

Predicting Bike Rental Count

Pranav Tewari | 26 January 2019

Chapter 1

Introduction

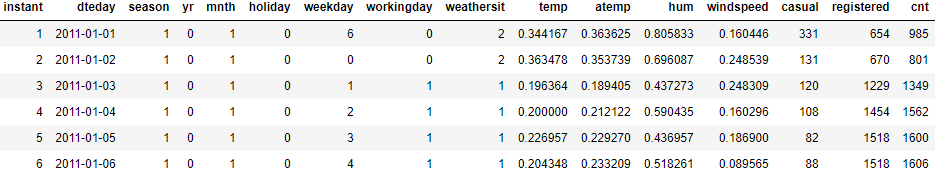
* 1. Problem Statement

The Objective of this case is Prediction of Bike rental count on daily, based on the environmental and seasonal settings.

* 1. Data

Below is the sample of data set which we are using to predict the count of bikes to be given for rental:

Table 1.1: Sample Data (Column: 1-6)



The table below shows the columns of the data set using which we have to predict the values.

Table 1.2: Columns of Data



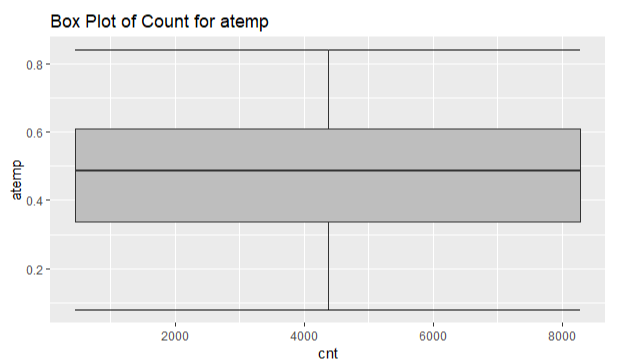
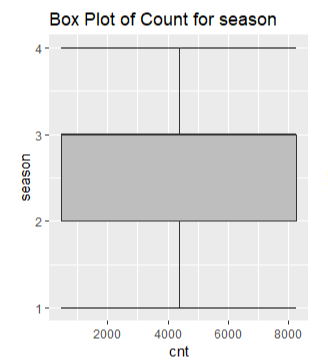
Chapter 2

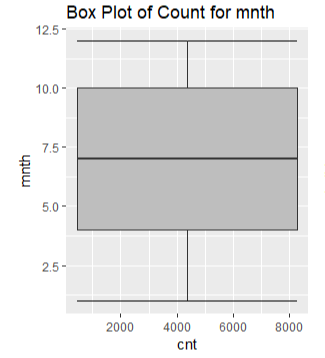
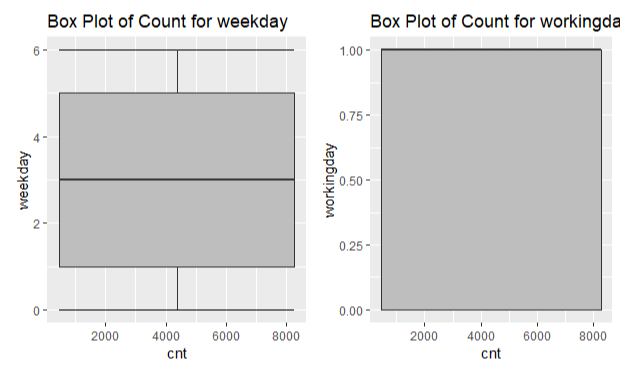
Methodology

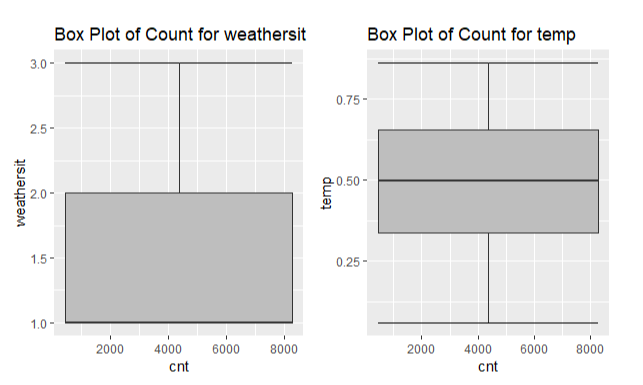
* 1. Pre Processing

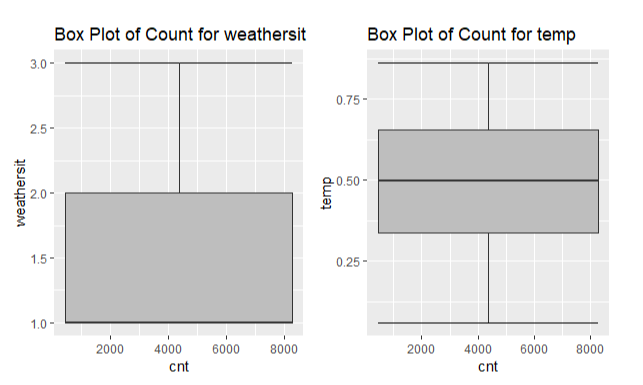
Any predictive modelling requires that we look at the data before we start modelling. However, in data mining terms looking at data refers to so much more than looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as **Exploratory Data Analysis**. To start this process we will first try to look at all the probability distributions of the variables. Most analysis like regression, requires data to be normally distributed. We can visualize that in a glance by looking at the probability distribution or probability density functions of the variable.

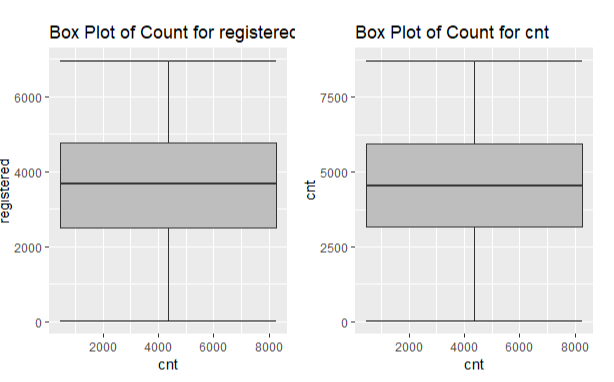
Below figures are box Plot of the Variables against the count.

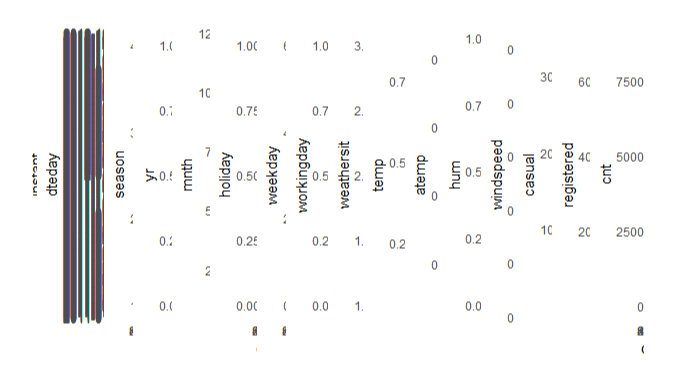
 

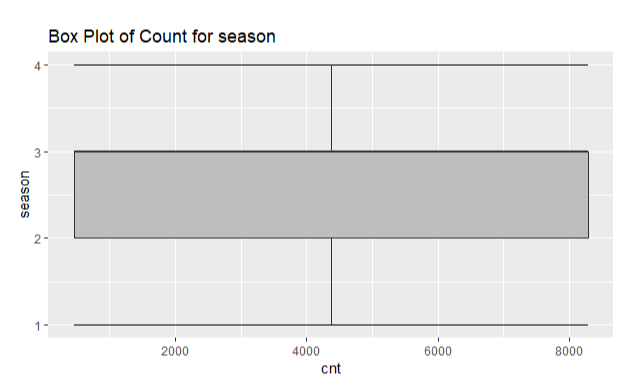
 

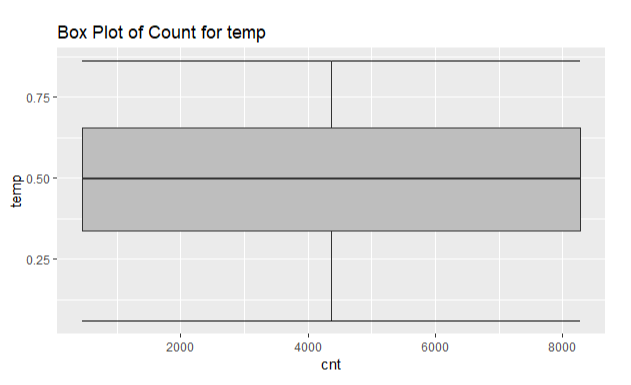










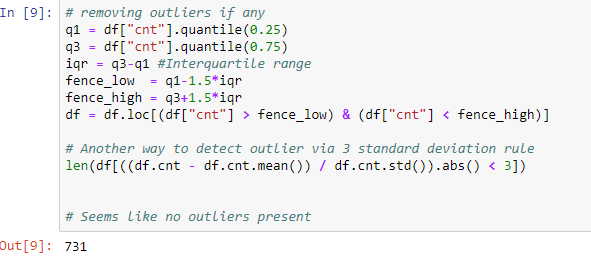


* + 1. Outlier Analysis

One of the step of Pre Processing is outlier analysis where we can see the variables having values which are varying from other values very widely.

These values are those which can be rejected or substituted using KNN.

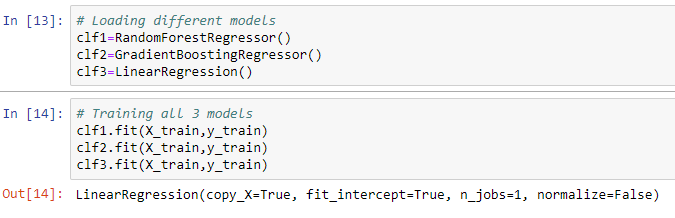
But herein case is different where we are not finding any outlier in the data saying that the data is almost normally distributed.



* 1. Modeling
     1. Model selection

Model selection can depend upon the independent variable, in our case its **count.** Here we can try and test multiple models and the one which best suits or gives the best result on train model.

You can start your model with the most simplest to the most complex and can try all these models simultaneously.



If the dependent variable is nominal the only predictive analysis that we can perform is **Classification,** and if the dependent variable is interval or ratio, the normal way is to do **regression** analysis, or classification after binning, but here in our case we do both classification and Linear Regression.

* + 1. Classification

Classification for the prediction of count cannot be done as count of anything is not a categorical value. Though the dependent variables are categorical but predicting count may not give the result expected and may have to categorize the count as well depending upon the dependent variables selected which will lead to a very hectic task.

Chapter 3

Conclusion

* 1. Model Evaluation

Now that we have few models for predicting the target variable, we need to decide which one to choose.

There are several criteria that exits for evaluating and comparing models. We can compare the models using any of the following criteria:

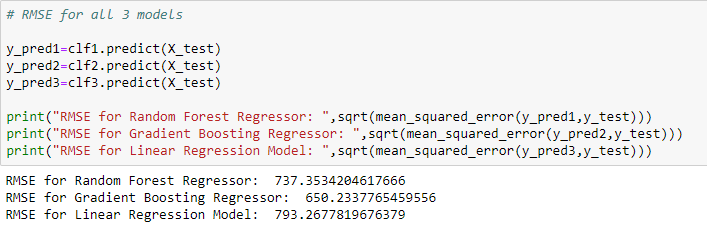
1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case i.e., predicting counts, the latter 2, Interpretability and computational Efficiency, do not hold much significance. Therefore we will use predictive performance as criteria to compare and evaluate models.

Predictive performance can be measured by comparing predictions of the models with real values of the target variables, and calculating some average error measures.

* + 1. Root Mean Square Error (RMSE)

RMSE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our model.



* 1. Model Selection

We can see that Gradient Boosting is comparatively performing better than linear regression and Random Forrest hence we will use that model to avoid any loss of information.

Appendix A – R code



Appendix B – Python Code





