Bike Sharing System - Analysis using R Data Mining

Prepared by:

Ajinkya Bhavik

Manasi Telang

Megha Chaudhry

Sree Narmadha Chandramohan

Under the Guidance of : Prof David Czerwinski

Abstract

For years, humans have been using the resources available on planet Earth, not only are we using the resources and extracting materials for our own personal use, but we are also generating waste twice the number of materials we are extracting. Technology today is generating massive amounts of data. Various real-life scenarios like Individuals connecting with each other, using devices which transfer information has led to data becoming the newest commodity for optimizing businesses and attracting consumers all around the world. It is very important to streamline this data and manage our resources in order to promote sustainability. This project aims to use data mining techniques to develop models which help analyze a bike sharing dataset and figure out the variables which affect and influence the usage of bike sharing service the most. These predictions are done based on the data provided by the two-year bike-sharing system from Capital Bikeshare system. For the analysis, the team used Linear regression, Regression Tree[CART], Cross Validation Fitting, and Random Forest as the major indicators of the data mining methodology.

Table of Contents

Introduction

The rising concerns of global climate change, pollution, health problems and motorization have led to citizens and companies searching for options for sustainable transportation like bike sharing systems. Bike sharing system is the new generation of traditional bike rentals where the whole process from membership, rental and return back becomes automatic " (Fanaee-T and Gamma, 2014). The increasing use of bike sharing systems has given us potential to analyze the data created by these systems. This analysis can be used to increase Bike Rental coverage to more areas thus helping us environmentally as well as economically.

The dataset belonging to a two-year bike-sharing system from the Capital Bikeshare system comprises of 17,379 observations with 17 variables like season, weather, temp, humidity, wind speed etc. The goal of this dataset is to analyze and figure out the variables which affect and influence the usage of a bike sharing service the most.

Data Cleaning and Preprocessing

The team worked on the initial stage of the project by loading and reading the csv file (hour.csv) in order to start reading and interpreting the dataset. The dataset belonging to a two-year bike-sharing system from the Capital Bikeshare system comprises 17,379 observations with 17 variables. After observing the summary, the team analyzed the structure of the data to understand different columns and their corresponding data types. The next steps involved data cleaning i.e. finding any missing or redundant values (NA). This was followed by data pre-processing wherein we converted data into its correct data types i.e. ‘char’ to ‘date’.

We also represented categorical data into their corresponding labels:

- Converting season data types from ‘int’ into ‘factors’ with 4 levels i.e. Spring, Summer, Fall, Winter
- Converting weather data types from ‘int’ into ‘factors’ with 4 levels i.e. Good, Normal, Bad, Very Bad

Data Exploration

Independent Variables:

The analysis of the data enabled the team to identify that it has one dependent variable, which is “cnt”, for which we would make the predictions.

Dependent Variables:

Now to understand the number of independent variables, we run the command dim (bike_Train). The output helped us identify 16 independent variables, many of which are predominantly categorical and continuous.

```
> dim(bike_Train)
[1] 8645   17
```

Below is a tabular summary of all the variables:

Name of Variables	Description	Type	Values
Instant	Instance of Record Created	Integer	Numeric
dteday	Date Stamp of each record	Date Format	Date Stamp from 2011 - 2012
season	Seasons within that duration	Factor w/4 levels	Spring, Fall, Winter, Summer
yr	Years within duration	Integer	Years from 2011 - 2012
month	Months within duration	Integer	Months from 2011 - 2012
hr	Hr records over that duration	Integer	Timestamp numeric Values
holiday	Holidays over the record	Integer	Holiday Numeric Values
weekday	Days Records when registered	Integer	Numeric
workingday	Working days used	Integer	Numeric
weathersit	Type of weather from good to bad	Factor w/4 levels	Good, Normal, Bad, Very Bad
temp	Range of Temperature within that period	Numeric	Numeric Range
atemp	Absolute Temperature	Numeric	Numeric Range
hum	Range of Humidity within that period	Numeric	Numeric Range
windspeed	Wind Speed during that time period	Numeric	Numeric Range
casual	Casual/ Occasionally used or enrolled	Integer	Numeric Range
registered	Registered over a longer period in that time	Integer	Numeric Range

Table 1: Data Exploration

Data Splitting

After the initial steps of Data Cleaning and preprocessing, the team split the dataset into training and testing sets. The team split the data into training, validation and testing sets as per the year in the dataset i.e. complete 2011 datapoints for training and 60% of 2012 data points for Validation. The remaining datapoints of 2012 were split into Testing since we wanted our predictions to be made for the consecutive dataset. We used the subset method for doing this.

```
> bike_Train= subset (Bike,yr==0)
> bike_Valid =
bike_TestandValid[sample(which(bike_TestandValid$yr==1),0.6*length(which(bike_TestandValid$yr==1))),]
> bike_Test = setdiff((bike_TestandValid),bike_Valid)
```

We applied different Data Mining methods to find the model which would fit best to our data giving us the lowest error.

Data Modelling

- Linear Regression

Linear regression is used to predict the value of an outcome variable *Y* based on one or more input predictor variables *X*. The aim is to establish a linear relationship (a mathematical formula) between the predictor variable(s) and the response variable, so that we can use this formula to estimate the value of the response *Y*, when only the predictors (*Xs*) values are known. Not every problem can be solved with the same algorithm. In this case, linear regression assumes that there exists a linear relationship between the response variable and the explanatory variables. This means that you can fit a line between the two (or more variables).

After completing data modelling, in order to work on the Linear Regression for hour.csv the team built a model without the year, date, casual and registered and instant. Since cnt variable includes both casual and registered and the dteday variable is not an independent variable but consists of variables that overlap with other variables like month, holiday, weekday.

For model lm1, the Multiple R-squared and Adjusted R-squared values received are .3969 and .3959 respectively. The weekday variable generated the highest p-value of .843007 thus it was removed for the next model lm2.

For model lm2, the Multiple R-squared and Adjusted R-squared values received are .3969 and .3959 respectively. The temp variable generated the highest p-value of .475678 thus it was removed for the next model lm3.

For model lm3, the Multiple R-squared and Adjusted R-squared values received are .3969 and .396 respectively. After removing weekdays, it was noted that the Adjusted R-squared value remains the same but there was a slight decrease in the Multiple R-squared value as the bias between them is comparatively smaller and suggests that the model has improved.

For model lm4, workingday variable was removed as it had a p-value as high as 0.23899. The Multiple R-squared and Adjusted R-squared values obtained in case of model lm4 are .3968 and .3959 respectively. It was found that the Adjusted R-squared value here stops increasing and starts to decrease slightly. The understood that removing working day does not improve the model and working day is insignificant here.

In the case of model lm5, the Multiple R-squared and Adjusted R-squared values received are .3966 and .3958 respectively. Since, the p-value in this case turns out to be less than 2.2 e- 16 which is less than the cutoff value, we reject the null hypothesis in favor of alternative hypothesis and conclude that the variables, month and season are dependent on each other. This implies that either month or season should be removed, and the team decided to remove season as one of its levels is not significant.

For model lm6, the Multiple R-squared and Adjusted R-squared values received are .3841 and .3835 respectively. Both the Multiple R-squared and the Adjusted R-squared stop increasing and start to reduce slightly (without any change in their difference), thus removing weathersit does not improve the model and the team decided to use model lm3 to predict the dataset to analyze its error levels.

Using model lm3 we first performed prediction on the validation set and calculated the respective errors. After that we performed prediction on the testing set and calculated the respective errors to check if the same trend has been followed as well.

The results derived using model lm3 are summarized in the table below:

Model Name	**MSE**	**RMSE**	**MAE**	**SSE**
Linear Regression on Validation Set	6869.37	82.88	128.64	188200200
Linear Regression on Testing Set	7021.97	83.80	132.38	131861330

Table 2: Summary of Results - Linear Regression

- Regression Tree [CART]

Recursive partitioning is a fundamental tool in data mining. It helps us explore the structure of a dataset, while developing easy to visualize decision rules for predicting a categorical (classification tree) or continuous (regression tree) outcomes. Basic regression trees partition a data set into smaller groups and then fit a simple model (constant) for each subgroup.

In order to grow our decision tree, we have to first load the rpart package. Then we can use the rpart() function, specifying the model formula, data, and method parameters. In this case, we wanted to classify the feature cnt using the predictor -instant-casual-registered-yr-dteday-weekday-atemp, so our call to rpart() should look like

BikeTree = rpart(cnt~.-instant-casual-registered-yr-dteday-weekday-atemp, data=bikeTrain)

The team then used the rpart.plot function to display the regression tree as a plot. This function combines and extends plot.rpart and text.rpart in the rpart package. It automatically scales and adjusts the displayed tree for best fit.

rpart.plot(BikeTree,type=3,digits = 3,fallen.leaves = TRUE)

Here type describes the type of plot to be displayed, digits gives the number of significant digits in displayed numbers and Fallen.leaves are numbers from 0.001 to 9999. By default, it is set to FALSE. If it is set to TRUE, it displays the leaves at the bottom of the graph.

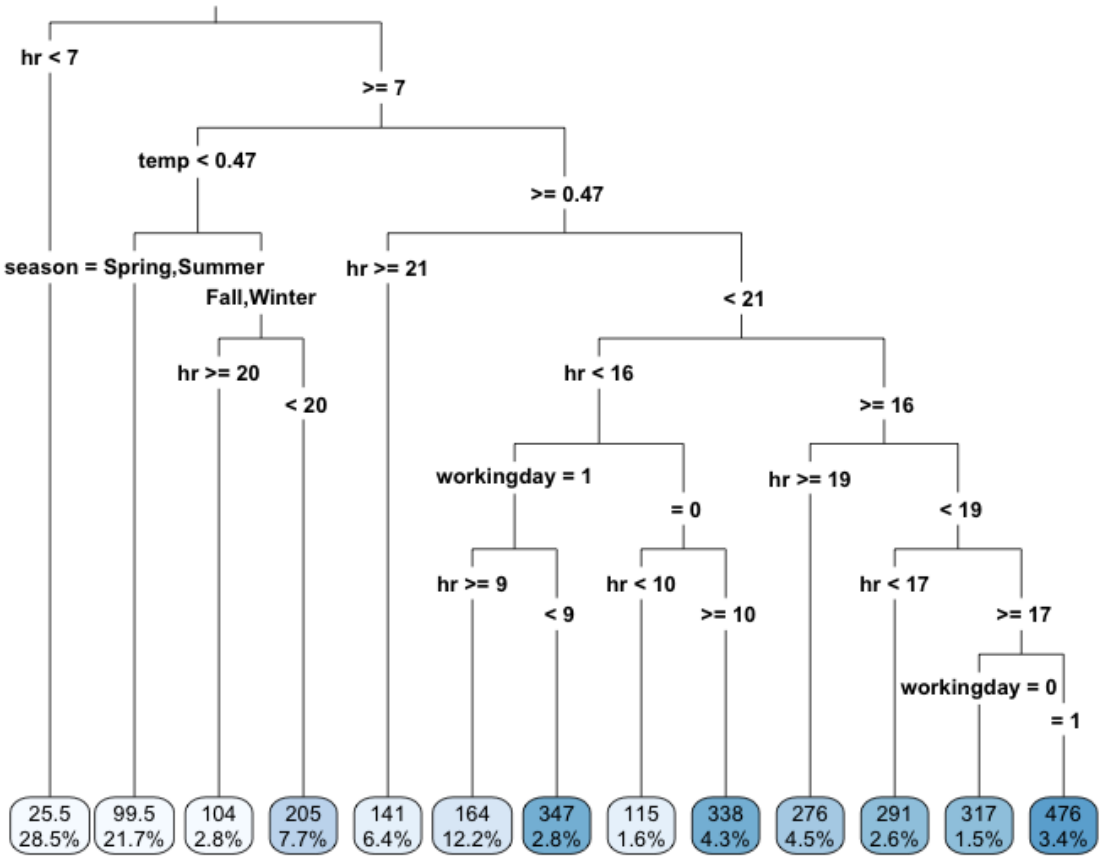


Figure 1: Regression tree plot

Next we used a tree to predict bike rentals for a given data. Predict function gives predictions from a fitted tree object. It Returns a vector of predicted responses from a fitted tree object. In our approach, the fitted tree object is BikeTree and we used test data as the second argument. The first argument is an object which is a fitted model object of class tree. The second argument is the data which in this case is the data frame containing the values at which predictions are required. Then we used the SSE to check the accuracy and store it in tree.sse. Next steps involved calculating the RMSE, MSE and MAE for the tree on the validation set as well as the testing set.

Model Name	MSE	RMSE	MAE	SSE
CART on Validation Testing	23031.16	151.76	106.23	120683268
CART on Testing Test	23334.85	152.76	106.07	81531959

Table 3: Summary of Results - Regression Tree

- Cross Validation Fitting

We used the cross-validation fitting method to further calculate if cross validation fit provides us with the better fit for the model. After applying this method, we also calculated the errors to get an estimate of the accuracy. The initial steps in this method involved installing the required packages for us to run a cross validation model. Those libraries are caret and e1071. The team also needed to modify the resampling method for this model; therefore, we used the function Train Control for defining train control of 10 folds.

> library(caret) - Library of Misc functions for training and plotting classification and regression mode

> library(e1071) - Library of Misc Functions of the Department of Statistics, Probability Theory Group

> train_control = trainControl(method = "cv", number = 10)

The team then changed the factor variables to numeric as we could not use categorical variables directly in our model before first processing them. We used the as.numeric function for this purpose.

We focused on the season and weather categorical variables and converted them. After that it was reinforced that they were in the correct format in the training set and the testing set followed by training our model on the train set.

We then assigned this to a variable for us to be able to plot a tree for it using the rpart.plot function.

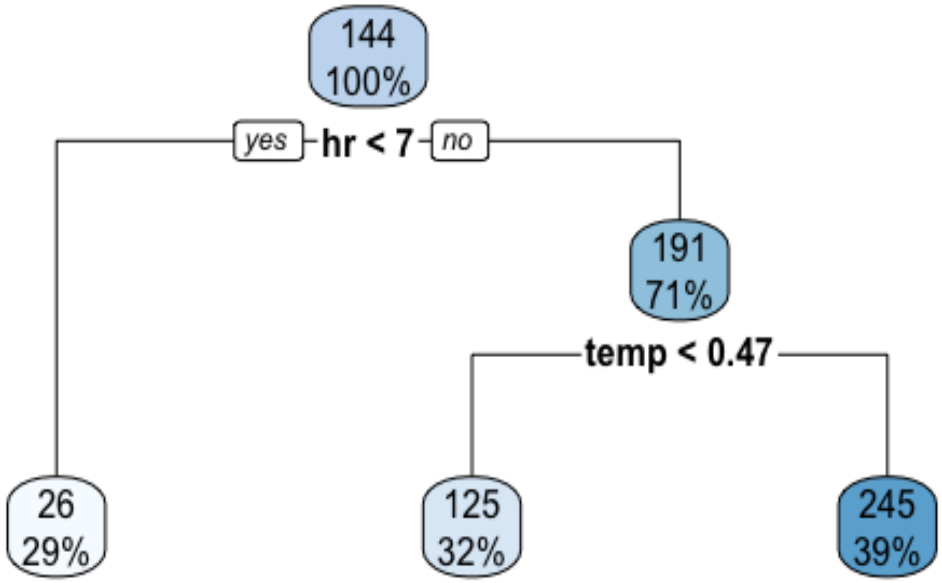


Figure 2: Cross-Validation Tree

The team then made predictions on the testing data in which we used the Tree 2 variable and used the testing data set as our data source. After the prediction using the testing set was complete, we calculated the different errors for the predictions on the validation set as well as testing set. Since having used CART before, we understood that the lower the RMSE, the better is the fit for the model.

Model Name	RMSE	MAE	MSE	SSE
Cross Validation on Validation Set	183.02	118.72	33495.94	175518707
Cross Validation on Testing Set	188.52	121.53	35539.77	124175962

Table 4: Summary of Results - Cross-Validation

- Random Forest

Random forests improve predictive accuracy by generating a large number of bootstrapped trees (based on random samples of variables), classifying a case using each tree in this new "forest", and deciding a final predicted

outcome by combining the results across all of the trees (an average in regression, a majority vote in classification).

We decided to proceed with the random forest algorithm, as it provides higher accuracy through the Cross-validation process. To proceed with the method, we installed the various packages like “dplyr”, “ggplot2”, “randomForest”. We converted the factor variables to numerical variables for both testing and training dataset. In order to generate the same numbers, we used the function “set.seed( )”. Later, fitted the random forest to the rental bike dataset by using the training dataset. The importance of the variables that were plotted for the rental bike training dataset was figured out from the function “varImpPlot( )”. We used the “which.min( )” function to eliminate the ties in the layers of the tree, which helped in getting a simpler and a clearer plot. Later we calculated the accuracies like RMSE, MAE, and MSE for the tree by using the predicted values from the rental bike validation as well as the testing dataset that we created.

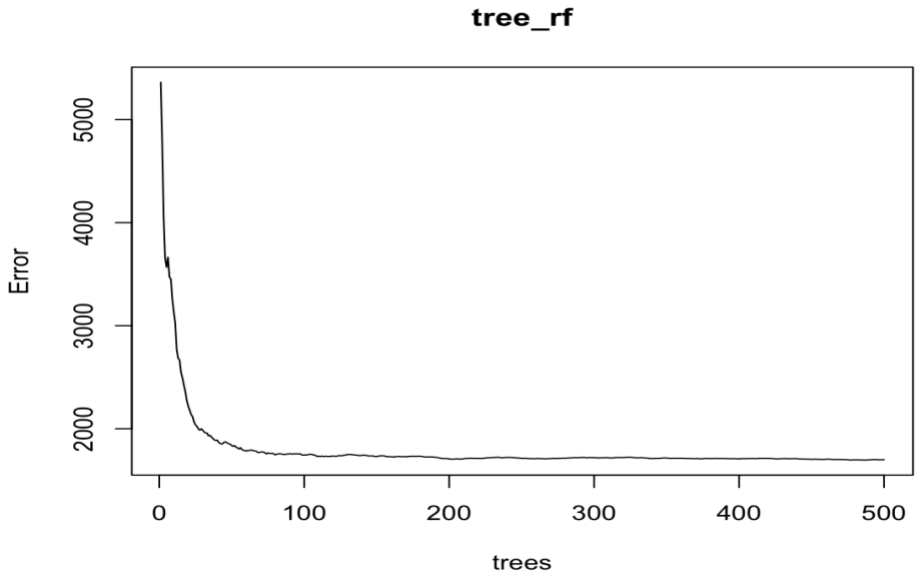


Figure 3: Random Forest Tree

Model Name	RMSE	MAE	MSE	SSE
Random Forest on Validation Set	135.87	91.82	18461.33	96737367
Random Forest on Testing Set	137.77	92.55	18981.39	66320971

Table 5: Summary of Results - Random Forest

Evaluation

Error Results for estimating better fit on Validation Set :

METHOD	RMSE	MAE	MSE	SSE
Linear Regression	82.88	128.64	6869.37	188200200
Regression Tree	151.76	106.23	23031.16	120683268
Cross Validation	183.02	118.72	33495.94	175518707
Random Forest	135.87	91.82	18461.33	96737367

Error results for estimating better fit on Testing Set :

METHOD	RMSE	MAE	MSE	SSE
Linear Regression	83.80	132.38	7021.97	131861330
Regression Tree	152.76	106.07	23334.85	81531959
Cross Validation	188.52	121.53	35539.77	124175962
Random Forest	137.77	92.55	18981.39	66320971

Table 6: Comparison on performance of the methods

On comparing the performance of all the 4 models (Linear Regression, CART, Cross Validation and Random Forest), the team noticed that the RMSE value is lowest for linear regression for both validation and the testing dataset. We would recommend the Linear Regression method since it has the least Root Mean Squared Value (i.e. 83.80) when compared to all other methods as we know the lower the RMSE value the better is the fit to the model. Also, the difference between all the calculated error values in the case of RMSE in both models is least in comparison to all the methods used.

MAE was also an interesting metric for us to consider since it is also used to express average model prediction error in units of the variable of interest and could be helpful if we benefit of penalizing larger errors was similar. But we considered RMSE for selecting the better method for modelling because in RMSE the errors are squared before they are averaged. This causes it to give a relatively high weight to large errors. Also, RMSE avoids the use of taking the absolute value. Therefore, RMSE is more useful in ours case since errors are particularly undesirable in our analysis

Business Insights

The following are the business insights which the team derived from its analysis:

1) Temperature and hour are the most significant factors in linear regression, cross validation, CART and random forest models, i.e. in all 4 methods. These parameters impact the total rental bike usages. This insight suggests that for the highest utilization and efficiency, the rental bike system should provide dynamic pricing, number of bike supplies and maintenance plan according to hour and temperature.
2) The Dataset tell us about the average bicycle counts for Summer, Fall, Winter and Spring. We notice that the average temperature during Summer, Fall, and Winter does not change the average demand trends of bicycles. However, since the temperature is not the only factor controlling the weather conditions, it seems that there are set of commoners factoring in a relatively constant bicycle demand during Summer, Fall, and Winter.
3) Upon looking at the correlation of various parameters in the dataset, we notice that the average bicycle demand during Summer, Fall, and Winter is higher compared to the demand during Spring.
4) Upon analyzing the hour parameter, we garner the insight that demand for bike sharing services is highest (peaks) during rush hours. Bicycle counts are also noticed to be cyclical through the day and through the year.

References

1. Chao, L. W., (Janell). (2018, April 29). Bike Sharing. Retrieved from https://www.kaggle.com/janellchao/bike-sharing
2. Kabacoff, R. I., Ph.D. (n.d.). Tree-Based Models. Retrieved from https://www.statmethods.net/advstats/cart.html
3. Kathuria, C. (2019, December 04). Regression — Why Mean Square Error? Retrieved from https://towardsdatascience.com/https-medium-com-chayankathuria-regression-why-mean-square-error-a8cad2a1c96f
4. Le, J. (2018, June 19). Decision Trees in R. Retrieved from https://www.datacamp.com/community/tutorials/linear-regression-R?utm_source=adwords_ppc&utm_campaignid=1565261270&utm_adgroupid=67750485268&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332661264374&utm_targetid=dsa-429603003980&utm_loc_interest_ms=&utm_loc_physical_ms=9032168&gclid=Cj0KCQjwt4X8BRCPARIsABmcnOoOPeP0qx6GWiFz1_yzpZv3V2WdgMAn_cdLZh1UaRTiN9UbuijC9KAaAtH5EALw_wcB
5. Czerwinski, D. (2020, September 01). *Modules*. Retrieved from https://sjsu.instructure.com/courses/1372259/modules

Appendix

Data Cleaning and Preprocessing

```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
> ## To load the dataset
> Bike <- read.csv("hour.csv")
>
> ##View the dataset
> Bike
   instant     dteday season yr mnth hr holiday weekday workingday weathersit temp  atemp  hum windspeed casual
1        1 2011-01-01      1  0    1  0       0       6          0          1 0.24 0.2879 0.81    0.0000      3
2        2 2011-01-01      1  0    1  1       0       6          0          1 0.22 0.2727 0.80    0.0000      8
3        3 2011-01-01      1  0    1  2       0       6          0          1 0.22 0.2727 0.80    0.0000      5
4        4 2011-01-01      1  0    1  3       0       6          0          1 0.24 0.2879 0.75    0.0000      3
5        5 2011-01-01      1  0    1  4       0       6          0          1 0.24 0.2879 0.75    0.0000      0
6        6 2011-01-01      1  0    1  5       0       6          0          2 0.24 0.2576 0.75    0.0896      0
7        7 2011-01-01      1  0    1  6       0       6          0          1 0.22 0.2727 0.80    0.0000      2
8        8 2011-01-01      1  0    1  7       0       6          0          1 0.20 0.2576 0.86    0.0000      1
9        9 2011-01-01      1  0    1  8       0       6          0          1 0.24 0.2879 0.75    0.0000      1
10      10 2011-01-01      1  0    1  9       0       6          0          1 0.32 0.3485 0.76    0.0000      8
11      11 2011-01-01      1  0    1 10       0       6          0          1 0.38 0.3939 0.76    0.2537     12
12      12 2011-01-01      1  0    1 11       0       6          0          1 0.36 0.3333 0.81    0.2836     26
13      13 2011-01-01      1  0    1 12       0       6          0          1 0.42 0.4242 0.77    0.2836     29
14      14 2011-01-01      1  0    1 13       0       6          0          2 0.46 0.4545 0.72    0.2985     47
15      15 2011-01-01      1  0    1 14       0       6          0          2 0.46 0.4545 0.72    0.2836     35
16      16 2011-01-01      1  0    1 15       0       6          0          2 0.44 0.4394 0.77    0.2985     40
17      17 2011-01-01      1  0    1 16       0       6          0          2 0.42 0.4242 0.82    0.2985     41
18      18 2011-01-01      1  0    1 17       0       6          0          2 0.44 0.4394 0.82    0.2836     15
19      19 2011-01-01      1  0    1 18       0       6          0          3 0.42 0.4242 0.88    0.2537      9
20      20 2011-01-01      1  0    1 19       0       6          0          3 0.42 0.4242 0.88    0.2537      6
21      21 2011-01-01      1  0    1 20       0       6          0          2 0.40 0.4091 0.87    0.2537     11
22      22 2011-01-01      1  0    1 21       0       6          0          2 0.40 0.4091 0.87    0.1940      3
23      23 2011-01-01      1  0    1 22       0       6          0          2 0.40 0.4091 0.94    0.2239     11
24      24 2011-01-01      1  0    1 23       0       6          0          2 0.46 0.4545 0.88    0.2985     15
25      25 2011-01-02      1  0    1  0       0       0          0          2 0.46 0.4545 0.88    0.2985      4
26      26 2011-01-02      1  0    1  1       0       0          0          2 0.44 0.4394 0.94    0.2537      1
27      27 2011-01-02      1  0    1  2       0       0          0          2 0.42 0.4242 1.00    0.2836      1
28      28 2011-01-02      1  0    1  3       0       0          0          2 0.46 0.4545 0.94    0.1940      2
29      29 2011-01-02      1  0    1  4       0       0          0          2 0.46 0.4545 0.94    0.1940      2
30      30 2011-01-02      1  0    1  6       0       0          0          3 0.42 0.4242 0.77    0.2985      0
```

```
> #numerical summary of each of our variables
> summary(Bike)
    instant          dteday              season           yr              mnth             hr
 Min.   :    1   Length:17379       Min.   :1.000   Min.   :0.0000   Min.   : 1.000   Min.   : 0.00
 1st Qu.: 4346   Class :character   1st Qu.:2.000   1st Qu.:0.0000   1st Qu.: 4.000   1st Qu.: 6.00
 Median : 8690   Mode  :character   Median :3.000   Median :1.0000   Median : 7.000   Median :12.00
 Mean   : 8690                      Mean   :2.502   Mean   :0.5026   Mean   : 6.538   Mean   :11.55
 3rd Qu.:13034                      3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:10.000   3rd Qu.:18.00
 Max.   :17379                      Max.   :4.000   Max.   :1.0000   Max.   :12.000   Max.   :23.00
    holiday           weekday          workingday       weathersit         temp            atemp              hum
 Min.   :0.00000   Min.   :0.000   Min.   :0.0000   Min.   :1.000   Min.   :0.020   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.340   1st Qu.:0.3333   1st Qu.:0.4800
 Median :0.00000   Median :3.000   Median :1.0000   Median :1.000   Median :0.500   Median :0.4848   Median :0.6300
 Mean   :0.02877   Mean   :3.004   Mean   :0.6827   Mean   :1.425   Mean   :0.497   Mean   :0.4758   Mean   :0.6272
 3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:0.660   3rd Qu.:0.6212   3rd Qu.:0.7800
 Max.   :1.00000   Max.   :6.000   Max.   :1.0000   Max.   :4.000   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
   windspeed          casual         registered        cnt
 Min.   :0.0000   Min.   :  0.00   Min.   :  0.0   Min.   :  1.0
 1st Qu.:0.1045   1st Qu.:  4.00   1st Qu.: 34.0   1st Qu.: 40.0
 Median :0.1940   Median : 17.00   Median :115.0   Median :142.0
 Mean   :0.1901   Mean   : 35.68   Mean   :153.8   Mean   :189.5
 3rd Qu.:0.2537   3rd Qu.: 48.00   3rd Qu.:220.0   3rd Qu.:281.0
 Max.   :0.8507   Max.   :367.00   Max.   :886.0   Max.   :977.0
>
> #structure of the data
> str(Bike)
'data.frame':   17379 obs. of  17 variables:
 $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : chr  "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
 $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
 $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
 $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
 $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
 $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
> head(Bike)
  instant     dteday season yr mnth hr holiday weekday workingday weathersit temp  atemp  hum windspeed casual
1       1 2011-01-01      1  0    1  0       0       6          0          1 0.24 0.2879 0.81    0.0000      3
2       2 2011-01-01      1  0    1  1       0       6          0          1 0.22 0.2727 0.80    0.0000      8
3       3 2011-01-01      1  0    1  2       0       6          0          1 0.22 0.2727 0.80    0.0000      5
4       4 2011-01-01      1  0    1  3       0       6          0          1 0.24 0.2879 0.75    0.0000      3
5       5 2011-01-01      1  0    1  4       0       6          0          1 0.24 0.2879 0.75    0.0000      0
6       6 2011-01-01      1  0    1  5       0       6          0          2 0.24 0.2576 0.75    0.0896      0
  registered cnt
1         13  16
2         32  40
3         27  32
4         10  13
5          1   1
6          1   1
>
> #count of missing values(NA)
> sum(is.na(Bike))
[1] 0
>
> #data preparation
> #converting char to dates
> Bike$dteday=as.Date(Bike$dteday)
> class(Bike$dteday)
[1] "Date"
> View(Bike)
>
> #as.factor, convert season and weathersit to factor
> #converting int to factor w/ level
> Bike$season=factor(Bike$season, labels = c("Spring", "Summer", "Fall", "Winter"))
> table(Bike$season)

Spring Summer   Fall Winter
  4242   4409   4496   4232
>
> Bike$weathersit=factor(Bike$weather, labels = c("Good", "Normal", "Bad", "Very Bad"))
> table(Bike$weathersit)

    Good   Normal      Bad Very Bad
   11413     4544     1419        3
>
```

```
> #structure of the data
> str(Bike)
'data.frame':   17379 obs. of  17 variables:
 $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : Date, format: "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
 $ season    : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weathersit: Factor w/ 4 levels "Good","Normal",..: 1 1 1 1 1 2 1 1 1 1 ...
 $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
 $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
 $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
 $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
> #Modelling - Splitting data before analysis
> #Creating training set for 2011
> bike_Train= subset(Bike,yr==0)
> #Combined Testing and Validation
> bike_TestandValid= subset(Bike,yr==1)
> #Creating validation set of year 2012
> bike_Valid = bike_TestandValid[sample(which(bike_TestandValid$yr==1),0.6*length(which(bike_TestandValid$yr==1))),]
> #Creating testing set of year 2012
> bike_Test = setdiff((bike_TestandValid),bike_Valid)
>
> dim(bike_Train)
[1] 8645   17
>
```

Linear Regression

Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
> #Using Linear regression
>
> #1st model - cnt variable includes both casual and registered
> #dteday is a dependent variable but consists of other overlapping variable
> #Hence we dont consider those variables for our model
> model_lm1 = lm(cnt~.-instant-casual-registered-yr-dteday, data=bike_Train)
> summary(model_lm1)

Call:
lm(formula = cnt ~ . - instant - casual - registered - yr - dteday,
    data = bike_Train)

Residuals:
    Min      1Q  Median      3Q     Max
-242.32  -70.07  -21.24   45.41  415.27

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          4.7830     7.3378   0.652 0.514530
seasonSummer        26.6612     4.1336   6.450 1.18e-10 ***
seasonFall           4.3804     5.6675   0.773 0.439605
seasonWinter        46.4892     5.6379   8.246  < 2e-16 ***
mnth                 1.8526     0.6087   3.043 0.002346 **
hr                   5.9178     0.1711  34.589  < 2e-16 ***
holiday            -21.1547     7.1020  -2.979 0.002903 **
weekday              0.1115     0.5630   0.198 0.843007
workingday          -2.9120     2.4941  -1.168 0.243028
weathersitNormal     9.6125     2.7427   3.505 0.000459 ***
weathersitBad      -20.9850     4.4090  -4.760 1.97e-06 ***
weathersitVery Bad  10.7615   104.1105   0.103 0.917674
temp               235.3742    49.2056   4.783 1.75e-06 ***
atemp               38.6059    53.8767   0.717 0.473666
hum               -149.4807     7.0470 -21.212  < 2e-16 ***
windspeed           18.6491    10.1154   1.844 0.065269 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 104 on 8629 degrees of freedom
Multiple R-squared:  0.3969,    Adjusted R-squared:  0.3959
F-statistic: 378.6 on 15 and 8629 DF,  p-value: < 2.2e-16
```

Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
> #2nd Model - many insignificant variables
> #so removing the weekday variables as the p value is the highest, 0.843007
> model_lm2 = lm(cnt~.-instant-casual-registered-yr-dteday-weekday, data=bike_Train)
> summary(model_lm2)

Call:
lm(formula = cnt ~ . - instant - casual - registered - yr - dteday -
    weekday, data = bike_Train)

Residuals:
    Min      1Q  Median      3Q     Max
-242.54  -70.15  -21.30   45.40  415.50

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.1961     7.0347   0.739 0.460146
seasonSummer        26.6771     4.1326   6.455 1.14e-10 ***
seasonFall           4.3972     5.6665   0.776 0.437769
seasonWinter        46.4566     5.6351   8.244  < 2e-16 ***
mnth                 1.8599     0.6076   3.061 0.002212 **
hr                   5.9173     0.1711  34.592  < 2e-16 ***
holiday            -21.2676     7.0787  -3.004 0.002668 **
workingday          -2.9125     2.4940  -1.168 0.242915
weathersitNormal     9.6318     2.7408   3.514 0.000443 ***
weathersitBad      -20.9416     4.4033  -4.756 2.01e-06 ***
weathersitVery Bad  10.7977   104.1045   0.104 0.917394
temp               235.4220    49.2023   4.785 1.74e-06 ***
atemp               38.4217    53.8657   0.713 0.475687
hum               -149.5859     7.0266 -21.289  < 2e-16 ***
windspeed           18.6829    10.1134   1.847 0.064730 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 104 on 8630 degrees of freedom
Multiple R-squared:  0.3969,    Adjusted R-squared:  0.3959
F-statistic: 405.7 on 14 and 8630 DF,  p-value: < 2.2e-16
```

Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
> #3rd Model - remove the atemp variables as the p value is the high, 0.475687
> model_lm3 = lm(cnt~.-instant-casual-registered-yr-dteday-weekday-atemp, data=bike_Train)
> summary(model_lm3)

Call:
lm(formula = cnt ~ . - instant - casual - registered - yr - dteday -
    weekday - atemp, data = bike_Train)

Residuals:
    Min      1Q  Median      3Q     Max
-241.83  -70.16  -21.22   45.42  415.93

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          6.5954     6.7555   0.976  0.32894
seasonSummer        26.6352     4.1320   6.446 1.21e-10 ***
seasonFall           4.0036     5.6394   0.710  0.47777
seasonWinter        46.3719     5.6337   8.231  < 2e-16 ***
mnth                 1.8833     0.6067   3.104  0.00191 **
hr                   5.9194     0.1710  34.611  < 2e-16 ***
holiday            -21.5959     7.0635  -3.057  0.00224 **
workingday          -2.9366     2.4937  -1.178  0.23899
weathersitNormal     9.6045     2.7405   3.505  0.00046 ***
weathersitBad      -21.1396     4.3945  -4.811 1.53e-06 ***
weathersitVery Bad   9.7763   104.0917   0.094  0.92517
temp               269.8540     9.5207  28.344  < 2e-16 ***
hum               -149.2977     7.0147 -21.283  < 2e-16 ***
windspeed           16.5938     9.6797   1.714  0.08651 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 104 on 8631 degrees of freedom
Multiple R-squared:  0.3969,    Adjusted R-squared:  0.396
F-statistic: 436.9 on 13 and 8631 DF,  p-value: < 2.2e-16
```

Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
> #After removing weekday, the Adjusted R-squared remains the same,
> #but the Multiple R-square reduces slightly.
> #bias between them becomes smaller. It means the model is improved.
> #remove the workingday variables as the p value is the high, 0.23899
> model_lm4 = lm(cnt~.-instant-casual-registered-yr-dteday-weekday-atemp-workingday, data=bike_Train)
> summary(model_lm4)

Call:
lm(formula = cnt ~ . - instant - casual - registered - yr - dteday -
    weekday - atemp - workingday, data = bike_Train)

Residuals:
    Min      1Q  Median      3Q     Max
-242.55  -70.43  -21.30   45.89  415.03

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          4.7649     6.5744   0.725 0.468612
seasonSummer        26.6750     4.1320   6.456 1.14e-10 ***
seasonFall           4.0746     5.6392   0.723 0.469984
seasonWinter        46.4024     5.6338   8.236  < 2e-16 ***
mnth                 1.8818     0.6067   3.102 0.001930 **
hr                   5.9207     0.1710  34.619  < 2e-16 ***
holiday            -19.5299     6.8424  -2.854 0.004324 **
weathersitNormal     9.4396     2.7370   3.449 0.000566 ***
weathersitBad      -21.4704     4.3856  -4.896 9.97e-07 ***
weathersitVery Bad   8.7137   104.0901   0.084 0.933287
temp               269.3734     9.5121  28.319  < 2e-16 ***
hum               -149.2343     7.0147 -21.275  < 2e-16 ***
windspeed           16.5453     9.6798   1.709 0.087440 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 104 on 8632 degrees of freedom
Multiple R-squared:  0.3968,    Adjusted R-squared:  0.3959
F-statistic: 473.1 on 12 and 8632 DF,  p-value: < 2.2e-16
```

Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
> #the Adjusted R-squared stops increasing and reduce slightly,
> #removing workingday does not improve the model. However, workingday is not significant.
> model_lm5 = lm(cnt~.-instant-casual-registered-yr-dteday-weekday-atemp-workingday-windspeed, data=bike_Train)
> summary(model_lm5)

Call:
lm(formula = cnt ~ . - instant - casual - registered - yr - dteday -
    weekday - atemp - workingday - windspeed, data = bike_Train)

Residuals:
    Min      1Q  Median      3Q     Max
-243.11  -70.73  -21.45   45.62  416.53

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)           9.6974     5.9078   1.641 0.100740
seasonSummer         26.6149     4.1323   6.441 1.25e-10 ***
seasonFall            3.5213     5.6306   0.625 0.531736
seasonWinter         46.0946     5.6315   8.185 3.11e-16 ***
mnth                  1.8501     0.6065   3.051 0.002291 **
hr                    5.9325     0.1709  34.712  < 2e-16 ***
holiday             -19.5735     6.8431  -2.860 0.004242 **
weathersitNormal      9.6830     2.7336   3.542 0.000399 ***
weathersitBad       -20.1288     4.3152  -4.665 3.14e-06 ***
weathersitVery Bad   11.7945   104.0861   0.113 0.909783
temp                270.1093     9.5034  28.422  < 2e-16 ***
hum                -152.3513     6.7742 -22.490  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 104 on 8633 degrees of freedom
Multiple R-squared:  0.3966,    Adjusted R-squared:  0.3958
F-statistic: 515.8 on 11 and 8633 DF,  p-value: < 2.2e-16
```

Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
> #using the Pearson's Chi-squared test to check correlation between month and seasons variables
> chisq.test(bike_Train$mnth, bike_Train$season, correct = FALSE)

        Pearson's Chi-squared test

data:  bike_Train$mnth and bike_Train$season
X-squared = 20829, df = 33, p-value < 2.2e-16

>
> #since the p-value less than 0.05(cutoff value),
> #the variables, mnth and season are dependent to each other.mnth or season should be removed.
> #Season removed as one of its level is not significant.
>
> model_lm6 = lm(cnt~.-instant-casual-registered-yr-dteday-weekday-atemp-workingday-windspeed-season, data=bike_Train)
> summary(model_lm6)

Call:
lm(formula = cnt ~ . - instant - casual - registered - yr - dteday -
    weekday - atemp - workingday - windspeed - season, data = bike_Train)

Residuals:
    Min      1Q  Median      3Q     Max
-236.33  -69.72  -20.61   45.01  418.02

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          11.7405     5.6501   2.078 0.037745 *
mnth                  5.0587     0.3509  14.415  < 2e-16 ***
hr                    6.0433     0.1712  35.293  < 2e-16 ***
holiday             -21.0242     6.9067  -3.044 0.002341 **
weathersitNormal     10.4011     2.7559   3.774 0.000162 ***
weathersitBad       -19.2952     4.3558  -4.430 9.55e-06 ***
weathersitVery Bad    3.1040   105.1294   0.030 0.976447
temp                250.1685     5.9903  41.762  < 2e-16 ***
hum                -145.8427     6.7158 -21.717  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 105.1 on 8636 degrees of freedom
Multiple R-squared:  0.3841,    Adjusted R-squared:  0.3835
F-statistic: 673.1 on 8 and 8636 DF,  p-value: < 2.2e-16
```

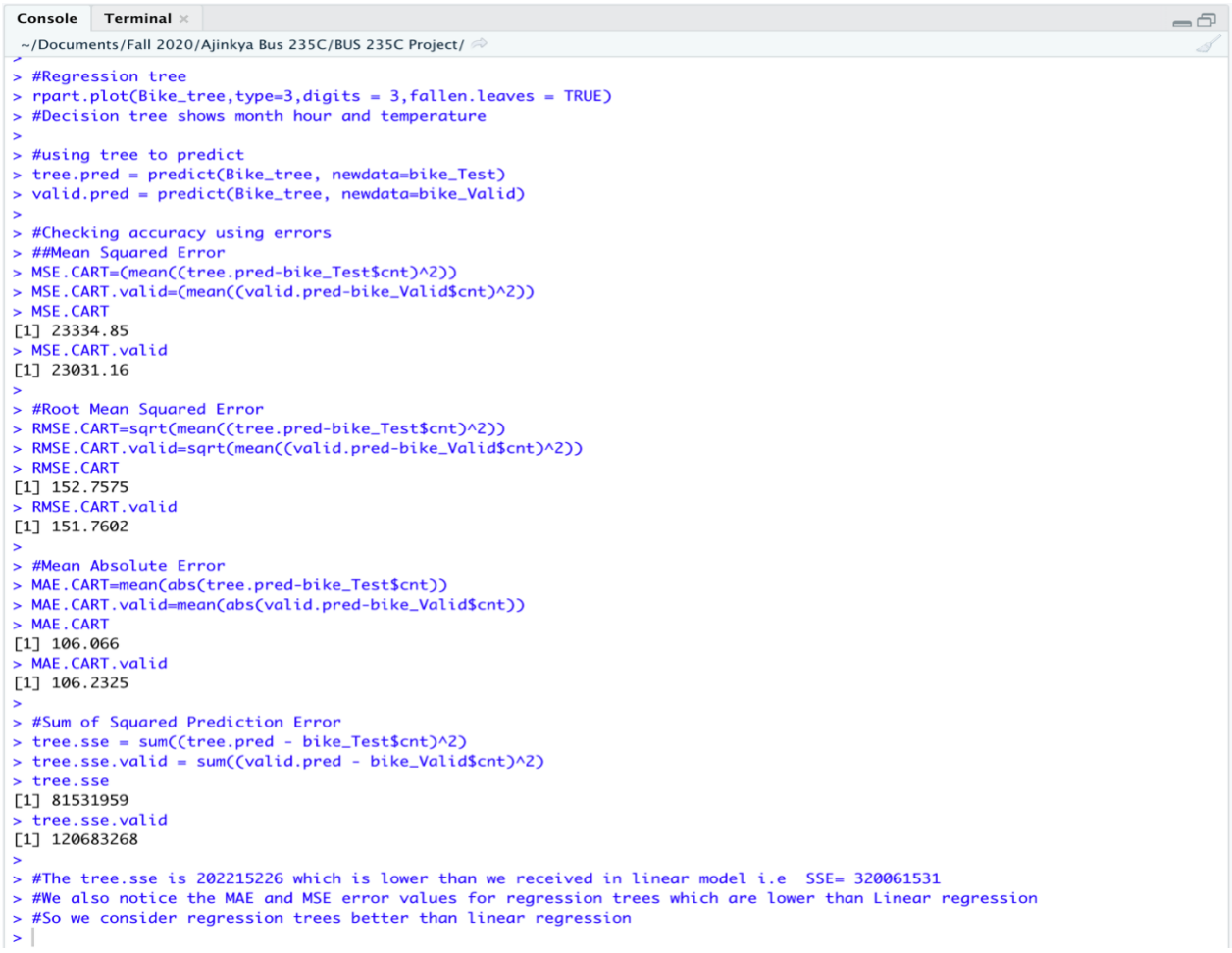
Console Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
>
> chisq.test(bike_Train$mnth, bike_Train$holiday, correct = FALSE)

        Pearson's Chi-squared test

data:  bike_Train$mnth and bike_Train$holiday
X-squared = 110.25, df = 11, p-value < 2.2e-16

> chisq.test(bike_Train$mnth, bike_Train$temperature, correct = FALSE)

        Chi-squared test for given probabilities

data:  bike_Train$mnth
X-squared = 15453, df = 8644, p-value < 2.2e-16


>
> #Multiple R-squared and the Adjusted R-squared stops increasing and reduce a little
> #Informing us that weathersit does not improve the model.
> #therefore we stick with the 3rd model after which the decrease started
>
> #Model3 - prediction check its error level
> test.pred.lin=predict(model_lm3,bike_Test)-1
> valid.pred.lin=predict(model_lm3,bike_Valid)-1
>
>
> #Assessing better fit using errors
> #Mean Squared Error
> MSE.lin.reg=mean(test.pred.lin-bike_Test$cnt)^2
> MSE.lin.reg.valid = mean(valid.pred.lin-bike_Valid$cnt)^2
> MSE.lin.reg
[1] 7021.973
> MSE.lin.reg.valid
[1] 6869.366
>
> #Root Mean Squared Error
> RMSE.lin.reg=sqrt(mean(test.pred.lin-bike_Test$cnt)^2)
> RMSE.lin.reg.valid=sqrt(mean(valid.pred.lin-bike_Valid$cnt)^2)
> RMSE.lin.reg
[1] 83.79721
> RMSE.lin.reg.valid
[1] 82.88164
>
> #Mean Absolute Error
> MAE.lin.reg=mean(abs(test.pred.lin-bike_Test$cnt))
> MAE.lin.reg.valid=mean(abs(valid.pred.lin-bike_Valid$cnt))
> MAE.lin.reg
[1] 132.3843
> MAE.lin.reg.valid
[1] 128.6425
>
> #Sum of Squared Prediction Error
> SSE.lin.reg = sum((test.pred.lin - bike_Test$cnt)^2)
> SSE.lin.reg.valid = sum((valid.pred.lin - bike_Valid$cnt)^2)
> SSE.lin.reg
[1] 131861330
> SSE.lin.reg.valid
[1] 188200200
```

Regression Tree

```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
> #Regression tree
> library(rpart)
> library(rpart.plot)
>
> #Prediction using Model 3
> #cnt -  function of month, hour, holiday, weather situation, temperature and humidity
> Bike_tree=rpart(cnt~.-instant-casual-registered-yr-dteday-weekday-atemp, data=bike_Train)
> Bike_tree
n= 8645

node), split, n, deviance, yval
      * denotes terminal node

  1) root 8645 154743700.0 143.79440
    2) hr< 6.5 2466   2294565.0  25.53731 *
    3) hr>=6.5 6179 104199500.0 190.99010
      6) temp< 0.47 2781  24189070.0 125.22110
       12) season=Spring,Summer 1877   9147025.0  99.54715 *
       13) season=Fall,Winter 904  11235920.0 178.52880
         26) hr>=19.5 239    576819.6 103.95820 *
         27) hr< 19.5 665   8852423.0 205.32930 *
      7) temp>=0.47 3398  58135890.0 244.81700
       14) hr>=20.5 552   1818016.0 141.22280 *
       15) hr< 20.5 2846  49244970.0 264.90970
         30) hr< 15.5 1805  22577430.0 220.83880
           60) workingday>=0.5 1293  11984570.0 198.50350
            120) hr>=8.5 1052   3142456.0 164.46290 *
            121) hr< 8.5 241   2301919.0 347.09540 *
           61) workingday< 0.5 512   8318870.0 277.24410
            122) hr< 9.5 140    600947.2 114.66430 *
            123) hr>=9.5 372   2624749.0 338.43010 *
         31) hr>=15.5 1041  17083130.0 341.32470
           62) hr>=18.5 391   2707172.0 276.18930 *
           63) hr< 18.5 650  11719220.0 380.50620
            126) hr< 16.5 227   1715203.0 290.85900 *
            127) hr>=16.5 423   7200702.0 428.61470
              254) workingday< 0.5 126   1164452.0 316.98410 *
              255) workingday>=0.5 297   3800000.0 475.97310 *
>
> #Regression tree
> rpart.plot(Bike_tree,type=3,digits = 3,fallen.leaves = TRUE)
> #Decision tree shows month hour and temperature
>
```



```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
>
> #Regression tree
> rpart.plot(Bike_tree,type=3,digits = 3,fallen.leaves = TRUE)
> #Decision tree shows month hour and temperature
>
> #using tree to predict
> tree.pred = predict(Bike_tree, newdata=bike_Test)
> valid.pred = predict(Bike_tree, newdata=bike_Valid)
>
> #Checking accuracy using errors
> ##Mean Squared Error
> MSE.CART=(mean((tree.pred-bike_Test$cnt)^2))
> MSE.CART.valid=(mean((valid.pred-bike_Valid$cnt)^2))
> MSE.CART
[1] 23334.85
> MSE.CART.valid
[1] 23031.16
>
> #Root Mean Squared Error
> RMSE.CART=sqrt(mean((tree.pred-bike_Test$cnt)^2))
> RMSE.CART.valid=sqrt(mean((valid.pred-bike_Valid$cnt)^2))
> RMSE.CART
[1] 152.7575
> RMSE.CART.valid
[1] 151.7602
>
> #Mean Absolute Error
> MAE.CART=mean(abs(tree.pred-bike_Test$cnt))
> MAE.CART.valid=mean(abs(valid.pred-bike_Valid$cnt))
> MAE.CART
[1] 106.066
> MAE.CART.valid
[1] 106.2325
>
> #Sum of Squared Prediction Error
> tree.sse = sum((tree.pred - bike_Test$cnt)^2)
> tree.sse.valid = sum((valid.pred - bike_Valid$cnt)^2)
> tree.sse
[1] 81531959
> tree.sse.valid
[1] 120683268
>
> #The tree.sse is 202215226 which is lower than we received in linear model i.e  SSE= 320061531
> #We also notice the MAE and MSE error values for regression trees which are lower than Linear regression
> #So we consider regression trees better than linear regression
>
```

Cross Validation

```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
> #Cross Validation
> library(caret)
> library(e1071)
>
> #define train control of 10 folds
> train_control = trainControl(method = "cv", number = 10)
>
> #cross validation fitting
>
> #changing factor variables to numerical variables
> Bike$season=as.numeric(Bike$season)
> bike_Train$season=as.numeric(bike_Train$season)
> bike_Test$season=as.numeric(bike_Test$season)
> bike_Valid$season=as.numeric(bike_Valid$season)
> Bike$weathersit=as.numeric(Bike$weathersit)
> bike_Train$weathersit=as.numeric(bike_Train$weathersit)
> bike_Test$weathersit=as.numeric(bike_Test$weathersit)
> bike_Valid$weathersit=as.numeric(bike_Valid$weathersit)
> str(bike_Train)
'data.frame':   8645 obs. of  17 variables:
 $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : Date, format: "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
 $ season    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weathersit: num  1 1 1 1 1 2 1 1 1 1 ...
 $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
 $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
 $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
 $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
> str(bike_Test)
'data.frame':   3494 obs. of  17 variables:
 $ instant   : int  8649 8650 8655 8661 8662 8669 8672 8674 8675 8677 ...
 $ dteday    : Date, format: "2012-01-01" "2012-01-01" "2012-01-01" "2012-01-01" ...
 $ season    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : int  3 4 9 15 16 23 2 5 6 8 ...
 $ holiday   : int  0 0 0 0 0 0 1 1 1 1 ...
```

```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
 $ weekday   : int  0 0 0 0 0 0 1 1 1 1 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weathersit: num  1 1 1 1 2 1 1 1 1 1 ...
 $ temp      : num  0.3 0.28 0.26 0.46 0.44 0.44 0.36 0.28 0.26 0.24 ...
 $ atemp     : num  0.333 0.303 0.273 0.455 0.439 ...
 $ hum       : num  0.81 0.81 0.93 0.51 0.54 0.51 0.34 0.45 0.41 0.35 ...
 $ windspeed : num  0 0.0896 0.1045 0.2985 0.2985 ...
 $ casual    : int  11 0 13 101 68 4 1 1 0 2 ...
 $ registered: int  41 8 27 164 147 25 6 3 14 51 ...
 $ cnt       : int  52 8 40 265 215 29 7 4 14 53 ...
> str(bike_Valid)
'data.frame':   5240 obs. of  17 variables:
 $ instant   : int  16256 13696 11219 13915 9385 12299 16644 10077 11392 12149 ...
 $ dteday    : Date, format: "2012-11-15" "2012-07-29" "2012-04-17" "2012-08-07" ...
 $ season    : num  4 3 2 3 1 2 4 1 2 2 ...
 $ yr        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ mnth      : int  11 7 4 8 1 6 12 2 4 5 ...
 $ hr        : int  1 20 15 23 22 15 6 22 20 9 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  4 0 2 2 2 5 6 3 2 6 ...
 $ workingday: int  1 0 1 1 1 1 0 1 1 0 ...
 $ weathersit: num  2 1 1 1 1 3 1 3 1 1 ...
 $ temp      : num  0.26 0.74 0.66 0.72 0.44 0.72 0.24 0.42 0.48 0.64 ...
 $ atemp     : num  0.273 0.697 0.621 0.697 0.439 ...
 $ hum       : num  0.65 0.66 0.24 0.79 0.35 0.66 0.87 0.94 0.21 0.83 ...
 $ windspeed : num  0.104 0.104 0.328 0 0.298 ...
 $ casual    : int  2 75 91 18 11 45 7 2 16 97 ...
 $ registered: int  14 228 217 120 118 213 20 53 269 192 ...
 $ cnt       : int  16 303 308 138 129 258 27 55 285 289 ...
> Bike_Tree.2=train(cnt~+season+mnth+hr+holiday+workingday+weathersit+temp+hum+windspeed,
+   data=bike_Train, trControl=train_control, method="rpart")
Warning message:
In nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,  :
  There were missing values in resampled performance measures.
> Bike_Tree.2
CART

8645 samples
   9 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7781, 7780, 7781, 7779, 7780, 7781, ...
Resampling results across tuning parameters:
```

Console | Terminal

~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/

```
RMSE was used to select the optimal model using the smallest value.
The final value used for the model was cp = 0.05382227.
>
> Tree2 = Bike_Tree.2$finalModel
>
> #plot
> rpart.plot(Tree2)
>
> #prediction using Tree2
> Tree2_pred = predict(Tree2, newdata=bike_Test)
> Tree2_pred.valid = predict(Tree2, newdata=bike_Valid)
>
> #Checking accuracy using errors
> #Sum of Squared Prediction Error
> Tree_sse2 = sum((Tree2_pred - bike_Test$cnt)^2)
> Tree_sse2.valid = sum((Tree2_pred.valid - bike_Valid$cnt)^2)
> Tree_sse2
[1] 124175962
> Tree_sse2.valid
[1] 175518707
>
> #Root Mean Squared Error
> RMSE_CV1=sqrt(mean((Tree2_pred-bike_Test$cnt)^2))
> RMSE_CV1.valid=sqrt(mean((Tree2_pred.valid-bike_Valid$cnt)^2))
> RMSE_CV1
[1] 188.5199
> RMSE_CV1.valid
[1] 183.019
>
> ##Mean Squared Error
> MSE_CV1=(mean((Tree2_pred-bike_Test$cnt)^2))
> MSE_CV1.valid=(mean((Tree2_pred.valid-bike_Valid$cnt)^2))
> MSE_CV1
[1] 35539.77
> MSE_CV1.valid
[1] 33495.94
>
> #Mean Absolute Error
> MAE_CV1=mean(abs(Tree2_pred-bike_Test$cnt))
> MAE_CV1.valid=mean(abs(Tree2_pred.valid-bike_Valid$cnt))
> MAE_CV1
[1] 121.5321
> MAE_CV1.valid
[1] 118.7164
```

Random Forest

```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
> #Random forest
> library(dplyr)
> library(ggplot2)
> library(randomForest)
>
> #changing factor variables to numerical variables
> str(bike_Train)
'data.frame':   8645 obs. of  17 variables:
 $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : Date, format: "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
 $ season    : num  1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weathersit: num  1 1 1 1 1 2 1 1 1 1 ...
 $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
 $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
 $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
 $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
> str(bike_Test)
'data.frame':   3494 obs. of  17 variables:
 $ instant   : int  8649 8650 8655 8661 8662 8669 8672 8674 8675 8677 ...
 $ dteday    : Date, format: "2012-01-01" "2012-01-01" "2012-01-01" "2012-01-01" ...
 $ season    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : int  3 4 9 15 16 23 2 5 6 8 ...
 $ holiday   : int  0 0 0 0 0 0 1 1 1 1 ...
 $ weekday   : int  0 0 0 0 0 0 1 1 1 1 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weathersit: num  1 1 1 1 2 1 1 1 1 1 ...
 $ temp      : num  0.3 0.28 0.26 0.46 0.44 0.44 0.36 0.28 0.26 0.24 ...
 $ atemp     : num  0.333 0.303 0.273 0.455 0.439 ...
 $ hum       : num  0.81 0.81 0.93 0.51 0.54 0.51 0.34 0.45 0.41 0.35 ...
 $ windspeed : num  0 0.0896 0.1045 0.2985 0.2985 ...
 $ casual    : int  11 0 13 101 68 4 1 1 0 2 ...
 $ registered: int  41 8 27 164 147 25 6 3 14 51 ...
 $ cnt       : int  52 8 40 265 215 29 7 4 14 53 ...
```

```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
> set.seed(123)
> tree_rf=randomForest(cnt~+season+mnth+hr+holiday+workingday+weathersit+temp+hum+windspeed, data=bike_Train,importance=TRUE,
ntree=500)
> tree_rf

Call:
 randomForest(formula = cnt ~ +season + mnth + hr + holiday +      workingday + weathersit + temp + hum + windspeed, data = b
ike_Train,      importance = TRUE, ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 1661.062
                    % Var explained: 90.72
>
> varImpPlot(tree_rf)
> which.min(tree_rf$mse)
[1] 481
> plot(tree_rf)
>
> ##Mean Squared Error
> Imp <- as.data.frame(sort(importance(tree_rf)[,1],decreasing = TRUE),optional = T)
> names(Imp)="%Inc MSE"
> Imp
            %Inc MSE
hr          231.25324
workingday 148.06211
temp         67.38285
hum          60.83853
weathersit   46.86328
mnth         46.43522
season       42.05867
windspeed    34.02570
holiday      29.01211
>
```

```
> #Predict
> predict_rf=predict(tree_rf,bike_Test)
> predict_rf_valid=predict(tree_rf,bike_Valid)
>
> #Checking accuracy using errors
>
> ##Root Mean Squared Error
> RMSE_forest=sqrt(mean((predict_rf-bike_Test$cnt)^2))
> RMSE_forest_valid=sqrt(mean((predict_rf_valid-bike_Valid$cnt)^2))
> RMSE_forest
[1] 137.773
> RMSE_forest_valid
[1] 135.8725
>
> ##Mean Squared Error
> MSE_forest=mean((predict_rf-bike_Test$cnt)^2)
> MSE_forest_valid=mean((predict_rf_valid-bike_Valid$cnt)^2)
> MSE_forest
[1] 18981.39
> MSE_forest_valid
[1] 18461.33
>
> #Mean Absolute Error
> MAE_forest=mean(abs(predict_rf- bike_Test$cnt))
> MAE_forest_valid=mean(abs(predict_rf_valid- bike_Valid$cnt))
> MAE_forest
[1] 92.54698
> MAE_forest_valid
[1] 91.82117
>
> #Sum of Squared Error
> tree.sserf = sum((predict_rf - bike_Test$cnt)^2)
> tree.sserf_valid = sum((predict_rf_valid - bike_Valid$cnt)^2)
> tree.sserf
[1] 66320971
> tree.sserf_valid
[1] 96737367
```

Evaluation

```
Console  Terminal
~/Documents/Fall 2020/Ajinkya Bus 235C/BUS 235C Project/
> #Evaluation for the methods performed
>
> Accuracy_test=data.frame(Method=c("Linear Regression","Regression Tree","Cross Validation","Random Forest"),
+                     RMSE=c(RMSE.lin.reg,RMSE.CART,RMSE_CV1,RMSE_forest),
+                     MAE=c(MAE.lin.reg,MAE.CART,MAE_CV1,MAE_forest),
+                     MSE=c(MSE.lin.reg,MSE.CART,MSE_CV1,MSE_forest),
+                     SSE=c(SSE.lin.reg,tree.sse,Tree_sse2,tree.sserf))
>
> Accuracy_valid=data.frame(Method=c("Linear Regression","Regression Tree","Cross Validation","Random Forest"),
+                     RMSE=c(RMSE.lin.reg.valid,RMSE.CART.valid,RMSE_CV1.valid,RMSE_forest_valid),
+                     MAE=c(MAE.lin.reg.valid,MAE.CART.valid,MAE_CV1.valid,MAE_forest_valid),
+                     MSE=c(MSE.lin.reg.valid,MSE.CART.valid,MSE_CV1.valid,MSE_forest_valid),
+                     SSE=c(SSE.lin.reg.valid,tree.sse.valid,Tree_sse2.valid,tree.sserf_valid))
>
> Accuracy_test
             Method      RMSE       MAE       MSE       SSE
1 Linear Regression  83.79721 132.38432  7021.973 131861330
2   Regression Tree 152.75748 106.06605 23334.848  81531959
3  Cross Validation 188.51995 121.53211 35539.772 124175962
4     Random Forest 137.77296  92.54698 18981.388  66320971
> Accuracy_valid
             Method      RMSE       MAE       MSE       SSE
1 Linear Regression  82.88164 128.64249  6869.366 188200200
2   Regression Tree 151.76020 106.23247 23031.158 120683268
3  Cross Validation 183.01895 118.71641 33495.936 175518707
4     Random Forest 135.87248  91.82117 18461.330  96737367
>
>
```

```
> Accuracy_valid$RMSE = round(Accuracy_valid$RMSE,2)
> Accuracy_valid$MAE = round(Accuracy_valid$MAE,2)
> Accuracy_valid$MSE = round(Accuracy_valid$MSE,2)
> Accuracy_valid$SSE = round(Accuracy_valid$SSE,2)
> Accuracy_valid
             Method    RMSE     MAE       MSE       SSE
1 Linear Regression  82.88 128.64  6869.37 188200200
2   Regression Tree 151.76 106.23 23031.16 120683268
3  Cross Validation 183.02 118.72 33495.94 175518707
4     Random Forest 135.87  91.82 18461.33  96737367
>
> Accuracy_test$RMSE = round(Accuracy_test$RMSE,2)
> Accuracy_test$MAE = round(Accuracy_test$MAE,2)
> Accuracy_test$MSE = round(Accuracy_test$MSE,2)
> Accuracy_test$SSE = round(Accuracy_test$SSE,2)
> Accuracy_test
             Method    RMSE     MAE       MSE       SSE
1 Linear Regression  83.80 132.38  7021.97 131861330
2   Regression Tree 152.76 106.07 23334.85  81531959
3  Cross Validation 188.52 121.53 35539.77 124175962
4     Random Forest 137.77  92.55 18981.39  66320971
>
```