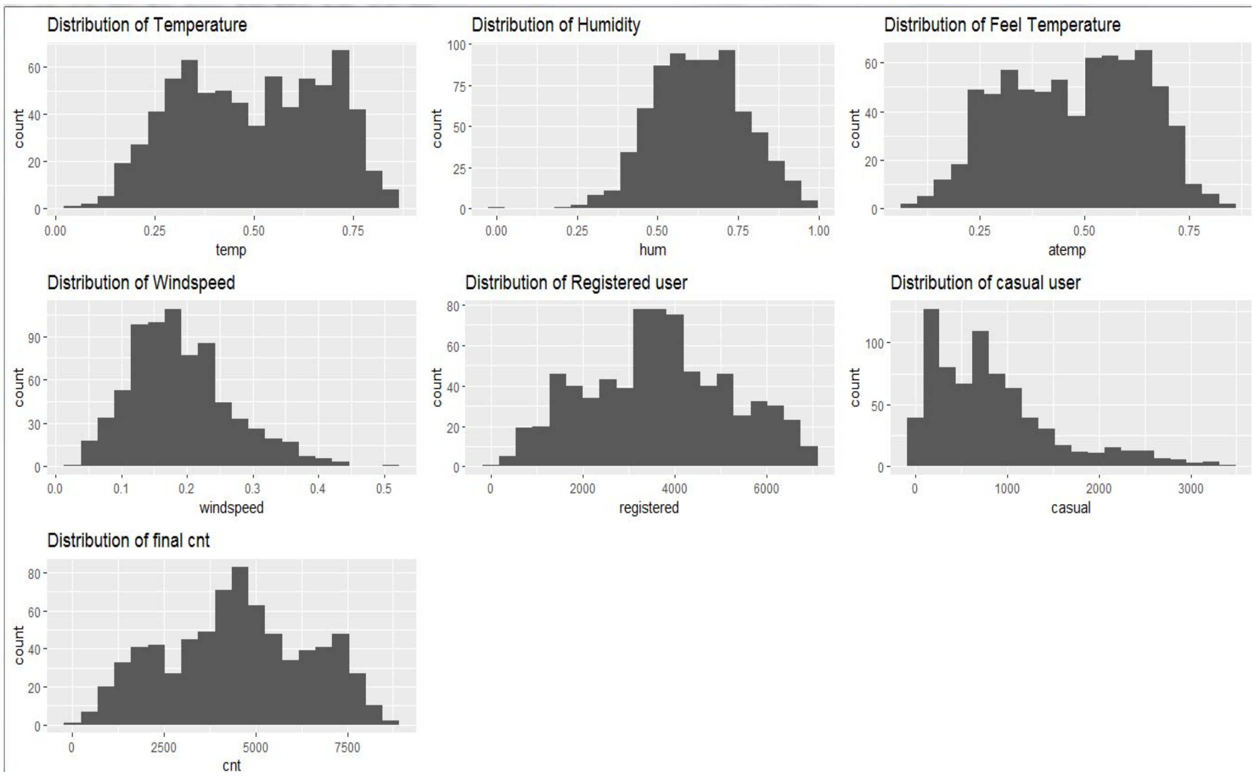


Figure 3 : Histograms of Continous Variables

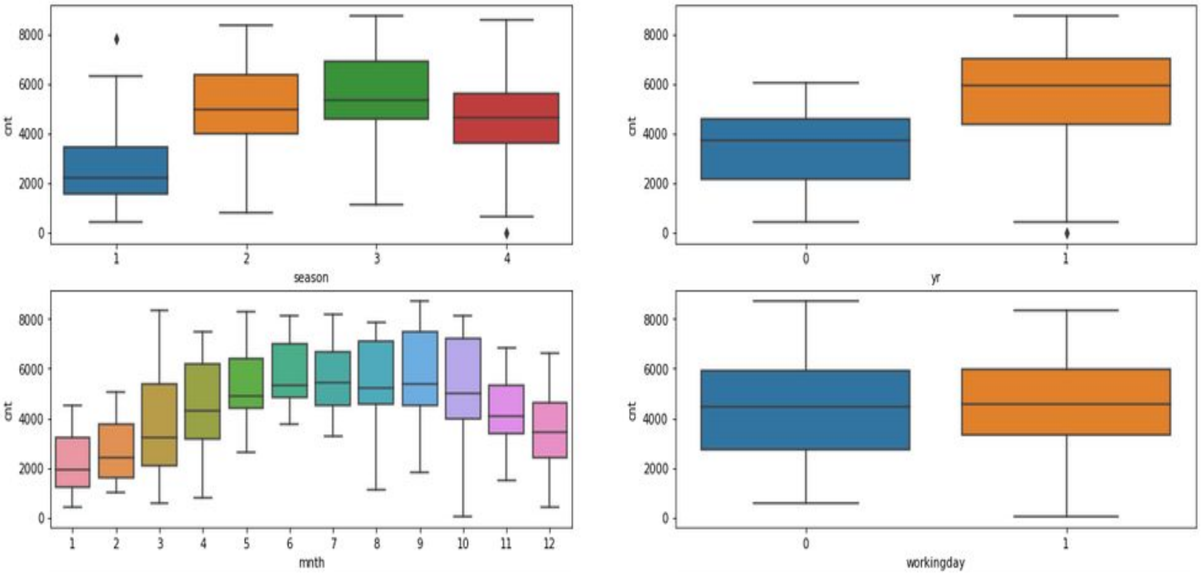


Figure 4: Boxplot of 1. Season, 2. Year, 3. Month and 4. The working day with a count

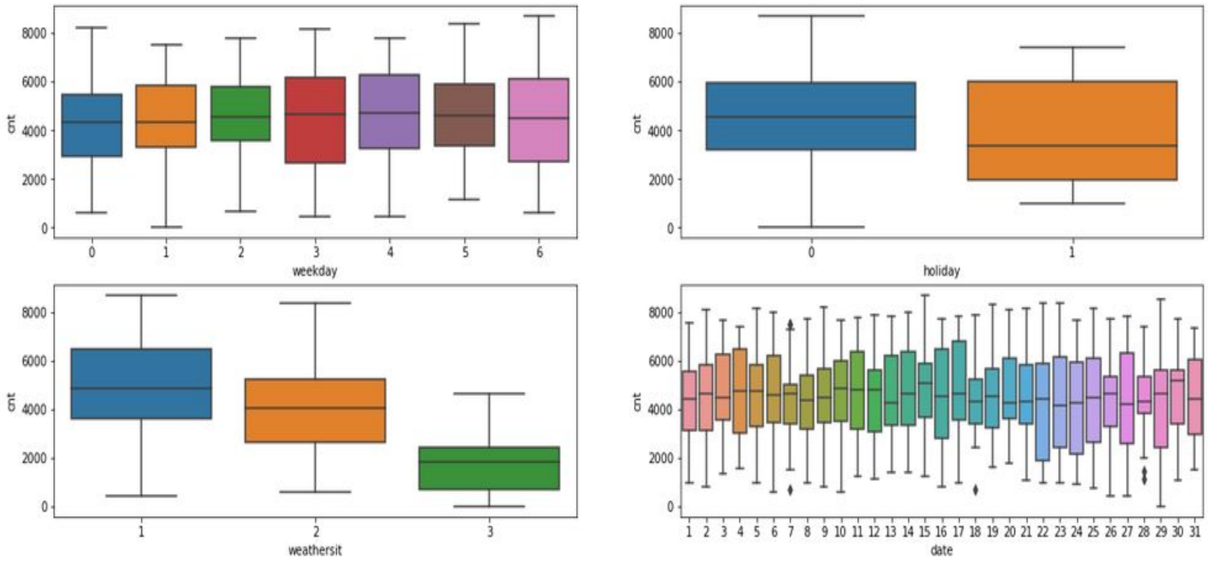


Figure 5: Boxplot of 1. Weekday, 2. Holiday, 3. Weathersit and 4. The date with a count

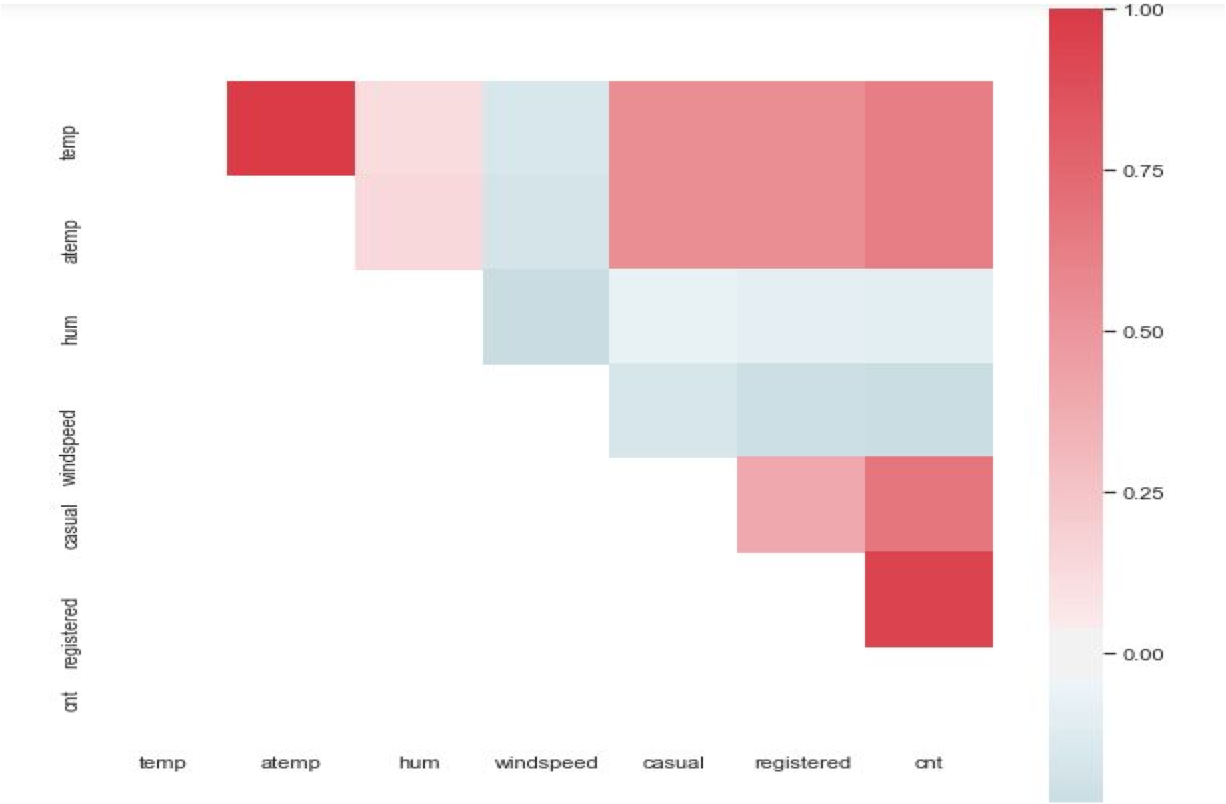
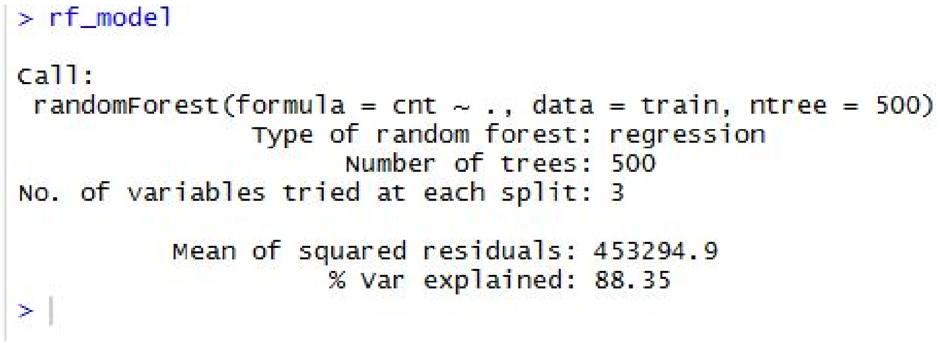


Figure 6 : Correlation Matrix of Continous variable



```
> rf_model

Call:
 randomForest(formula = cnt ~ ., data = train, ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 453294.9
                    % Var explained: 88.35
> |
```

Figure 7: Random Forest Parameters

THANK YOU

***