Bike Renting Prediction

Vikas Soni

31st August, 2019

Contents

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In any business, knowing the quantity of resources to be kept at any point of time is a very crucial thing to know. Similar is the case for bike renting business where it is beneficial to know how many bikes are usually rented what time of a year so that the demand can be met. This project is about the same prediction using the data provided for past few years details.

1.2 Data

Our task is to predict the number of bikes that will be rented on a given day,at any given month and year. Data provided by the firm looks like this.

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
4	2011-01-04	1	0	1	0	2	1	1	0.2	0.212122	0.590435	0.160296	108	1454	1562

The variable cnt is the number of counts of bike rentings and is our variable to be predicted.
All other variables are predictors.

CHAPTER 2

METHODOLOGY

2.1 Pre-processing

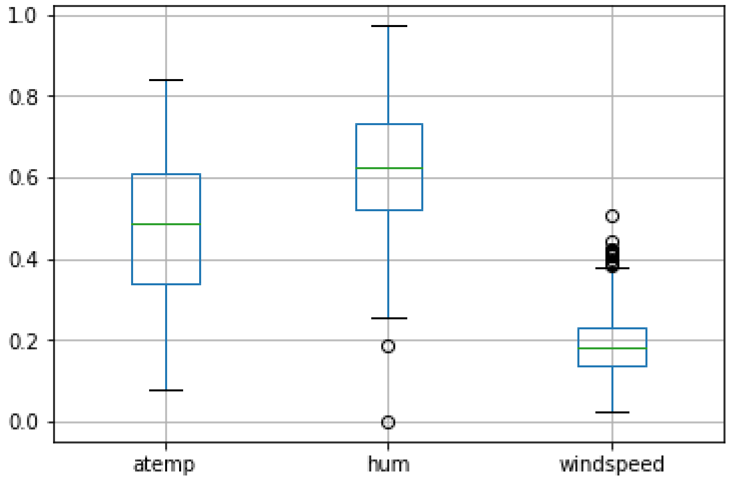
Data that is collected by different means by a company is almost always messy and wrong . Apart from that , data collected is always a lot and a lot of stuff is not even useful. So, the first step that is needed to be done is clean the data and process it in a form that works well with our machine learning model. While normally the steps to clean the data are not always fixed and different methods need to be applied with different data, there are certain things that are always done and are considered as a standard in industry. We will discuss some of the techniques here and how they prove to be useful for our case. So, lets start with missing value analysis.

2.1.1 Missing Value Analysis

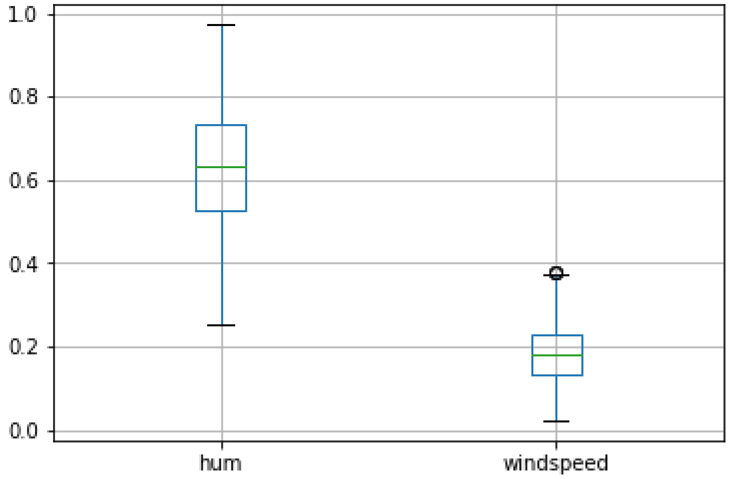
Data can be collected by various means. These can be by some sensors or from data that is recorded over a period of time or it can be from asking different people to give feedbacks and things like that. Now, in each of these cases there is always a chance that some reading are not recorded for e.g.- person denied of some information, sensors was not able to take some instants of reading, some information was not available at the time of recording .
But in our case, no missing values are found and it seems that data has always been processed a bit. So, that is a lucky thing for us.

2.1.2 Outlier Analysis

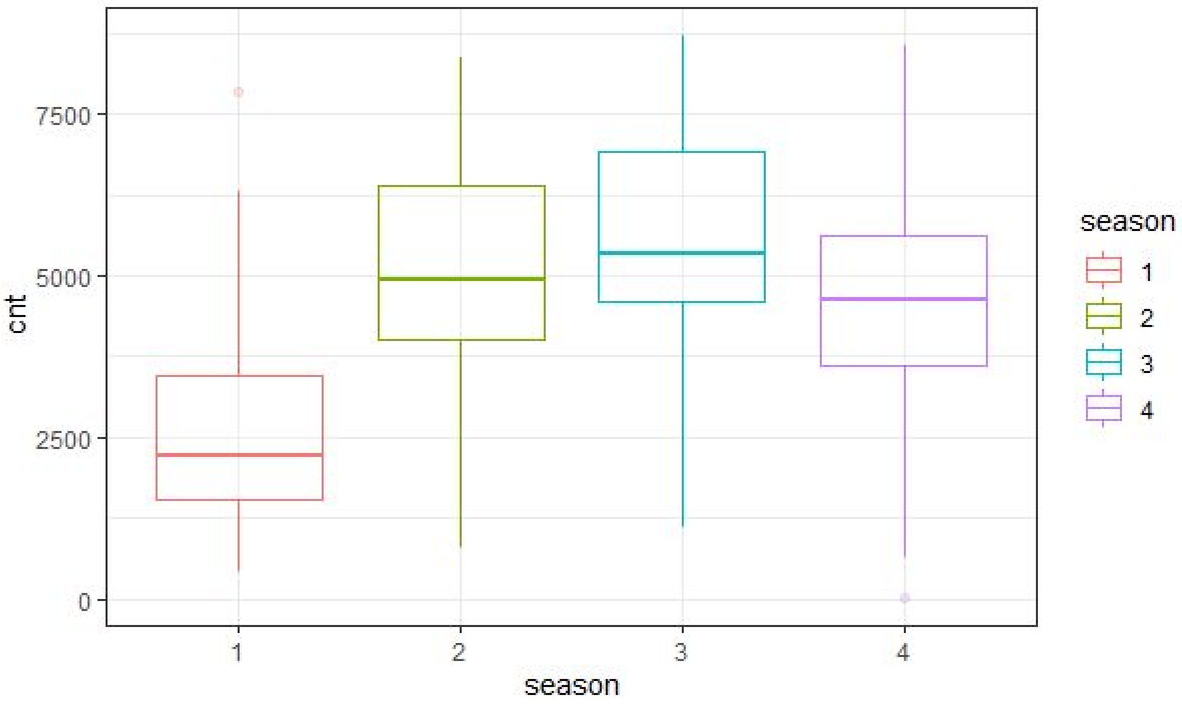
We are going to use the most common method for outlier analysis i.e. by using boxplots. Now, as we can see from those boxplots that there are some outliers in continuous variables.

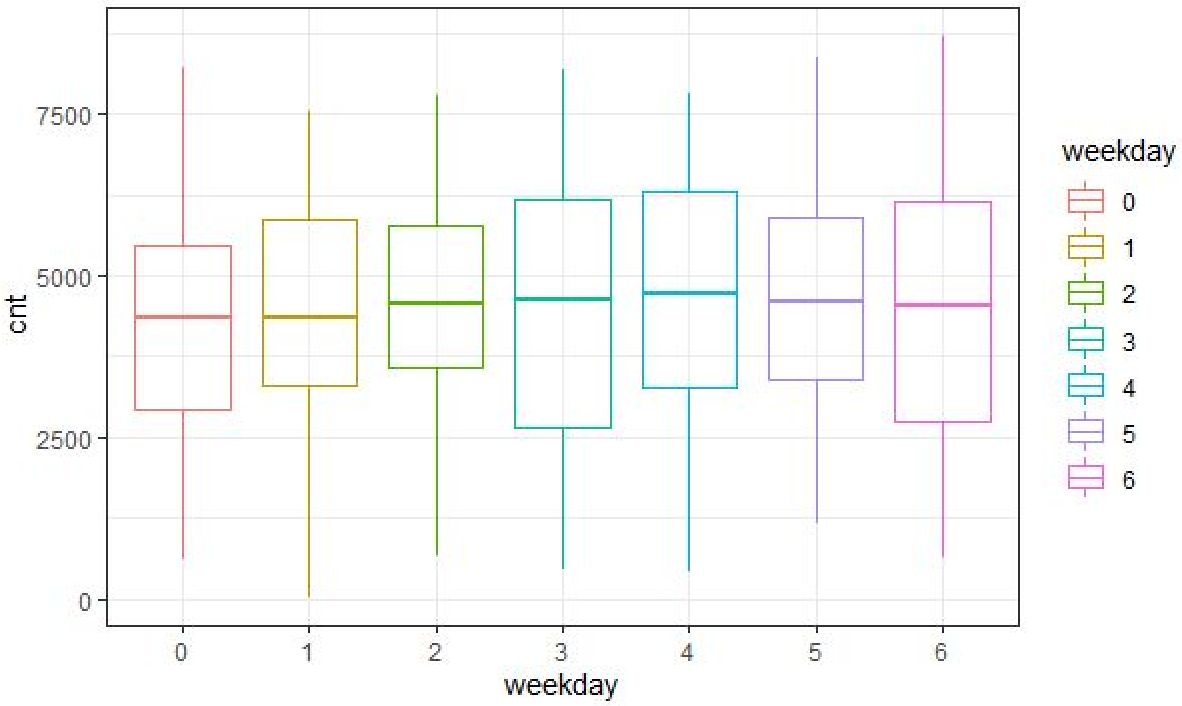


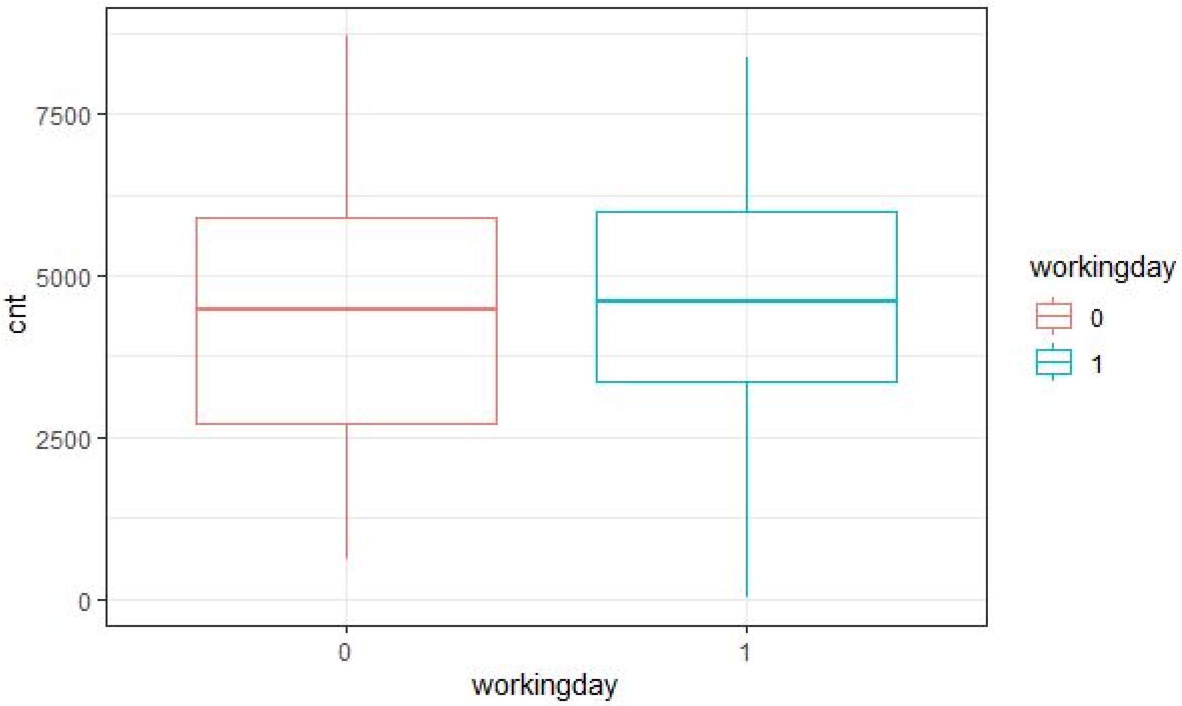
There were only three variables that were of importance in our dataset. So, we only used those and as can be seen humidity and windspeed has some outliers while atemp variable is fine. So, first of all we will remove the outliers from these outliers which will hopefully make our distribution of the variable close to normal.

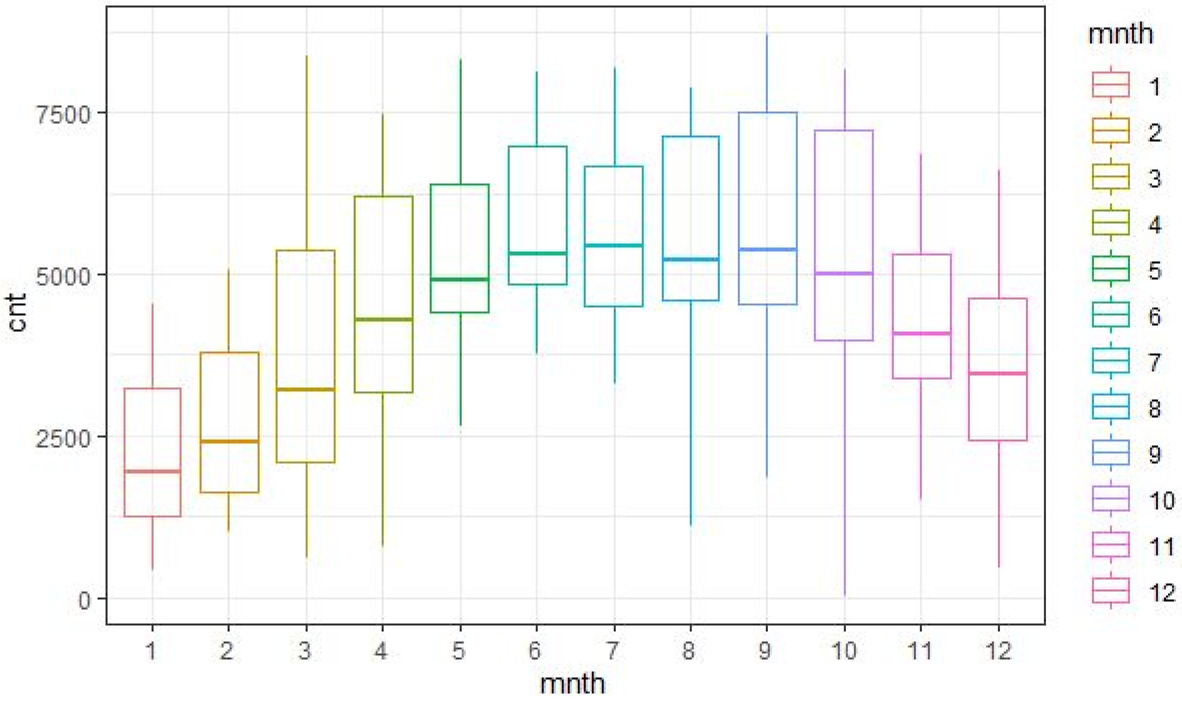


As we can see that, outliers are removed. Now, we can plot the boxplots for categorical variables so that to have more intuition for our model.


cnt
0
2500
5000
7500
1
2
3
4
season
season
1
2
3
4


cnt
0
2500
5000
7500
0
1
2
3
4
5
6
weekday
weekday
0
1
2
3
4
5
6





So, these are the relations of counts with different categorical variables. Now, these tell us some things that are a bit less obvious to us . Like if we see the boxplots for the working day and non working day , there is not much difference and it is a bit counterintuitive because we except more people to rent a bike to go to work than any other non working day. Same type of trend can be seen in the weekday column as the bike rent average count appear to be similar on any weekday. Whereas, these things are less helpful more helpful intuitions are found when season boxplots and plotted and compared. We can see that its very clear that the season with most bike rent counts is always winter. Now, this is the trend that is followed in the month column as well . All the blue graphs in mnth column are marked as blue and count there is more than anywhere else. Now, as we have some intuition about the variable distributions. We can go to our next step.

2.1.3 Feature selection

We can clearly see from our initial data that some columns are useless to us and should not be used while making predictions. First column is instant which is nothing but index so we do not need it and was removed. Now, we can see that dteday column has information about month, year and weekday which is all stored in other columns so it is obsolete and there is no harm in removing that as well. Then we will have some data which is highly correlated which is usually shown by correlation matrix but this time it is a bit more obvious than temp and atemp are the same thing and we do not necessarily need one if we are including the other and so removing that as well.Other than that , as our task is to only predict the total count of the bicycles irrespective of the fact that whether they are registered customers or casual, we can safely remove those columns from our dataset as well. All this data cleaning is necessary as in almost every case data is recorded without much care and much of the data is useless and does not contribute to the model. Now, as we are done with removing so many columns our dataset is reduced by a significant amount.

Now, when the structure of the data is checked , it is found that some of the datatypes are wrongly recognised by our platform. For e.g. the season, mnth, yr, weekday, workingday are all categorical variables and are all in the form of numbers. So, these need to be converted before feeding them to our model.

With this, we are done with our pre-processing and EDA stage and can move to model selection and prediction stage.

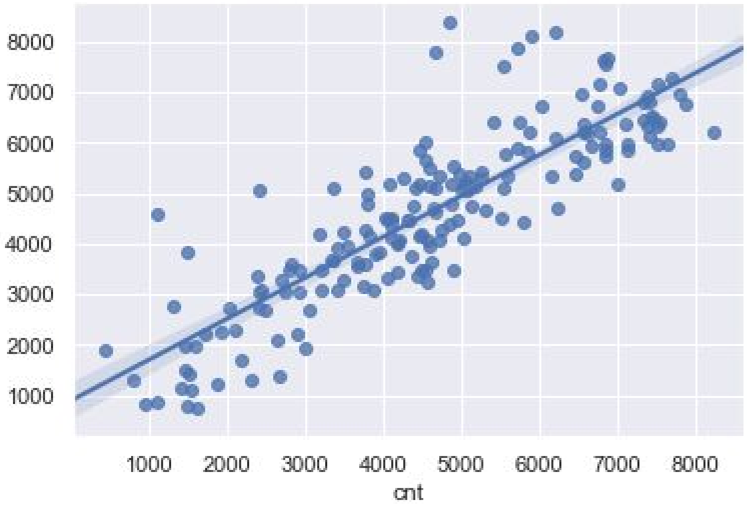
2.2 Modelling

2.2.1 Model Selection

This is a basic regression problem as we have to predict the number of rents over a given interval of time which is a number. As we saw in the early EDA stage that our dependent variable has linear relationship with temperature, we can make an inference that linear regression might be a model that can work well. So, there is no harm in trying linear regression of different kinds like ridge and lasso for the problem . Although, it is normally the case that when we have a lot of categorical variables as is in our case, decision tree based methods tends to work better than normal regression methods.So, random forest might be a good choice here. Apart from that , standard gradient boosting is a very powerful model for regression problems and we will give it a shot here.

2.2.2 Linear Regression

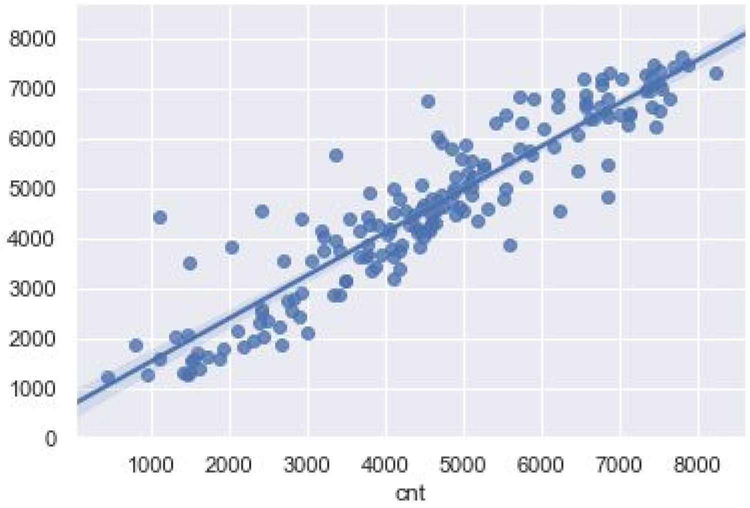
Linear regression is a simple mathematical method which will try to fit a straight line to a given set of points so that it has least distance from all the points.When this method is applied to our problem, graph comes out like this.



As we can see that this line is not giving a good prediction as some points are a bit far away. This is usually the case and reason behind it is that one straight line can not be drawn to get good results and we can rather use a curve instead of a line. Whatever be the case, the value for the MAE and MSE we got in this case were not good . So, it turns out that this model can not be used. Furthermore , there are some sophisticated versions of linear regression which are ridge and lasso regression. As checked , these methods were not found to be useful as well.

2.2.3 Standard Gradient Boosting

Standard Gradient Boosting(SGD) is a method which is more complex than the linear regression and there is a good implementation of it in python scikit learn library. This method almost always gives better results in any regression problem than any other algorithm. So, it was applied to fit our model and it turns out that it works a lot better than our linear regression and accuracy is increased by a significant amount.



As we can see from the graph that it works a lot better than linear regression but this graph only can not give as clear a representation of how much the accuracy is improved and in next section error metrics are explained which will provide us with a better insight for the same.

2.2.4 Random Forest

This method is based on decision trees. So, it tends to work better when there are a lot of categorical values which is our case. This method is applied and it is found out that however there is a significant improvement over linear regression but a bit worse than gradient boosting. But in r, we do not have as good implementation for gradient boosting as python and random forest is used there.

In inference, we have tried the most common algorithms that are used for a regression problem and compared them. Comparing algorithm is a difficult task at times and so error metrics are used. Our next section will discuss in detail about those metrics.

CHAPTER 3

CONCLUSION

3.1 Model Evaluation

As we have tried different models for our problems, it is beneficial to have a mean to tell us which of the model is working better. It is a tricky thing to find the best mathematical formula or quantity that will tell us how our model will do in real world or even on our test set. But luckily there are some standard error metrics that are generally used for a type of problem. Some of the error metrics that are applied to our model are explained below.

3.1.1 Mean Absolute Error(MAE)

It is most common metrics for machine learning algorithms. It is just mean of our absolute errors between actual values and predicted values.
Values that we got after applying this metrics to our algorithms are given as:

Linear regression : 714.497897811095
Random Forest : 514.4833333333333
Gradient Boosting : 495.5734288816848

3.1.2 Mean Squared Error(MSE)

Mean squared error is the mean of squares of errors of all the predictions.

Linear regression : 899359.9028579483
Random Forest : 531448.3463333334
Gradient Boosting : 245025.366457

As we can see that gradient boosting is working better in both those cases.

3.1.3 RMSLE

These metrics ,however, gave us a relation between algorithms but we can not really tell how good is our model as our error is quite huge. To get rid of this problem, we will use a metrics that will have smaller scale values. It works as follows. We will first take logarithm and square and find mean values of those and root in the end.

Values that we got are as follows:

Linear regression:	0.238
Gradient Boosting :	0.227

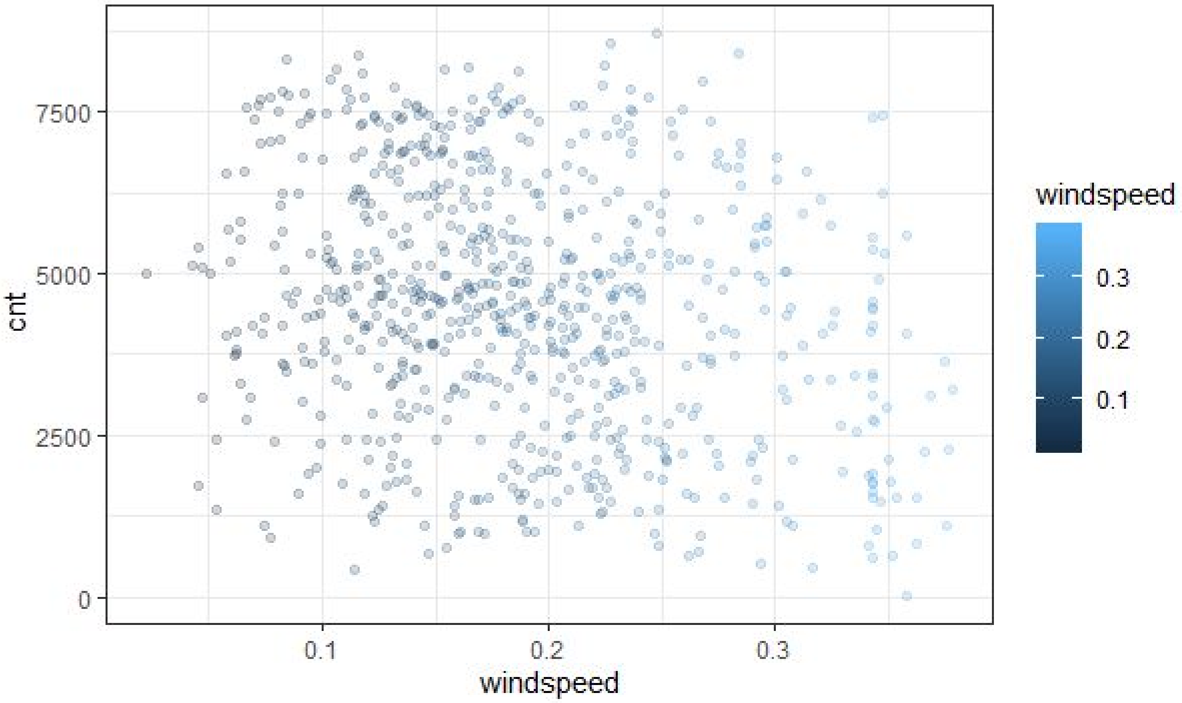
3.2 Model Selection

From all these metrics evaluation, we can easily infer that random forest and gradient boost methods are close but SGD works better. So, we will use SGD in python and Random Forest in R.

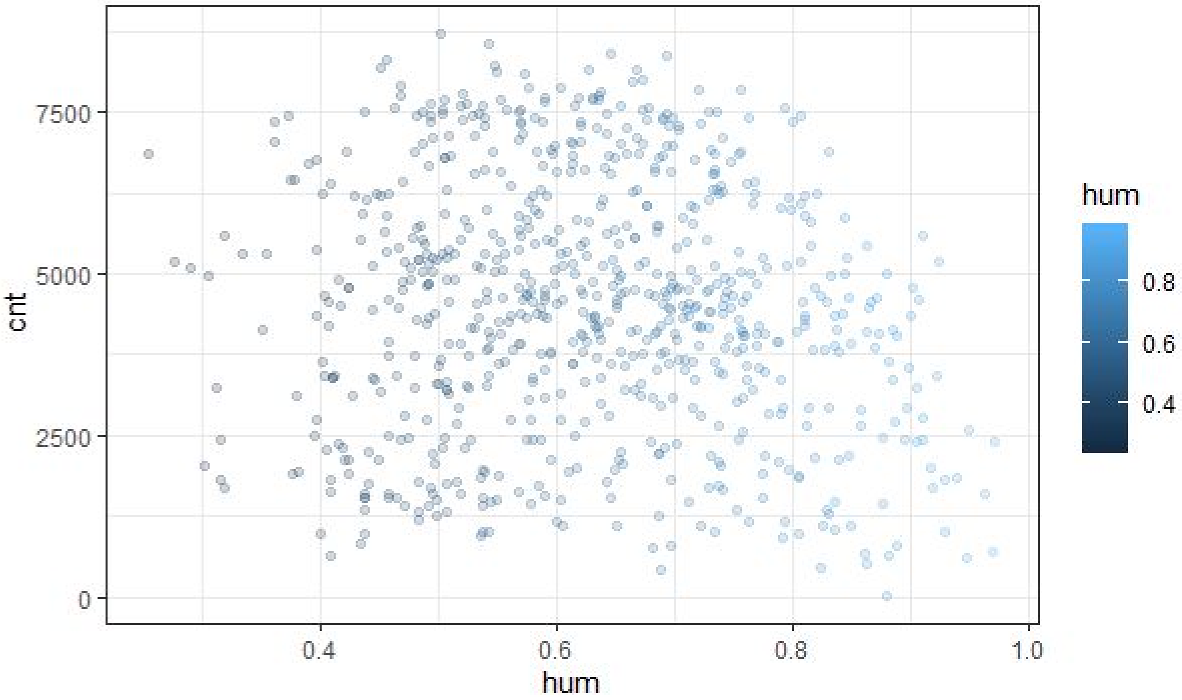
APPENDIX A

Additional Graphs and Inferences

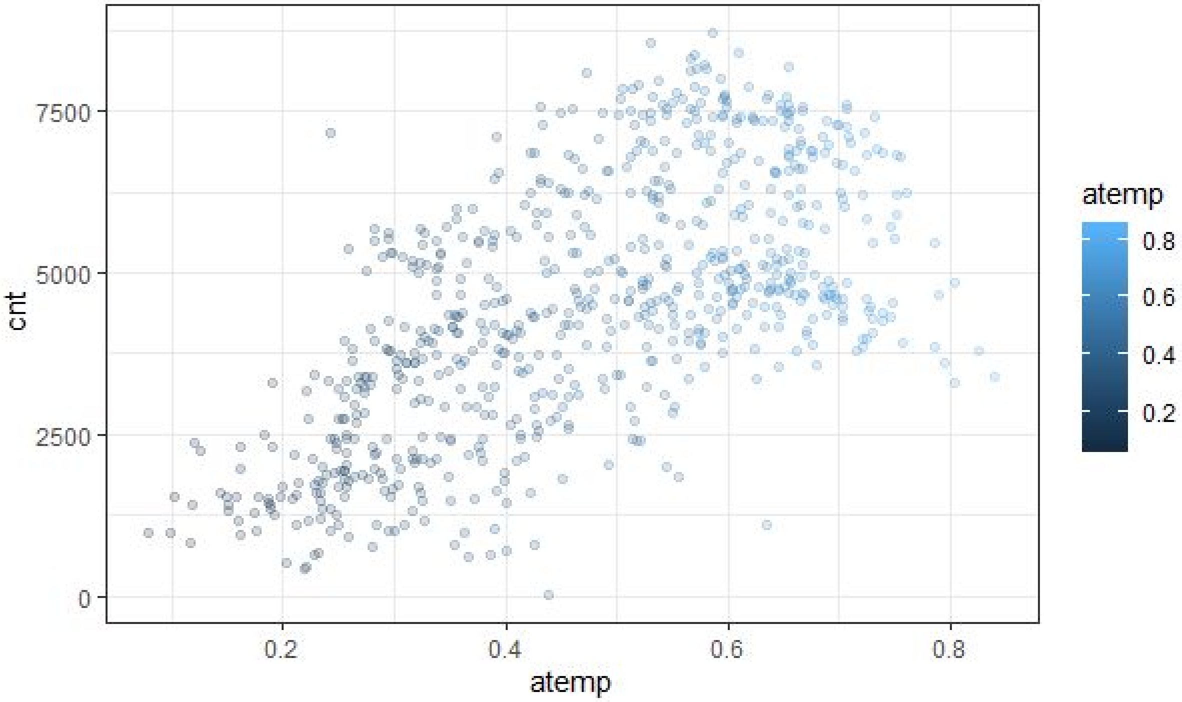
There are some graphs that were also useful in selecting models and helped to make certain assumptions . Some of them and their descriptions are:



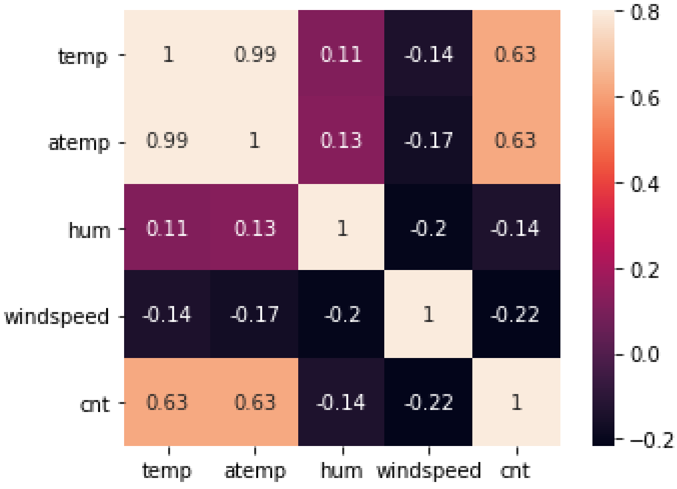
This is the relation between windspeed and count graph which is rather confusing but when inspected will tell us that counts are more when windspeed is average.



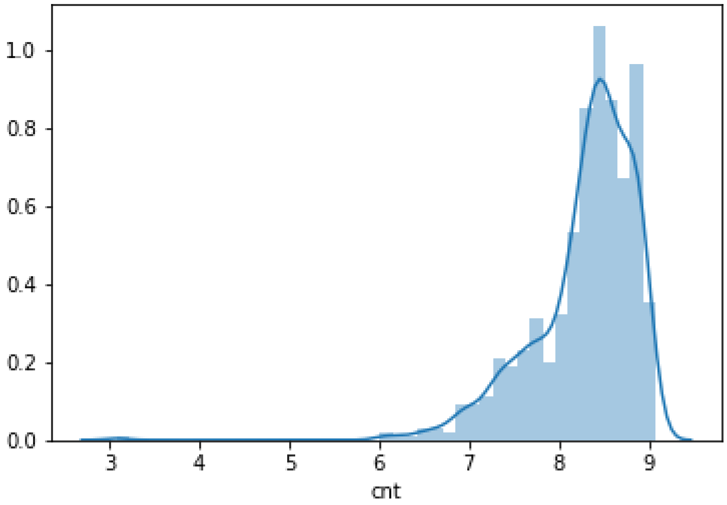
Again, relation between humidity and count suggest that counts are more in the case of average humidity which is rather obvious.



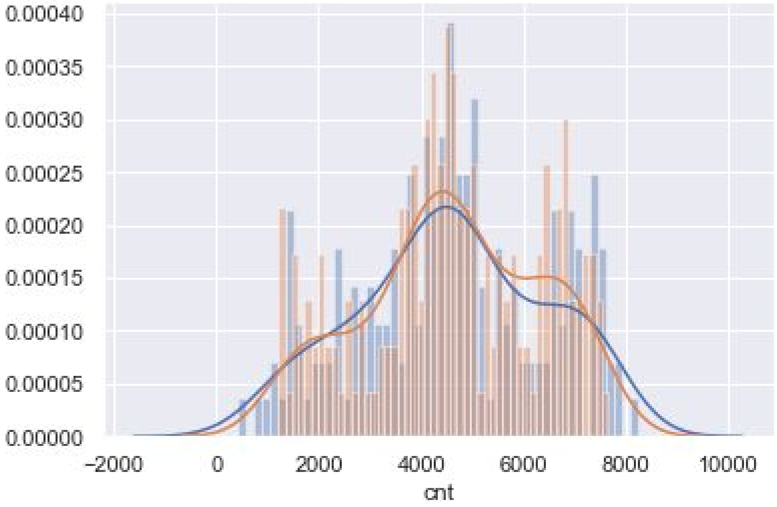
Graph showing that there is a linear relationship between counts and temperature and linear model may be applied.



Confusion matrix showing relationship correlation between various variables which helped us to detect that atemp and temp are correlated by a large extent and one of them can be removed.Also, the correlation between count and temp is highest and all others are neither too high nor too low and should be taken in consideration.



Distribution of log of count variable which is not normal but close to normal when compared to distribution of count.



Finally, in blue we have count variable and in red we have our predicted values which are not too bad.The model can be improved a bit more by using libraries like xgboost and more distribution control methods but this model seems to be working fine with not so much work and less variables to take care of.