Project Report

on

Predicting Daily Bike Rental

Submitted by:
Prashant Nookala
Date: 25/08/2019

Contents

Executive Summary

Prashant Nookala is pleased to submit a Project report on Daily Bike Rental Prediction. This document states the approach to the Data Set provided by Edwisor for Bike rentals and the algorithms used to predict the daily bike rental.

Assumptions:

- Data will be provided to me in the normalized format as provided for this project.

1. Introduction

1.1 Problem Statement

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

1.2 Data

Task is to build a regression model to identify the count of daily bike rentals depending on environmental & seasonal settings. Below is the sample dataset we are using to predict the daily rental count:

	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

Figure 1: Sample Dataset

Our dataset contains 731 observations and 16 variables. Following are the 16 variables in the dataset:

S.No.	Variables
1	instant
2	dteday
3	season
4	yr
5	mnth
6	holiday
7	weekday
8	workingday
9	weathersit
10	temp
11	atemp
12	hum
13	windspeed
14	casual
15	registered
16	cnt

Table 1: Variables

From these 16 variables 13 variables except casual, registered, cnt are predictors.

2. Methodology

2.1 Pre-Processing

Data pre-processing is the major step before putting the data into ML models/algorithms. It is very important to look at the data types, distribution of Data, missing values, outliners to standardize or Normalize the data if required. Also it is important to check the multicollinearity of variables so that we can select the variables which is important to predict the target variable.

In this case as all the predictors are already normalized, we need not to go through the process of normalization or standardization. Below is are the data types of all the variables:

S.No.	Variables	Data Type
1	instant	int64
2	dteday	Object
3	season	int64
4	yr	int64
5	mnth	int64
6	holiday	int64
7	weekday	int64
8	workingday	int64
9	weathersit	int64
10	temp	float64
11	atemp	float64
12	hum	float64
13	windspeed	float64
14	casual	int64
15	registered	int64
16	cnt	int64

Table 2: Variable Data Types

2.1.1 Missing Value analysis

After analysing the data, we can see that there are no missing values in this dataset

2.1.2 Distribution & Outlier Analysis

In the below figures, we can see the distribution & box plot for variables to understand the data distribution and see if there are any outliers in the dataset.

In Figure 2 & 3, we can see the distribution of factor & integer predictors respectively.

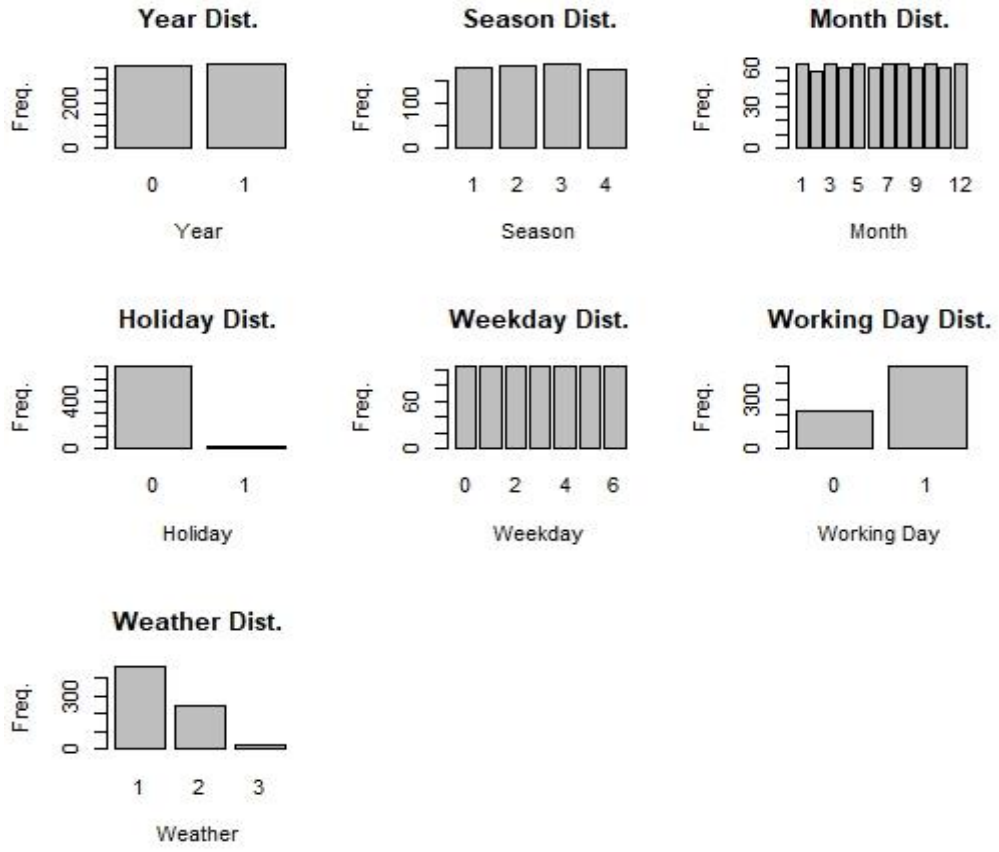

Figure 2: Bar plot of Factor Predictor Variables

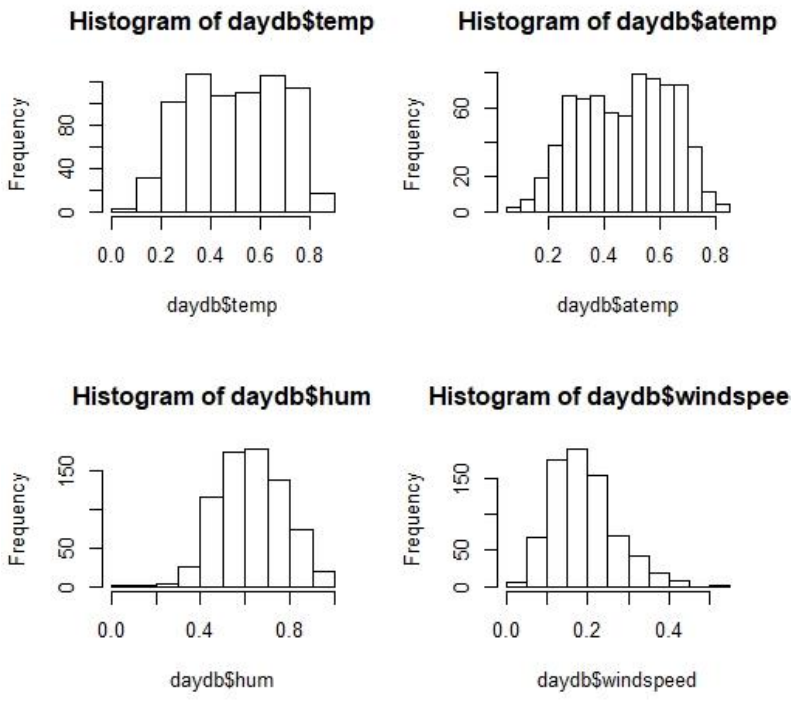

Figure 3: Histogram of Integer Predictor variables

From Figure 2, we can say that data is evenly distributed for these variables and from Figure 3, data in variables temp and atemp is evenly distributed however, hum is left skewed and windspeed is right skewed.

In the below figures, we can see the distribution & outliers (if any) of target variables.

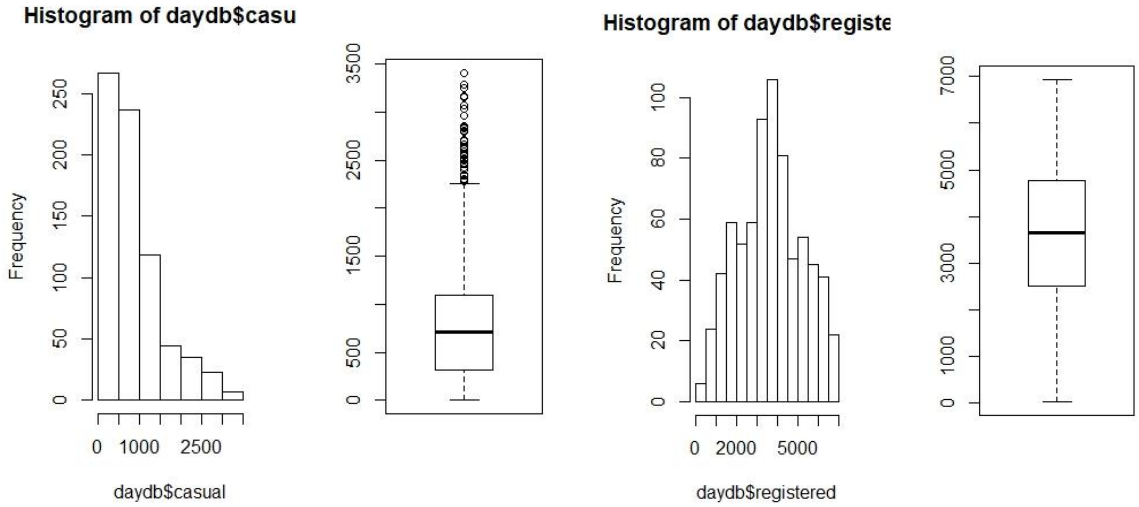

Figure 4: Histogram and box plot of casual & registered

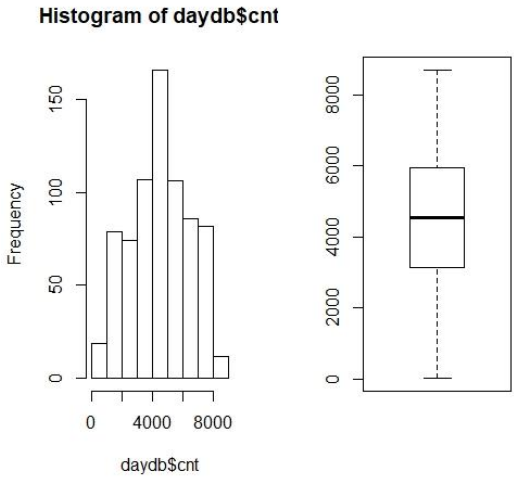

Figure 5: Histogram and box plot of cnt

In the above Figure 4, we can see that distribution of casual variables is right skewed & there are no. of outliers. However, registered variable is normally distributed and also from Figure 5, cnt is also normally distributed and also there are no outliers in both these variables. As we need to predict the daily total bike rentals, we need to predict the total count of bike rentals on a day. So, as we don't have outliers in cnt. However, as it has wide distribution, min=22 and max=8714, we will be taking its log value in the model.

2.1.2 Feature Selection

It is very important to select the variables which can actually impact the target variable, so we will perform the correlation analysis to check the multicollinearity between the variables. Below is the correlation table & figure.

	Season	yr	mnth	holiday	weekday	workingday	weathersit
Season	1	-0.00184	0.83144	-0.01054	-0.00308	0.012485	0.019211
Yr	-0.00184	1	-0.00179	0.007954	-0.00546	-0.00201	-0.04873
Mnth	0.83144	-0.00179	1	0.019191	0.009509	-0.0059	0.043528
Holiday	-0.01054	0.007954	0.019191	1	-0.10196	-0.25302	-0.03463
weekday	-0.00308	-0.00546	0.009509	-0.10196	1	0.03579	0.031087
workingday	0.012485	-0.00201	-0.0059	-0.25302	0.03579	1	0.0612
weathersit	0.019211	-0.04873	0.043528	-0.03463	0.031087	0.0612	1
Temp	0.334315	0.047604	0.220205	-0.02856	-0.00017	0.05266	-0.1206
Atemp	0.342876	0.046106	0.227459	-0.03251	-0.00754	0.052182	-0.12158
Hum	0.205445	-0.11065	0.222204	-0.01594	-0.05223	0.024327	0.591045
windspeed	-0.22905	-0.01182	-0.2075	0.006292	0.014282	-0.0188	0.039511
Casual	0.210399	0.248546	0.123006	0.054274	0.059923	-0.51804	-0.24735
registered	0.411623	0.594248	0.293488	-0.10875	0.057367	0.303907	-0.26039
Cnt	0.4061	0.56671	0.279977	-0.06835	0.067443	0.061156	-0.29739

	Temp	atemp	Hum	windspeed	casual	registered	cnt
Season	0.334315	0.342876	0.205445	-0.22905	0.210399	0.411623	0.4061
Yr	0.047604	0.046106	-0.11065	-0.01182	0.248546	0.594248	0.56671
Mnth	0.220205	0.227459	0.222204	-0.2075	0.123006	0.293488	0.279977
Holiday	-0.02856	-0.03251	-0.01594	0.006292	0.054274	-0.10875	-0.06835
weekday	-0.00017	-0.00754	-0.05223	0.014282	0.059923	0.057367	0.067443
workingday	0.05266	0.052182	0.024327	-0.0188	-0.51804	0.303907	0.061156
weathersit	-0.1206	-0.12158	0.591045	0.039511	-0.24735	-0.26039	-0.29739
Temp	1	0.991702	0.126963	-0.15794	0.543285	0.540012	0.627494
Atemp	0.991702	1	0.139988	-0.18364	0.543864	0.544192	0.631066
Hum	0.126963	0.139988	1	-0.24849	-0.07701	-0.09109	-0.10066
windspeed	-0.15794	-0.18364	-0.24849	1	-0.16761	-0.21745	-0.23455
Casual	0.543285	0.543864	-0.07701	-0.16761	1	0.395282	0.672804
registered	0.540012	0.544192	-0.09109	-0.21745	0.395282	1	0.945517
Cnt	0.627494	0.631066	-0.10066	-0.23455	0.672804	0.945517	1

Table 3: Correlation Table

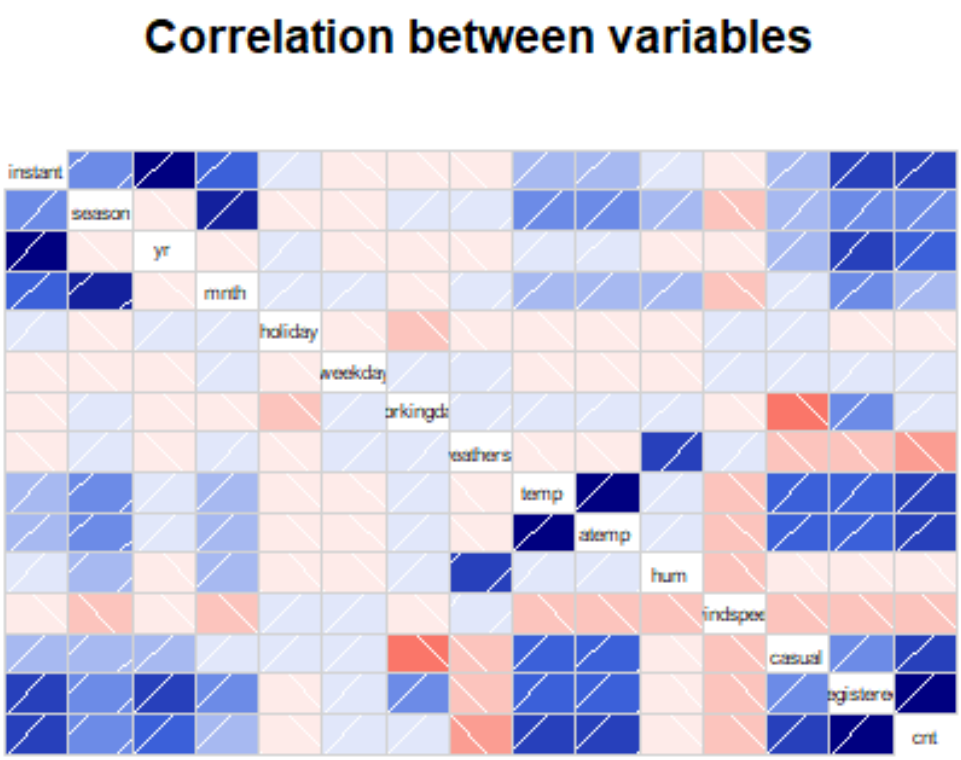

Figure 6: Correlation between variables

In this dataset, as instant & dateday is unique for each observation, they won't have any impact on the target value so we can remove it. Also, in the above matrix & fig. we can see that season & mnth and temp & atemp variables are highly correlated. So, I am omitting mnth & atemp from the dataset. Also, as we need not to predict casual & registered bike rentals on a day, we can also omit casual & registered from the dataset.

2.1.3 Feature Scaling

As discussed in the section, 2.1.1, we will need to add a variable cntlog, which will be log10 (cnt). It will help us negate the wide distribution of cnt variable and will be passing cntlog variable excluding cnt variable to the model.

#Cntlog = log10(cnt)

2.2 Modeling

2.2.1 Model Selection

In our early stages of pre-processing we understood that observations in our target variable is interval type of data, so we need to predict the cnt using regression techniques. Here data is not segregated into train & test, so we need to divide the data into train & test. So after sampling data into test & train, we have 584 observations in train and 147 in test.

Now I have used all regression models as shown below and calculated the error using MAPE error matric. Below is the R code:

I.
```
# Error Metrics
#calculate MAPE
MAPE = function(y, yhat){
  mean(abs((y - yhat)/y))
}
```

II.
```
#Decision Tree Regression
 library(rpart)
 d_fit = rpart(cntlog ~ ., data = train, method = "anova")
 pred_new = predict(d_fit, test[,-(which(colnames(test) == "cntlog"))])

 MAPE(test[,which(colnames(test) == "cntlog")], pred_new)

 # Error Rate: 0.02219195
 # Accuracy: 97.78%
```

III.
```
#Liner Regression Model
install.packages("usdm")
library(usdm)
lm_model = lm(cntlog ~ ., data = train)
predictions_LR = predict(lm_model, test[,-(which(colnames(test) == "cntlog"))])

MAPE(test[,10], predictions_LR)

# Error Rate: 0.02601011
# Accuracy: 97.39%
```

IV.
```
###Random Forest Model
library(randomForest)

RF_model = randomForest(cntlog ~ ., train, importance = TRUE, ntree = 100)
pred_RF = predict(RF_model, test[, -(which(colnames(test) == "cntlog"))])

MAPE(test[,which(colnames(test) == "cntlog")], pred_RF)

# Error Rate: 0.01712123
# Accuracy: 98.28%
```

V. #Logistic Regression

```
logit_model = glm(cntlog ~ ., data = train, family = "quasi")
pred_logit = predict(logit_model, test[, -(which(colnames(test) == "cntlog"))])

MAPE(test[,10], pred_logit)
```

 # Error Rate: 0.02601011
 # Accuracy: 97.39%

3. Conclusion

As we can see from the above code; after calculation of MAPE on each of these regression models, we found that error rate for Random Forest Model with 100 trees is less i.e. 0.01712 (in R) and 0.01802 (in Python), hence accuracy from this model comes out to be 98.28% (in R) & 98.18 (in Python). Below is comparison graph of Original Daily Count & Predicted Daily Count:

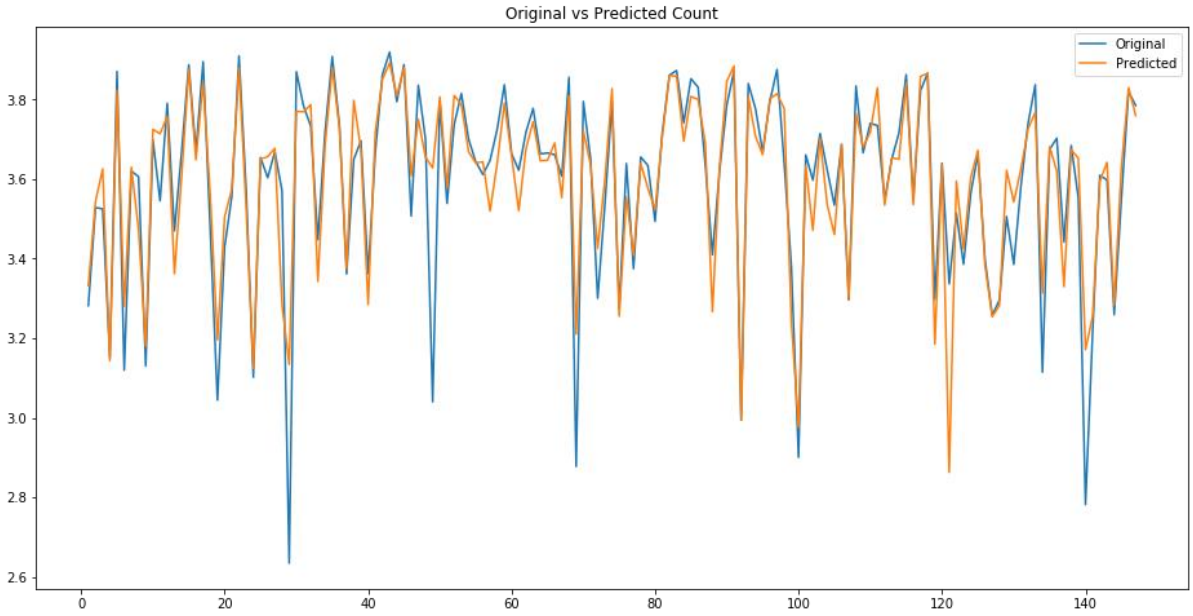

Figure 7: Original vs Predicted

3.1 Sample Data & Output

Input Value:

instant:1000, dteday:10-05-2014, season:2, yr:4, mnth:5, holiday:0, weekday:6, workingday:0, weathersit:1, temp:0.5325, atemp:0.522721, hum:0.489167, windspeed: 0.115671

Output Value:

cnt = 7363