



Bike Renting

Sahil Chahal
25/9/2018

Contents

1 Introduction

1.1 Problem Statement

1.2 Data

2 Methodology

2.1 Pre Processing

2.1.1 Exploratory Data Analysis

2.1.2 Missing Value Analysis

2.1.3 Outlier Analysis

2.1.4 Feature Selection

2.1.5 Visualizations

2.2 Modeling

2.2.1 Dummy variables

2.2.2 Model Selection

2.2.3 Classification model

2.2.3 Regression Trees

2.2.4 Multiple linear regression

3 Conclusion

3.1 Model Evaluation

3.1.1 Mean Absolute percentage error (MAPE)

3.2 Model Selection

1. INTRODUCTION

1.1 Problem statement

A bike rental is a bicycle business that rents bikes for short periods of time. Most rentals are provided by bike shops as a sideline to their main businesses of sales and service, but some shops specialize in rentals. Bike rental shops rent by the day or week as well as by the hour, and these provide an excellent opportunity for people who don't have access to a vehicle, typically travelers and particularly tourists. Specialized bike rental shops thus typically operate at beaches, parks, or other locations that tourists frequent. In this case, the fees are set to encourage renting the bikes for a few hours at a time, rarely more than a day. The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings so that required bikes would be arranged and managed by the shops according to environmental and seasonal conditions.

1.2 Data

Our task is to build regression models which will predict the count of bike rented depending on various environmental and seasonal conditions Given below is a sample of the data set that we are using to predict the count of bike rents:

Table 1.1: Sample Data (Columns: 1-8)

instant	dteday	season	yr	mnth	holiday	weekday	workingday
1	1/1/2011	1	0	1	0	6	0
2	1/2/2011	1	0	1	0	0	0
3	1/3/2011	1	0	1	0	1	1
4	1/4/2011	1	0	1	0	2	1
5	1/5/2011	1	0	1	0	3	1
6	1/6/2011	1	0	1	0	4	1

Table 1.2: Sample Data (Columns: 7-16)

weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	0.363478	0.353739	0.696087	0.248539	131	670	801
1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
1	0.2	0.212122	0.590435	0.160296	108	1454	1562
1	0.226957	0.22927	0.436957	0.1869	82	1518	1600
1	0.204348	0.233209	0.518261	0.0895652	88	1518	1606

Variables present in given dataset are instant, dteday, season, yr, mnth, holiday, weekday, workingday, weathersit, temp, atemp, hum, windspeed, casual, registered, cnt

The details of variable present in the dataset are as follows -
instant: Record index

dteday: Date

season: Season (1:springer, 2:summer, 3:fall, 4:winter)

yr: Year (0: 2011, 1:2012)

mnth: Month (1 to 12)

hr: Hour (0 to 23)

holiday: weather day is holiday or not (extracted fromHoliday Schedule)

weekday: Day of the week

workingday: If day is neither weekend nor holiday is 1, otherwise is 0.

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

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
 $(t - t_{\min}) / (t_{\max} - t_{\min})$,

$t_{\min} = -16$, $t_{\max} = +50$ (only in hourly scale)

hum: 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: count of registered users

cnt: count of total rental bikes including both casual and registered

Methodology

2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms *looking at data* refers to so much more than just 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.

2.1.1 Exploratory Data Analysis

In exploring the data we have

- Converted season, yr, mnth, holiday, weekday, workingday, weathersit into categorical variables.
- Changed de day variables's date value to day of date and converted to categorical variable having 31 levels as a month has 31 days.
- Deleted instant variable as it is nothing but an index.
- Omitted registered and casual variable as sum of registered and casual is the total count that is what we have to predict.

2.1.2 Missing Value Analysis

Missing value analysis is done to check is there any missing value present in given dataset. Missing values can be easily treated using various methods like mean, median method, KNN method to impute missing value.

In R `function(x){sum(is.na(x))}` is the function used to check the sum of missing values.

In Python `.isnull().sum()` is the function used to check the sum of the missing values

There is no missing value found in given dataset.

Filter	
apply.data..2..function.x...	
instant	0
dteday	0
season	0
yr	0
mnth	0
holiday	0
weekday	0
workingday	0
weathersit	0
temp	0
atemp	0
hum	0
windspeed	0
cnt	0

Figure 2.1 Missing Value analysis

2.1.3 Outlier Analysis

Outlier analysis is done to handle all inconsistent observations present in given dataset. As outlier analysis can only be done on continuous variable.

Figure 2.1 and 2.2 are visualization of numeric variable present in our dataset to detect outliers using boxplot. Outliers will be detected with red color

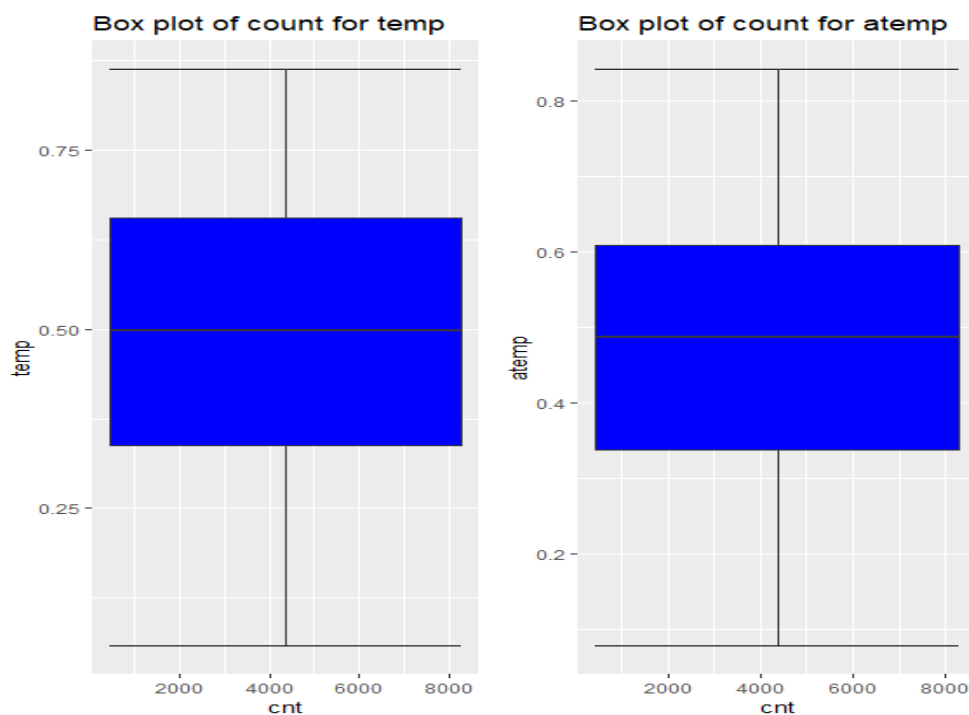


Figure 2.2 Boxplot graph of temp and atemp variables

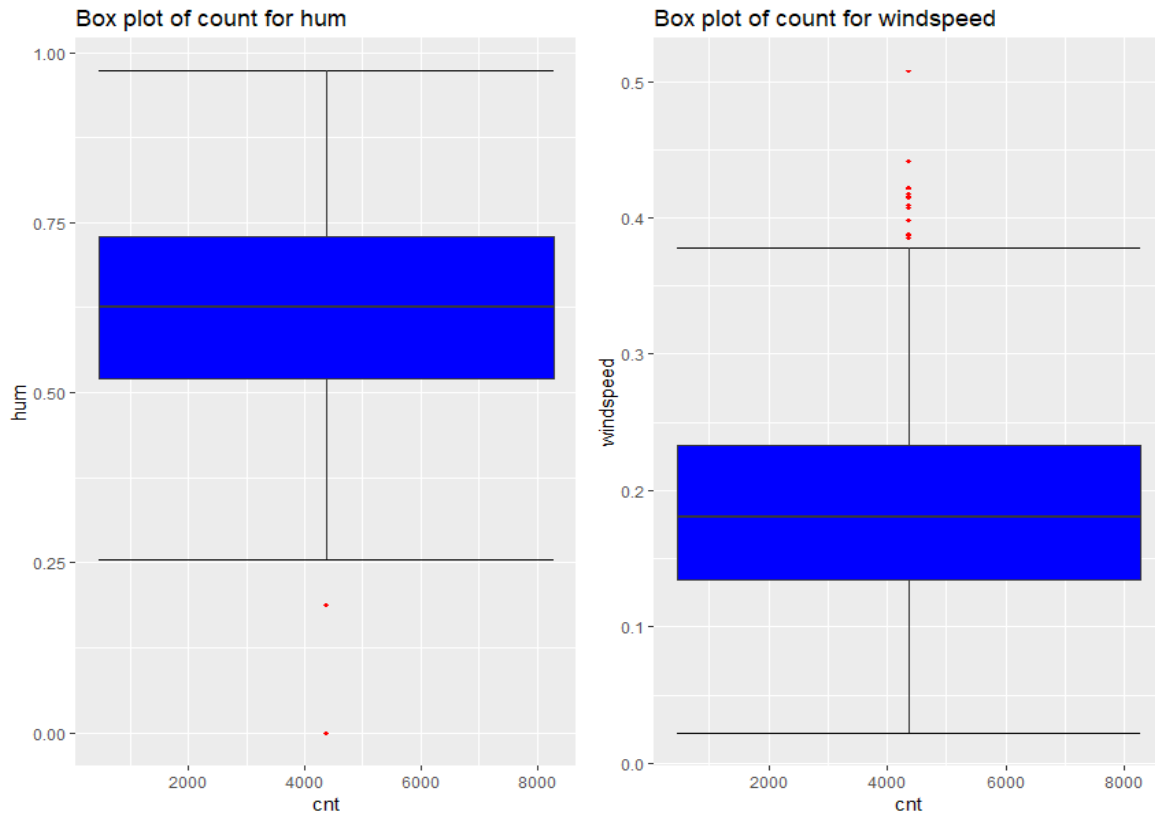


Figure 2.3 Boxplot graph of hum and windspeed variables

According to above visualizations there is no outlier found in temp and atemp variable but there are few outliers found in windspeed and hum variable. As windspeed variable defines the windspeed on a particular day and hum defines the humidity of that day so we can neglect these outliers because both these variable define environmental condition. Due to drastic change in weather like strome, heavy rain condition.

2.1.4 Feature Selection

Feature selection analysis is done to Select subsets of relevant features (variables, predictors) to be in model construction.

As our target variable is continuous so we can only go for correlation check. As chi-square test is only for categorical variable.

Figure 2.4 show a correlation plot for all numeric variable present in dataset

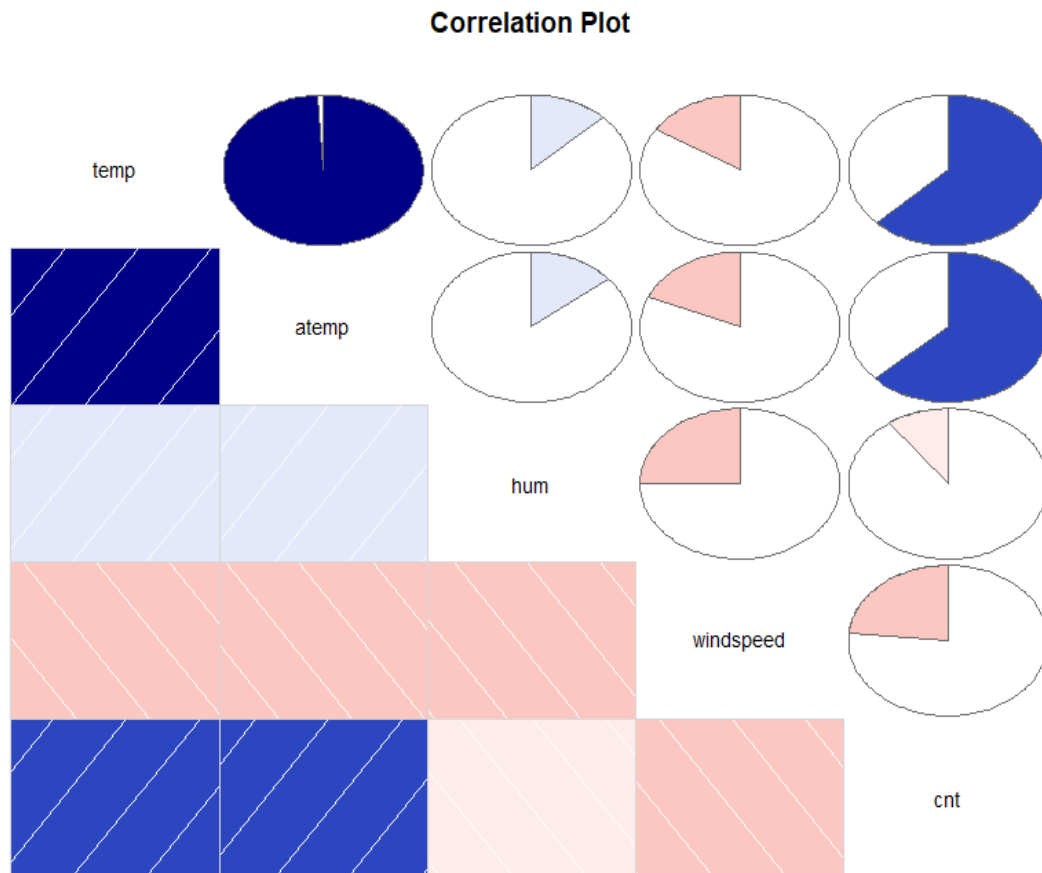
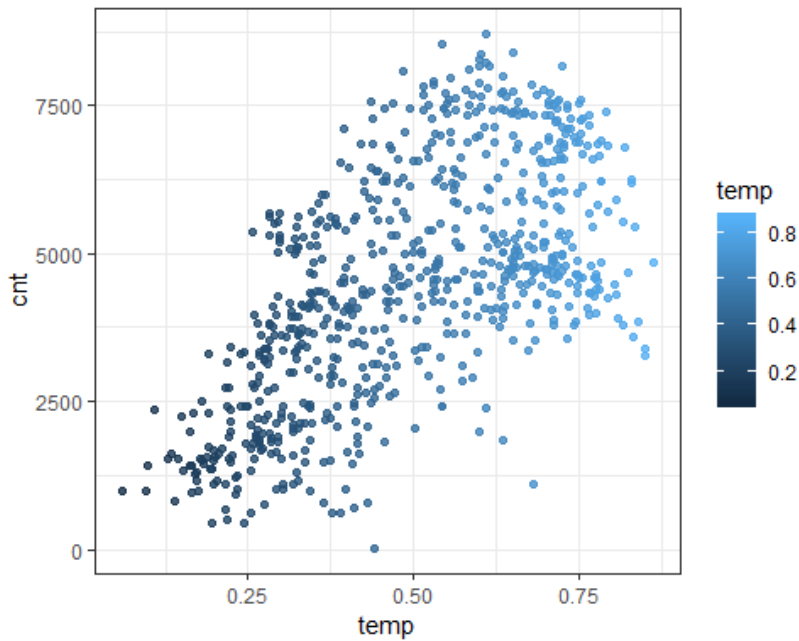


Figure 2.4 correlation plot

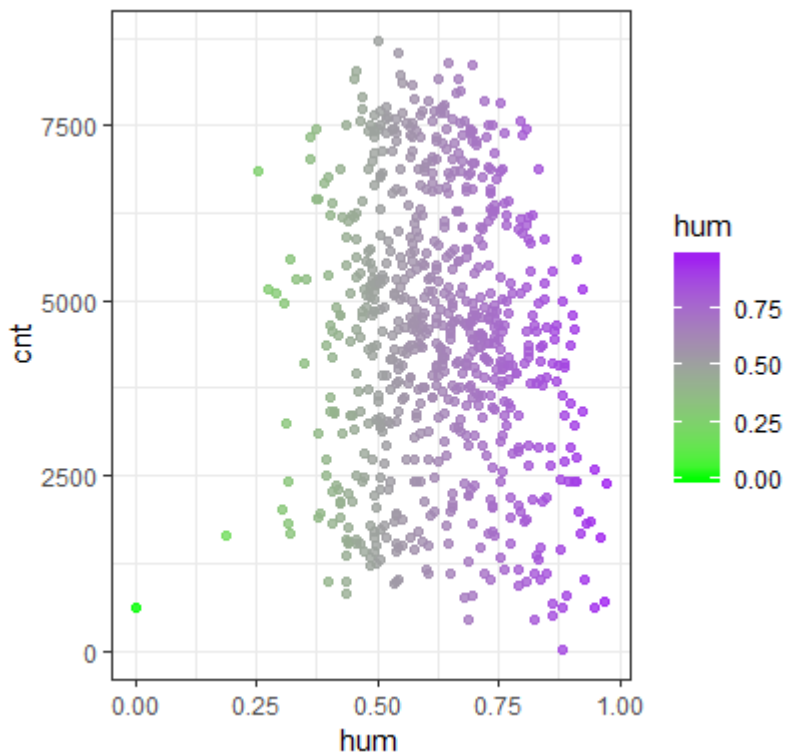
In above visualization we can see that only 2 variables are highly correlated with each other. Dark blue color represent highly correlated and light color represent very less correlated so we have a choice to remove either temp or atemp because these variables contains nearly equal information. So I have removed atemp variable from dataset.

2.1.5 Visualizations

Data visualization is the general term that describes any effort to help people understand the significance of data by placing it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data visualization software.



This is a scatterplot of temperature versus count with a color gradient based on temperature. The plot depicts the bike rental count increases as the temperature increases.



This is a scatterplot of humidity versus count with a color gradient based on humidity. The plot depicts the bike rental count increases as the humidity between 0.50 to 0.25.

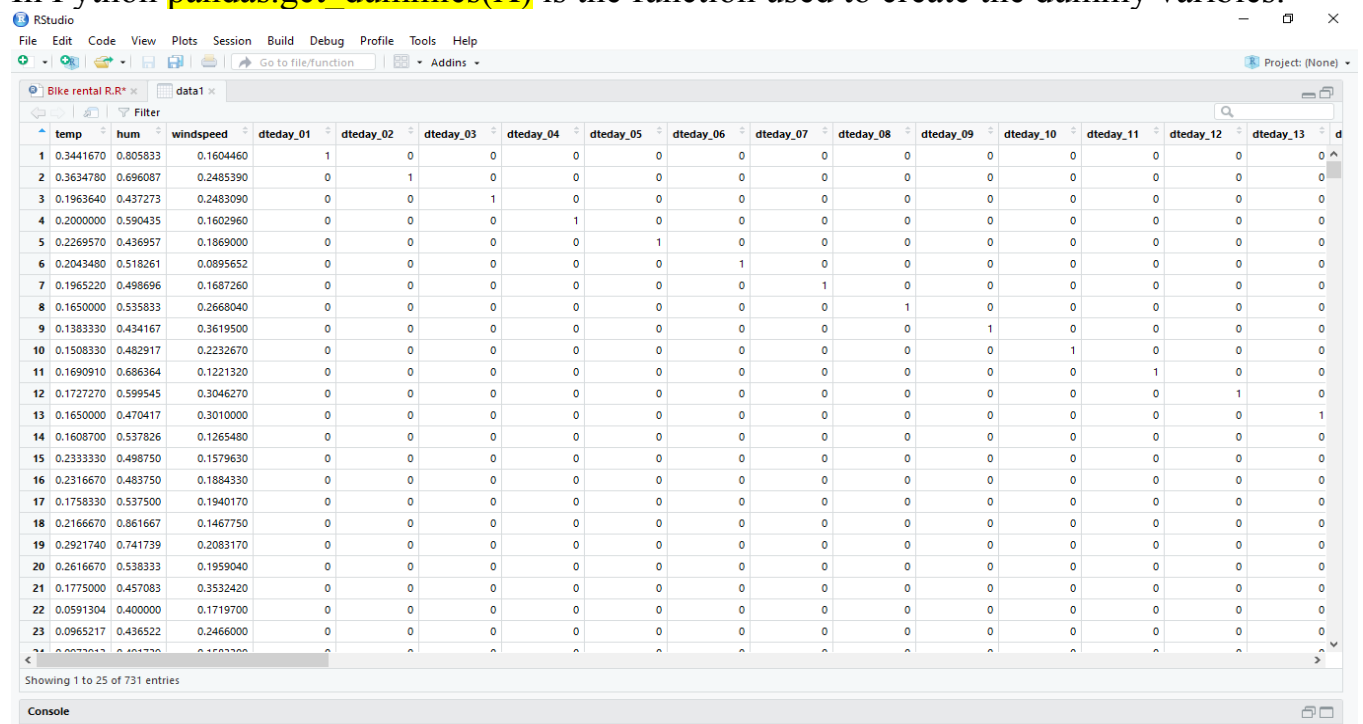
2.2 Modeling

2.2.1 Dummy Variables

Here we have some categorical variables like season, weather, month. To include these in our model, we'll need to make binary dummy variables.

In R `fastDummies::dummy_cols(X)` is the function used to create the dummy variables.

In Python `pandas.get_dummies(X)` is the function used to create the dummy variables.



The screenshot shows the RStudio interface with a data frame named 'data1' loaded. The data frame contains 25 rows and 17 columns. The columns are: temp, hum, windspeed, dteday_01, dteday_02, dteday_03, dteday_04, dteday_05, dteday_06, dteday_07, dteday_08, dteday_09, dteday_10, dteday_11, dteday_12, dteday_13, and d. The data is displayed in a table with 25 rows and 17 columns. The first 25 rows are shown, and the total number of entries is 731.

	temp	hum	windspeed	dteday_01	dteday_02	dteday_03	dteday_04	dteday_05	dteday_06	dteday_07	dteday_08	dteday_09	dteday_10	dteday_11	dteday_12	dteday_13	d
1	0.3441670	0.805833	0.1604460	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0.3634780	0.696087	0.2485390	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0.1963640	0.437273	0.2483090	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0.2000000	0.590435	0.1602960	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0.2269570	0.436957	0.1869000	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6	0.2043480	0.518261	0.0895652	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7	0.1965220	0.498696	0.1687260	0	0	0	0	0	0	1	0	0	0	0	0	0	0
8	0.1650000	0.535833	0.2668040	0	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0.1383330	0.434167	0.3619500	0	0	0	0	0	0	0	0	1	0	0	0	0	0
10	0.1508330	0.482917	0.2232670	0	0	0	0	0	0	0	0	0	1	0	0	0	0
11	0.1690910	0.686364	0.1221320	0	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0.1727270	0.599545	0.3046270	0	0	0	0	0	0	0	0	0	0	0	1	0	0
13	0.1650000	0.470417	0.3010000	0	0	0	0	0	0	0	0	0	0	0	0	1	0
14	0.1608700	0.537826	0.1265480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0.2333330	0.498750	0.1579630	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0.2316670	0.483750	0.1884330	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0.1758330	0.537500	0.1940170	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0.2166670	0.861667	0.1467750	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0.2921740	0.741739	0.2083170	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0.2616670	0.538333	0.1959040	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0.1775000	0.457083	0.3532420	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0.0591304	0.400000	0.1719700	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0.0965217	0.436522	0.2466000	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.2.1

The above data is showing to as that all the categorical variable which has more than two level attributes in it.

2.2.2 Model Selection

In this case we have to predict the count of bike renting according to environmental and seasonal condition. So the target variable here is a continuous variable. For Continuous we can use various Regression models. Model having less error rate and more accuracy will be our final model.

Models built are

1. c50 (Decision tree for regression target variable)
2. Random Forest (with 160 trees)
3. Linear regression

For each model we have divided the dataset into train and test part
Where train contains 80% data of data set and test contains 20% data to test the accuracy and error rate for a particular model.

2.2.2 Classification (C50)

This model is also known a Decision tree for regression target variable.
For this model we have divided the dataset into train and test part using random sampling. Where train contains 80% data of data set and test contains 20% data Which contains 64 variable where 64th variable is the target variable.

Error Rate of model is 20 % (MAPE *100)
Accuracy of model is 80 %

2.2.3 Random Forest

In Random forest we have divided the dataset into train and test part using random sampling. For this model we have divided the dataset into train and test part using random sampling. Where train contains 80% data of data set and test contains 20% data Which contains 64 variable where 64th variable is the target variable.

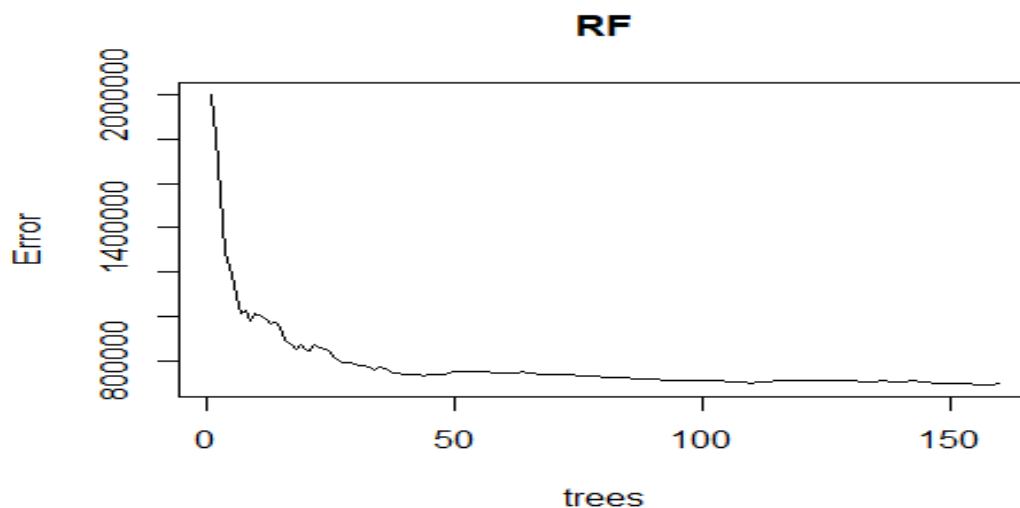


Figure 2.2.2

Above Figure 2.2.3 represents the curve of error rate as the number of trees increases. After 200 trees the error rate reaches to be constant. In this model we are using 200 trees to predict the target variable.

Error Rate of Random forest model is 14% (MAPE *100)
Accuracy of Random forest Model is 86%

2.2.4 Multiple linear Regression

For linear regression model we have divided the dataset into train and test part using random sampling. For this model we have divided the dataset into train and test part using random sampling. Where train contains 80% data of data set and test contains 20% data Which contains 64 variable where 64th variable is the target variable

Error rate of Linear Regression model is 17% (MAPE*100)
Accuracy of Linear Regression model is 83%
Bellow is the summary of the model.

Call:

```
lm(formula = cnt ~ ., data = train)
```

Residuals:

Min	1Q	Median	3Q	Max
-3983.6	-388.2	66.9	433.4	2451.7

Coefficients: (6 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2842.884	495.492	5.737	1.62e-08 ***
temp	4146.041	462.549	8.963	< 2e-16 ***
hum	-1285.597	330.635	-3.888	0.000114 **
windspeed	-2761.191	468.625	-5.892	6.81e-09 ***
dteday_01	-300.739	284.397	-1.057	0.290787
dteday_02	-210.045	274.563	-0.765	0.444604
dteday_03	-138.993	276.761	-0.502	0.615729
dteday_04	28.967	282.323	0.103	0.918319
dteday_05	-218.081	282.686	-0.771	0.440781
dteday_06	-50.656	269.798	-0.188	0.851142
dteday_07	-242.797	273.650	-0.887	0.375346
dteday_08	-151.395	284.678	-0.532	0.595080
dteday_09	-97.876	274.724	-0.356	0.721780
dteday_10	-198.936	280.134	-0.710	0.477930
dteday_11	-1.085	279.293	-0.004	0.996901
dteday_12	-135.587	273.873	-0.495	0.620756

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 759.9 on 526 degrees of freedom
Multiple R-squared: 0.8652, Adjusted R-squared: 0.8506
F-statistic: 59.25 on 57 and 526 DF, p-value: < 2.2e-16

Conclusion

3.1 Model Evaluation

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

There are several criteria that exist 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 of Bike rental Data, the latter two, Interpretability and Computation Efficiency, do not hold much

significance. Therefore we will use Predictive performance as the 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 measure.

3.1.1 Mean Absolute percentage error (MAPE)

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation. It usually expresses accuracy as a percentage, and is defined by the formula:

$$M = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|,$$

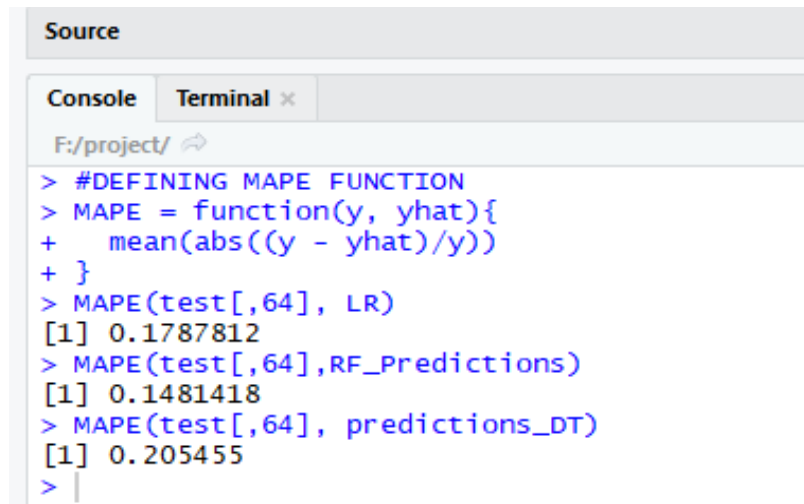
In R we have defined the MAPE Function as follows:

```
MAPE = function(y, yhat){  
  mean(abs((y - yhat)/y))*100  
}
```

In Python we have defined the MAPE Function as follows:

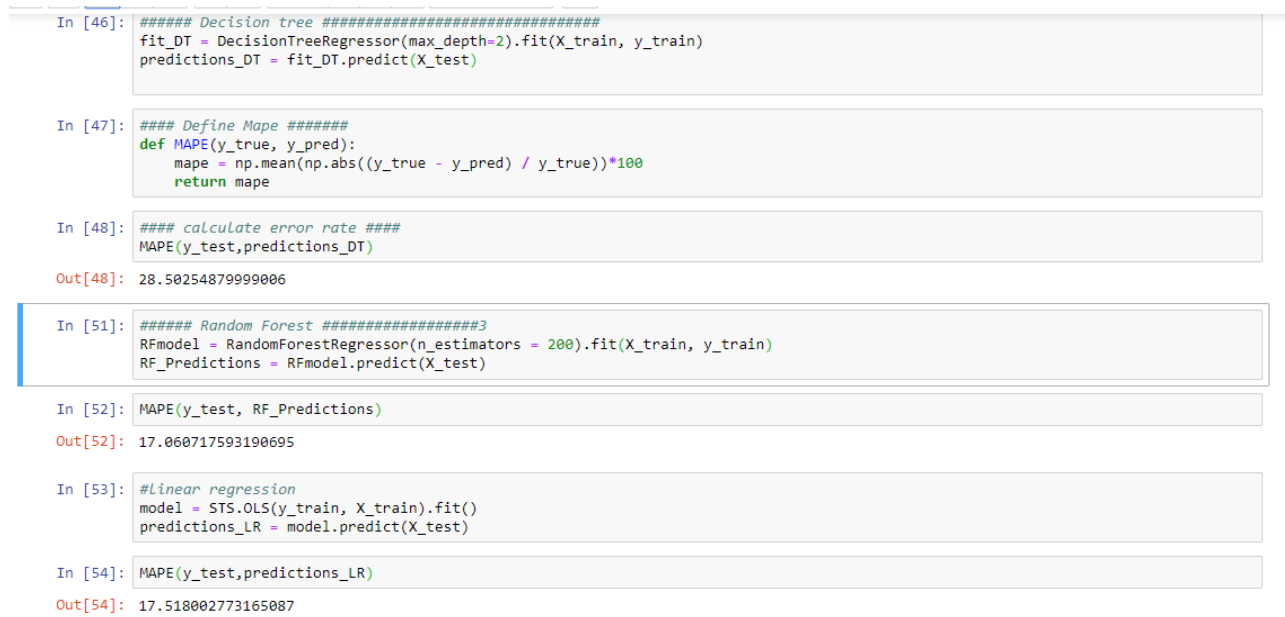
```
def MAPE(y_true, y_pred):  
    mape = np.mean(np.abs((y_true - y_pred) / y_true))*100  
    return mape
```

3.2 Model Selection



```
Source
Console Terminal x
F:/project/
> #DEFINING MAPE FUNCTION
> MAPE = function(y, yhat){
+   mean(abs((y - yhat)/y))
+ }
> MAPE(test[,64], LR)
[1] 0.1787812
> MAPE(test[,64],RF_Predictions)
[1] 0.1481418
> MAPE(test[,64], predictions_DT)
[1] 0.205455
> |
```

Figure 3.2.1



```
In [46]: ##### Decision tree #####
fit_DT = DecisionTreeRegressor(max_depth=2).fit(X_train, y_train)
predictions_DT = fit_DT.predict(X_test)

In [47]: ### Define Mape #####
def MAPE(y_true, y_pred):
    mape = np.mean(np.abs((y_true - y_pred) / y_true))*100
    return mape

In [48]: ### calculate error rate ###
MAPE(y_test,predictions_DT)

Out[48]: 28.50254879999006

In [51]: ##### Random Forest #####3
RFmodel = RandomForestRegressor(n_estimators = 200).fit(X_train, y_train)
RF_Predictions = RFmodel.predict(X_test)

In [52]: MAPE(y_test, RF_Predictions)

Out[52]: 17.060717593190695

In [53]: #Linear regression
model = STS.OLS(y_train, X_train).fit()
predictions_LR = model.predict(X_test)

In [54]: MAPE(y_test,predictions_LR)

Out[54]: 17.518002773165087
```

Figure 3.2.2

From the above figure we can see that Decision tree(C50) model has the most error rate hence less accuracy so we will not consider the C50 model

On the other hand random forest model perform comparatively well in R so we are selecting Random forest for the R .

In Python both models perform comparatively on average and therefore we can select either of the two models without any loss of information.