



Bike Renting

Vijay Gonsalves
31/8/2019

Table of Contents

1. INTRODUCTION	3
1.1 Problem statement.....	3
1.2 Data	3
2. Methodology	4
2.1 Pre Processing.....	4
2.1.1 Exploratory data analysis.....	4
2.1.2 Missing value Analysis.....	5
2.1.3 Outlier Analysis	5
2.1.4 Feature Selection.....	6
2.1.5 Feature Scaling.....	7
2.2 Modeling	8
2.2.1 Model Selection	8
2.2.2 C50 (Decision Tree).....	8
2.2.3 Random Forest.....	9
2.2.4 Linear Regression.....	9
3. Conclusion	11
3.1 Model Evaluation.....	11
3.1.1 Mean Absolute Percentage Error (MAPE).....	11
3.2 Model selection.....	12
Appendix A--R Code.....	13
Appendix B--Python code.....	13
Appendix C--Source data.....	13

1. INTRODUCTION

1.1 Problem statement

A Bike rental is a business which gives bikes like normal cycles or e-cycles to customers on rent. Rent can be typically for a day but sometimes maybe for more than a day also. People pick up a bike from a designated docking spot in the city and leave it at any other designated docking spot in the same city. Time taken from picking up to dropping off is used to charge the customers accordingly.

Typical customers of this business are tourist wanting to explore the city on weekends and daily office workers who use it to travel to their office.

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

1.2 Data

Our task is predict the count of bike rented depending on various environmental and seasonal conditions.

Since we have to predict the count i.e. a number this becomes a regression problem.

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	workingday	weekday
1	1/1/2011	1	0	1	0	0	6
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	1	2
5	1/5/2011	1	0	1	0	1	3
6	1/6/2011	1	0	1	0	1	4
7	1/7/2011	1	0	1	0	1	5
8	1/8/2011	1	0	1	0	0	6

Table 1.2: Sample Data (Columns: 9-16)

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	0.089565	88	1518	1606
2	0.196522	0.208839	0.498696	0.168726	148	1362	1510
2	0.165	0.162254	0.535833	0.266804	68	891	959

Variables present in given dataset are instant, dteday, season, yr, mnth, holiday, workingday, weekday, 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

2. 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

Deleted instant variable as it is nothing but an index.


Removed 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 if there are any missing values present in the 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 `Bike_renting.isnull().sum()` is used to detect any missing value



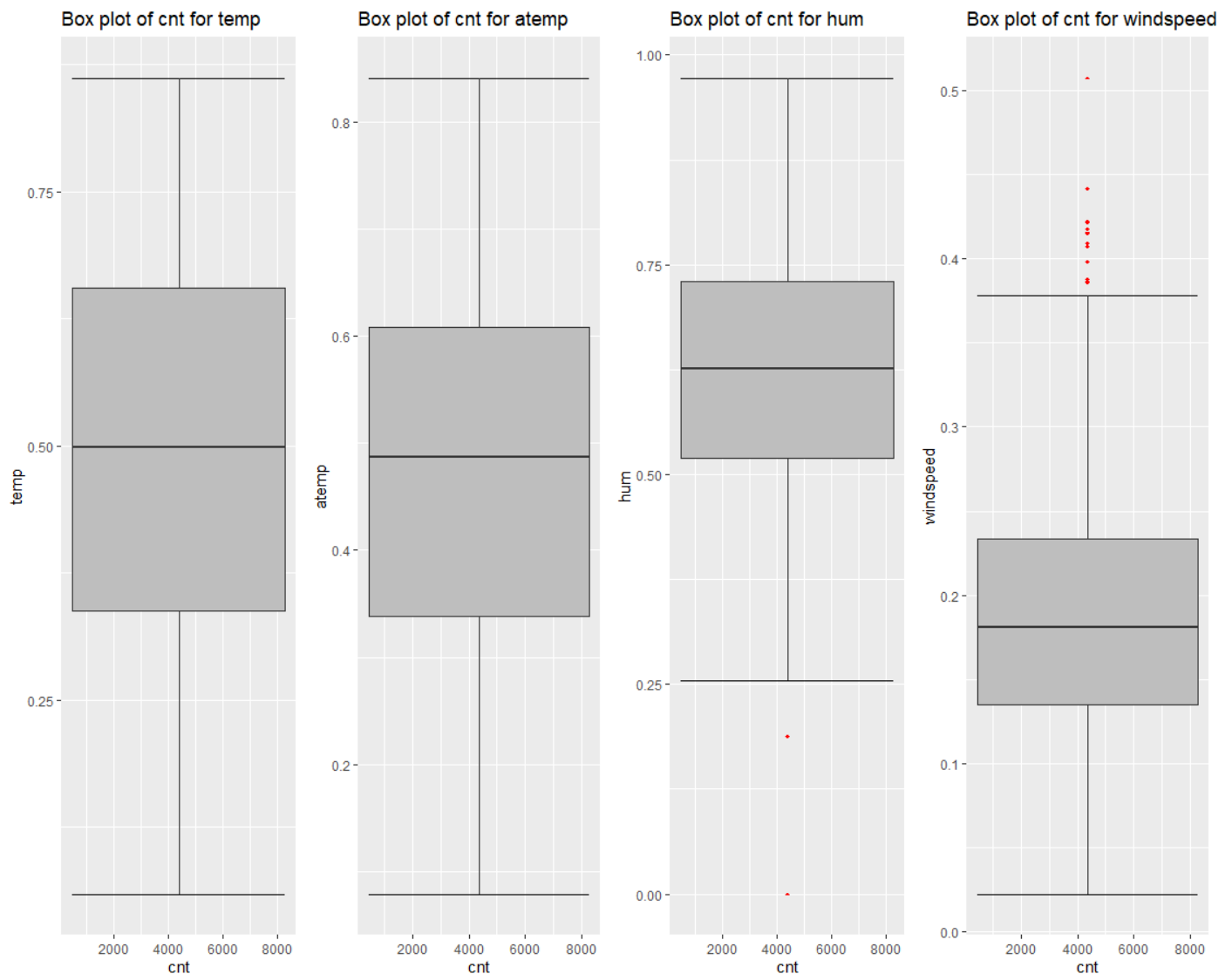
	apply.Bike_renting...2..function.x...
instant	0
dteday	0
season	0
yr	0
mnth	0
holiday	0
weekday	0
workingday	0
weathersit	0
temp	0
atemp	0
hum	0
windspeed	0
casual	0
registered	0
cnt	0

There is no missing value found in the given dataset.

2.1.3 Outlier Analysis

Outlier analysis is done to handle all inconsistent observations i.e. rows having values very far away from the mean and which distort the mean present in the given dataset. As outlier analysis can only be done on continuous variables.

Below figure is a visualization of numeric variables present in our dataset to detect outliers using a boxplot. Outliers will be detected with red color.



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.

We can either make the outlier values as null or then do Missing Value analysis again or we can delete the rows having outlier values.

As outliers are very small i.e. only 14/731 observations we delete them.

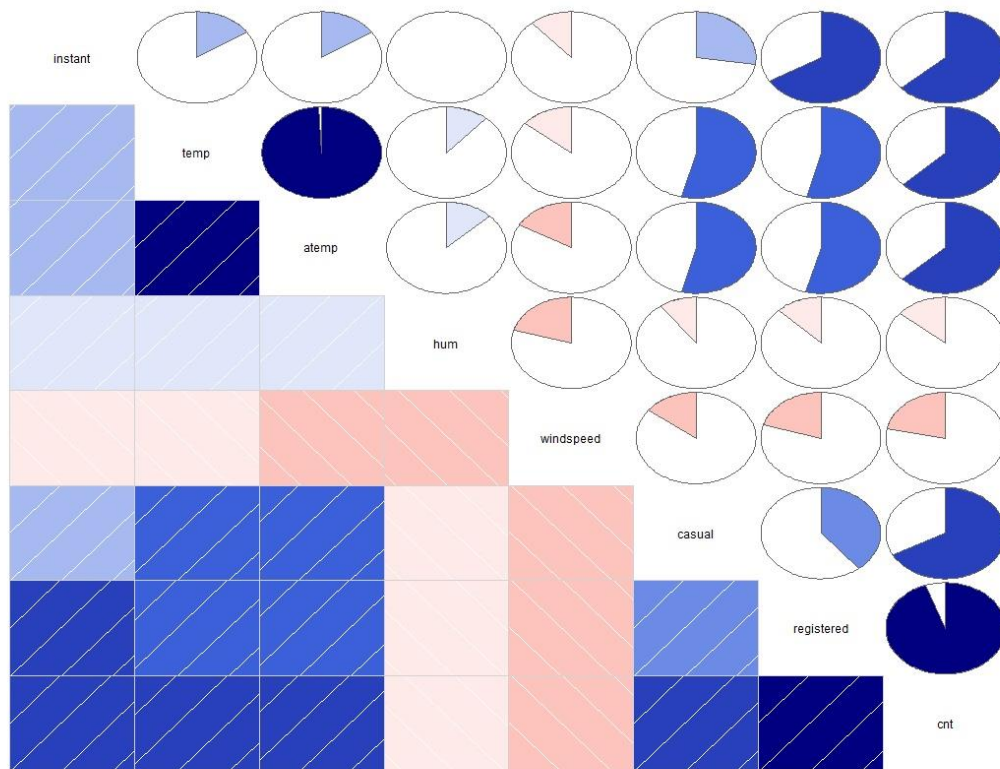
2.1.4 Feature Selection

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

We use correlation method for feature selection of continuous variables and chi-square test/anova test for feature selection of categorical variable.

Below Figure shows a correlation plot for all numeric variable present in dataset

Correlation plot



In above visualization we can see that only 2 variables are highly correlated with each other. Dark 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 we have removed atemp variable from dataset.

The above is supported by corr function which gives output as below:

```
> cor(corr_Bike_Renting)
      Bike_renting.cnt Bike_renting.temp Bike_renting.atemp Bike_renting.windspeed Bike_renting.hum
Bike_renting.cnt      1.0000000      0.6258917      0.6292045      -0.2161933      -0.1366214
Bike_renting.temp      0.6258917      1.0000000      0.9917378      -0.1401690      0.1141910
Bike_renting.atemp      0.6292045      0.9917378      1.0000000      -0.1660383      0.1265874
Bike_renting.windspeed -0.2161933      -0.1401690      -0.1660383      1.0000000      -0.2044964
Bike_renting.hum       -0.1366214      0.1141910      0.1265874      -0.2044964      1.0000000
>
```

Note that both temp and atemp have almost same value wrt to cnt variable.

From anova test using F-value statistic we exclude dteday and weekday variables.

2.1.5 Feature Scaling

Feature scaling includes two functions normalization and standardization. It is done reduce unwanted variation either within or between variables and to bring all of the variables into proportion with one another.

In given dataset all numeric values are already present in normalized form so no feature scaling required

2.2 Modeling

As can be seen from corr function below, both registered and casual tend to flow in the same way as cnt variable and as cnt is anyways sum of registered and casual, we drop registered and casual and model on cnt only.

```
> cor(corr_Bike_Renting)
      Bike_renting.cnt
Bike_renting.cnt      1.0000000
Bike_renting.registered 0.9445814
Bike_renting.casual     0.6705468
```

2.2.1 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

c50 (Decision tree for regression target variable)

Random Forest (with 900 trees) (Trees finalized after increasingly adding tress till accuracy no longer improves)

Linear regression

2.2.2 C50 Decision Tree

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 and contains 10 variable where 10th variable is the target variable.

Creating Model in R

```
## Training decision tree algo
fit = rpart(c(cnt) ~ ., data = train_cnt, method = "anova")
predictions_DT = predict(fit, test_cnt[, -c(10)])
MAPE = function(y, yhat) {
  mean(abs((y - yhat) / y)*100)
}
MAPE(test_cnt[, c(10)], predictions_DT)
```

In python

```
In [30]: #####c50#####
fit_DT = DecisionTreeRegressor(max_depth=2).fit(train_cnt.iloc[:,0:9], train_cnt.iloc[:,9])
predictions_DT = fit_DT.predict(test_cnt.iloc[:,0:9])
```


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 and contains 10 variable where 10th variable is the target variable.

In this model we are using 900 trees to predict the target variable.

Creating Model in Python

```
In [33]: RF_model = RandomForestRegressor(n_estimators = 500).fit(train_cnt.iloc[:,0:9], train_cnt.iloc[:,9])
RF_Predictions = RF_model.predict(test_cnt.iloc[:,0:9])
```

In R

```
RF_Model <- randomForest(cnt~.,train_cnt,ntree = 700, importance = TRUE)
varImpPlot(RF_Model)
imp <- importance(RF_Model)
RF_Predictions <- predict(RF_Model,test_cnt[, -c(10)])
MAPE(test_cnt[, c(10)], RF_Predictions)
plot(RF_Predictions)
# >only cnt >16.87 % error after increasing trees to 700
```

2.2.4 Linear Regression

In Linear regression we first check collinearity of numeric variables and only if there is no collinearity we go for model creation.

In Linear regression 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 and contains 10 variable where 10th variable is the target variable.

Creating Model in R

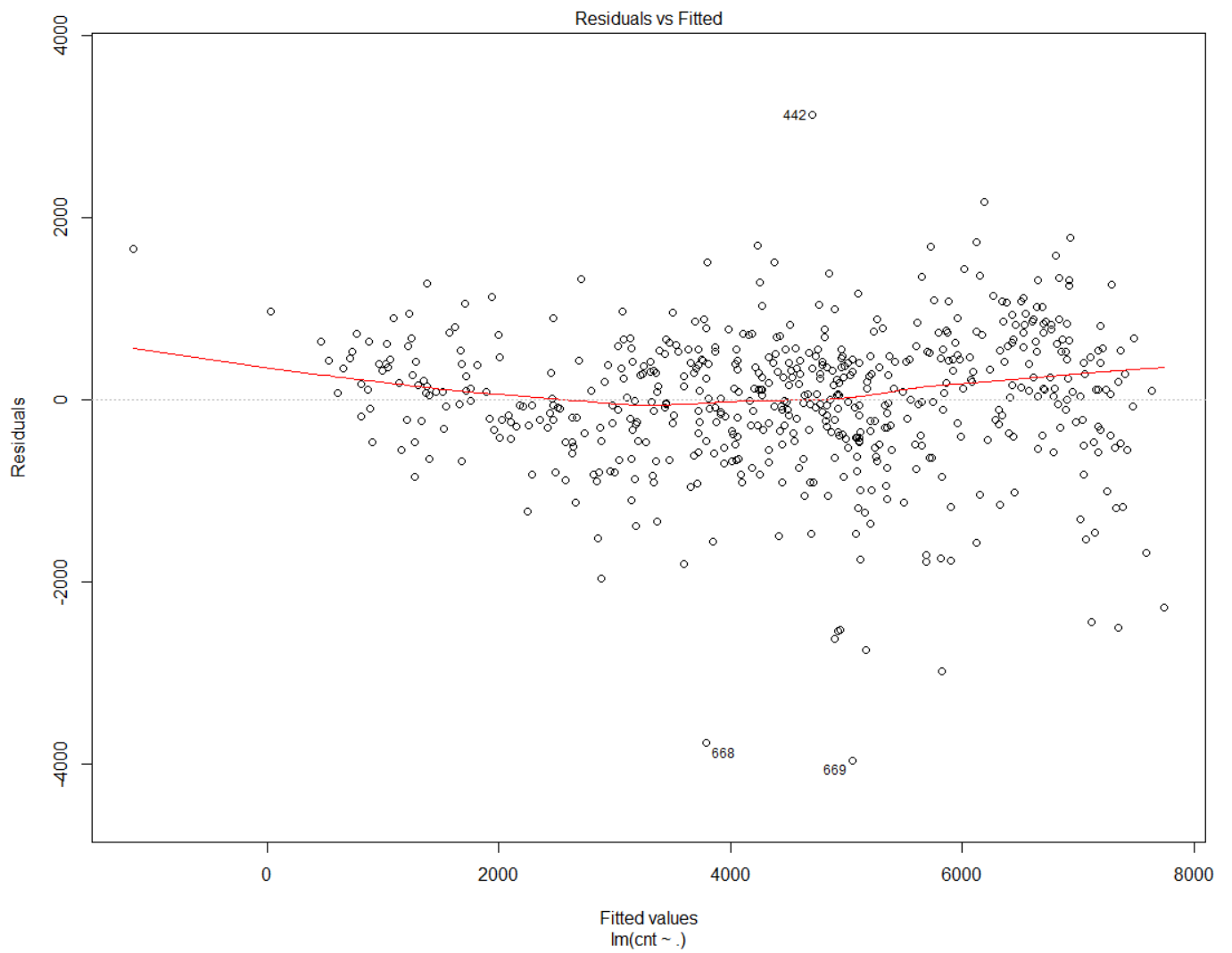
```
#-->using linear regression
vifcor(Bike_renting_cnt[,c(7:9)],th=0.9)
#-->No collinearity problem and hence we can go for linear regression
lm_model=lm(cnt~.,data=train_cnt)
summary(lm_model)
predictions_LR=predict(lm_model,test_cnt[, -c(10)])
MAPE(test_cnt[, c(10)], predictions_LR)
#-->only cnt-->16.78% error
#-----
```

In python

```
In [35]: ln_model=sm.OLS(train_cnt.iloc[:,9],train_cnt.iloc[:,0:9].astype(float)).fit()
```

```
In [36]: LN_Predictions = ln_model.predict(test_cnt.iloc[:,0:9])
```

```
In [37]: MAPE(test_cnt.iloc[:,9],LN_Predictions)
```



3. 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.. We can compare the models using any of the following criteria:

Predictive Performance

Interpretability

Computational Efficiency

In our case of Bike Renting, 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)

MAPE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous sections

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

In above function y is the actual value and yhat is the predicted value. It will provide the error percentage of model.

MAPE value in Python are as follow

```
In [37]: MAPE(test_cnt.iloc[:,9],LN_Predictions)
```

```
Out[37]: 18.861589196040615
```

```
In [38]: MAPE(test_cnt.iloc[:,9],RF_Predictions)
```

```
Out[38]: 15.219185603202245
```

```
In [39]: MAPE(test_cnt.iloc[:,9],predictions_DT)
```

```
Out[39]: 27.903661061469943
```

MAPE values in R are as follows:

```
C:/Users/vgonsalv/Desktop/datascience/project/
> MAPE(test_cnt[, c(10)], predictions_DT)
[1] 24.41227
> MAPE(test_cnt[, c(10)], RF_Predictions)
[1] 16.74375
> MAPE(test_cnt[, c(10)], predictions_LR)
[1] 16.78243
>
```

Where predictions_DT are predicted values from C50 model. RF_predictions are predicted values from random forest model and predictions_LR are predicted values from linear regression model

3.2 Model Selection

We can see that from both R and Python Random forest model performs best out of c50 and linear regression. So random forest model is selected with 84% accuracy in R and with 86% accuracy in python.

Extracted predicted value of random forest model are saved with .csv file format.

Appendix A—R code



Bike_renting_(R_Code
)_Vijay_Gonsalves.R

Appendix B—Python code



Bike_renting_(Python
_Code)_Vijay_Gonsalv

Appendix C—Source data



day.csv