**Bike Rental Count**

**SUBMITTED BY:**

**Shubham Gaikwad**

Shubhamgaikwad0322@gmail.com

**(May 2020)**

**Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page Num.** |
| **1** | **Introduction** | 3 |
|  | 1.1 Problem statement | 3 |
|  | 1.2 Data | 3 |
|  | 1.3 Software and hardware requirements | 5 |
|  |  |  |
| **2** | **Methodology** | 6 |
|  | 2.1 Pre-Processing | 6 |
|  | 2.1.1 Exploratory Data Analysis | 7 |
|  |  |  |
| **3** | **Missing Value Analysis** | 8 |
|  |  |  |
| **4** | **Outlier Analysis** | 9 |
|  |  |  |
| **5** | **Feature Selection and Scaling** | 12 |
|  | 5.1 Feature Selection | 12 |
|  | 5.2 Feature Scaling | 13 |
|  |  |  |
| **6** | **Modelling** | 14 |
|  | 6.1 Model Selection | 14 |
|  | 6.1.1 Decision Trees | 15 |
|  | 6.1.2 Random Forest Classifier | 15 |
|  | 6.1.3 Linear Regression | 16 |
|  |  |  |
| 7 | Conclusion | 18 |
|  |  |  |
|  |  |  |
|  |  |  |

**Chapter 1**

**Introduction**

1. **Problem Statement**

Choosing bike sharing system as a medium of transport will allow an eco-friendlier way of transportation. A bike rental is a bicycle business that rents bikes for short periods of time. Bike rental shops rent by the day or week as well as by the hour, and these provide an excellent opportunity for people like travelers and tourists, who don't have access to a vehicle.

Specialized bike rental shops thus typically operate at beaches, parks, or other locations that tourists frequently visit. 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 predict the bike rental count 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. **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:

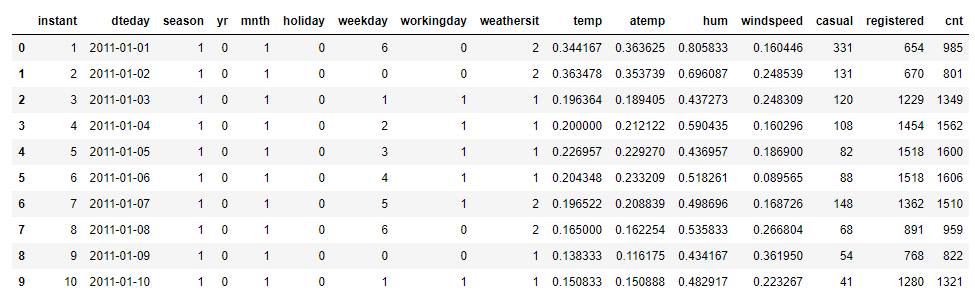


Table 1.1: Sample data (Column 1 to 8)

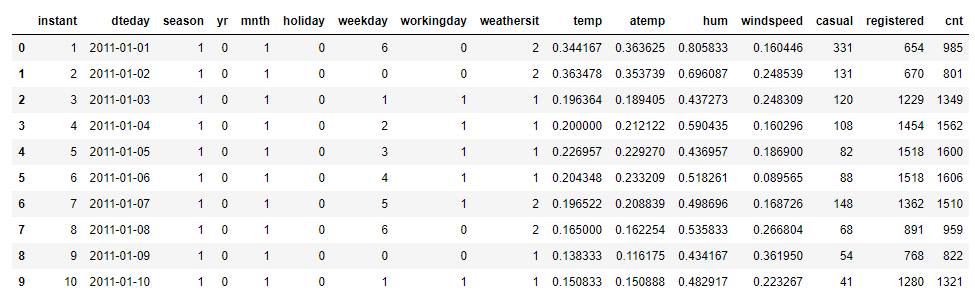


Table 1.2: Sample Data (Columns:8 to 16)

Variables present in given dataset are as follow:

|  |  |
| --- | --- |
| Sr.no | Column Name |
| 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 1.3: Customer default status Prediction variables

The details of variable present in the dataset are as follows:

1. instant: Record index
2. dteday: Date
3. season: Season (1: Springer, 2: Summer, 3: Fall, 4: Winter)
4. yr: Year (0: 2011, 1:2012)
5. mnth: Month (1 to 12)
6. hr: Hour (0 to 23)
7. holiday: weather day is holiday or not (extracted from Holiday Schedule)
8. weekday: Day of the week
9. workingday: If day is neither weekend nor holiday is 1, otherwise is 0.
10. weathersit: (extracted fromFreemeteo)
11. Clear, Few clouds, Partly cloudy, Partly cloudy
12. Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
13. Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
14. Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
15. 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)

1. atemp:

Normalized feeling temperature in Celsius. The values are derived via (t-t\_min)/(t\_maxt\_min), t\_min=-16, t\_max=+50 (only in hourly scale)

1. hum: Normalized humidity. The values are divided to 100 (max)
2. windspeed: Normalized wind speed. The values are divided to 67 (max)
3. casual: count of casual users
4. registered: count of registered users
5. cnt: count of total rental bikes including both casual and registered
6. **Software and Hardware Requirements:**
7. R 3.6.1 for 64 bit
8. Anaconda 3 for 64 bit
9. R studio
10. 64 bit OS
11. Python 3
12. Jupyter Notebook
13. 4GB of RAM

**Chapter 2**

**Methodology**

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. To start this process, we will first try and look at class imbalance of Target variable in most of the classification class imbalance will create severe problems during the modelling. This process is often call as exploratory data analysis.

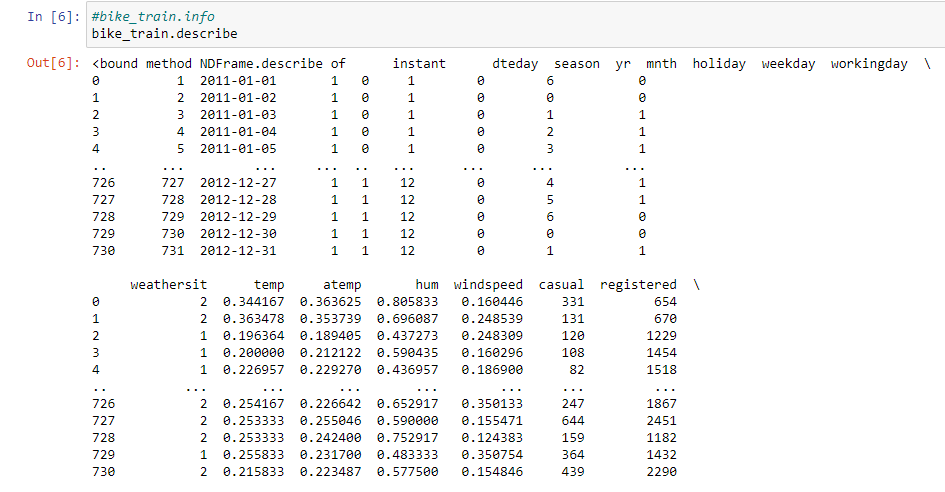


Table 2.1: Preprocessing (A)

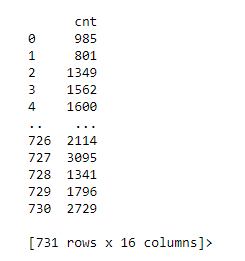


Table 2.2: Preprocessing (B)

**2.1.1 Exploratory Data Analysis:**

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it.

In exploring the data, we have,

Converted season, mnth, workingday, weathersit into categorical variables Feature Engineering Changed deday 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.

**Chapter 3**

**Missing Value Analysis:**

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models. 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.

Below table illustrate about the missing value in the dataset.

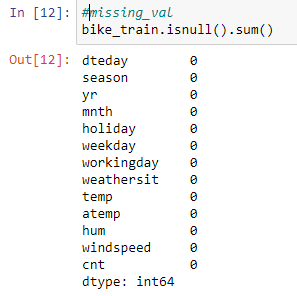


Table 3.1 Missing Values in bank loan data (In Python)



Figure 3.2 Missing Values in bank loan data (In RStudio)

In R function(x){sum(is.na(x))} is the function used to check the sum of missing values. In python df.isnull().sum() is used to detect any missing value.

There are no any missing values here in the dataset.

**Chapter 4**

**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.

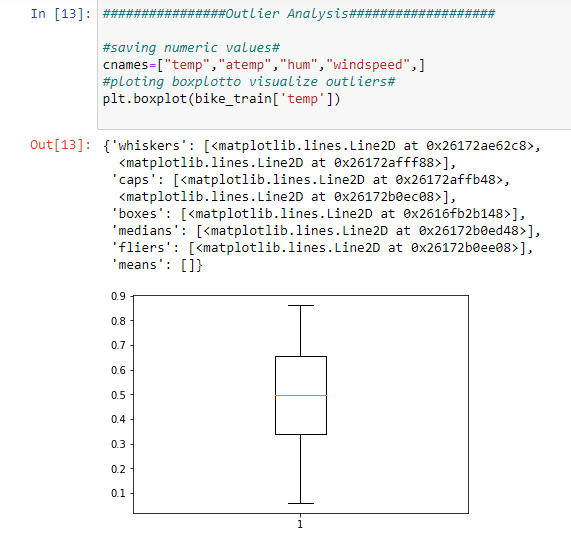
Figure 4.1 and 4.2 are visualization of numeric variable present in our dataset to detect outliers using boxplot and distribution plots. Outliers will be detected with black color.

Figure 4.1 Boxplot graph (In Python)

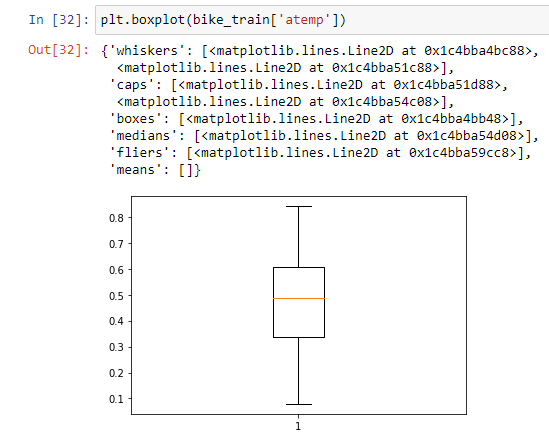


Figure 4.2 Boxplot graph of atemp variable (In Python)

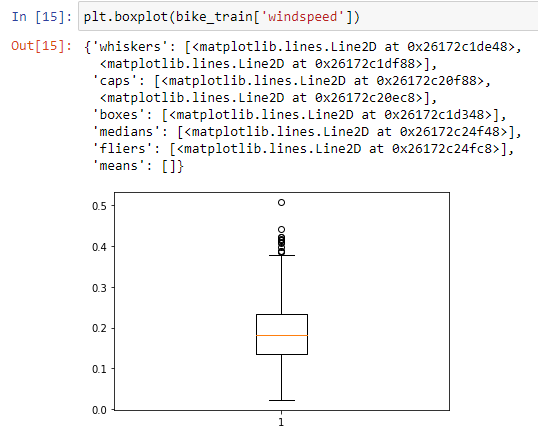


Figure 4.3 Boxplot graph of windspeed variable (In Python)

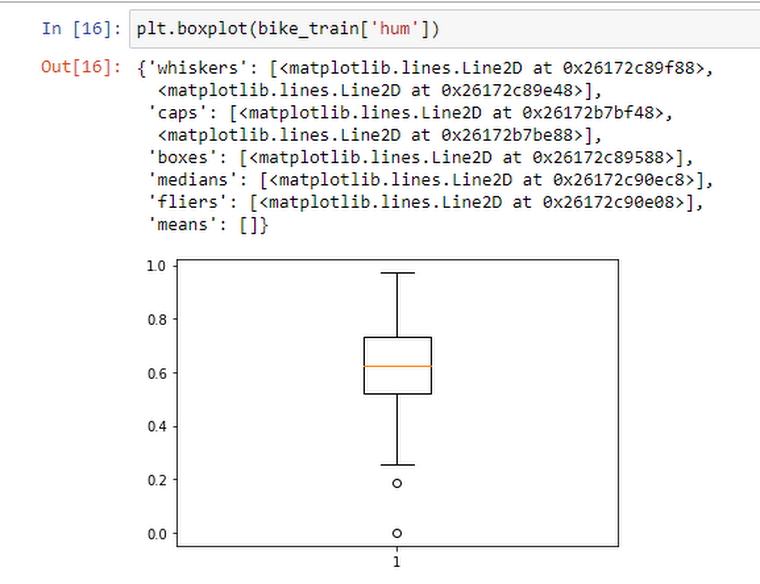


Figure 4.4 Boxplot graph of hum variable (In Python)

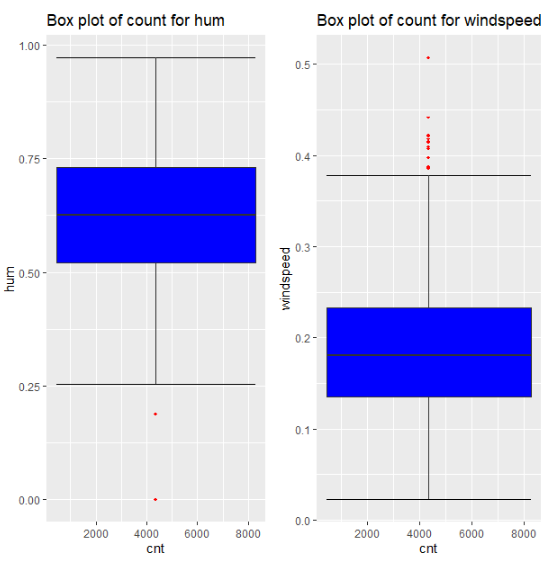


Figure 4.5 Boxplot graph of hum and windspeed variables (In RStudio)

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 variables define environmental condition. Due to drastic change in weather like strome, heavy rain condition.

**Chapter 5**

**Feature Selection and Scaling**

**5.1 Features Selection**

Machine learning works on a simple rule – if you put garbage in, you will only get garbage to come out. By garbage here, I mean noise in data.

This becomes even more important when the number of features are very large. You need not use every feature at your disposal for creating an algorithm. You can assist your algorithm by feeding in only those features that are really important. I have myself witnessed feature subsets giving better results than complete set of features for the same algorithm or – “Sometimes, less is better!”.

We should consider the selection of feature for model based on below criteria

1. The relationship between two independent variables should be less.
2. The relationship between Independent and Target variables should be high.

Feature Selection is the process of selecting the attributes that can make the predicted variable more accurate or eliminating those attributes that are irrelevant and can decrease the model accuracy and quality.

Data and feature correlation are considered one important step in the feature selection phase of the data pre-processing especially if the data type for the features is continuous.

**Positive Correlation:** Means that if feature A increases then feature B also increases or if feature A decreases then feature B also decreases. Both features move in tandem and they have a linear relationship.

**Negative Correlation:** Means that if feature A increases then feature B decreases and vice versa.

**No Correlation:** No relationship between those two attributes.

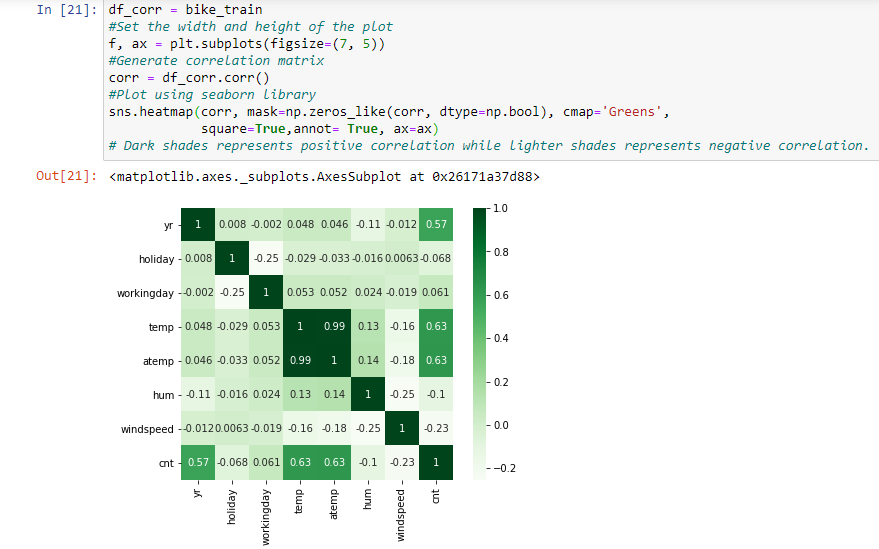


Figure 5.1 Plot of numeric variables (In Python)

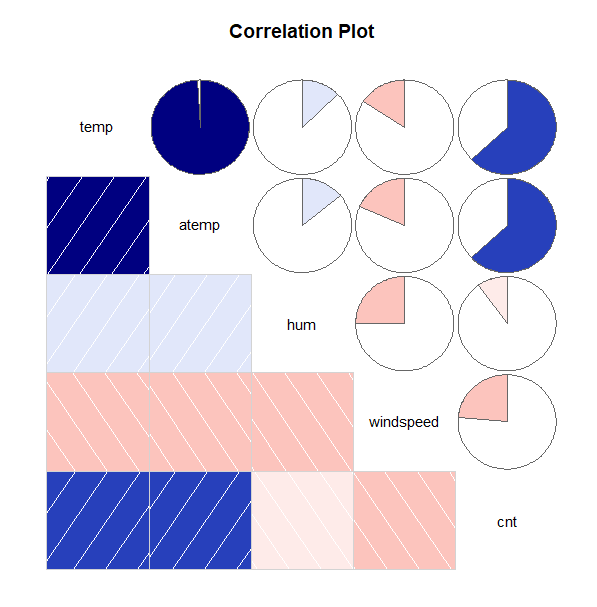


Figure 5.2 correlation plot of numeric variables (In RStudio)

In above visualization we can see that only 2 variables are highly correlated with each other. Dark blue color represents 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/dropped atemp variable from dataset.

**5.2 Feature Scaling**

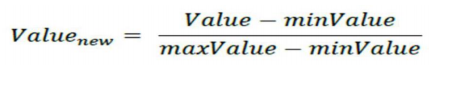
Feature scaling includes two functions normalization and standardization. It is done reduce unwanted variation either within or between variables and to bring all of the variables into proportion with one another.

**In given dataset all numeric values are already present in normalized form.**

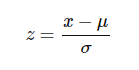
**Features Scaling Using Standardization**: Most of the Machine Learning algorithms performance depends on data we are passing through it.

If two variables are in different ranges than there is chance that Model will bias towards that higher range variable so it is important to Scale Numeric variables in same range.

As we observed in histogram that there is almost all the variable are varying from different range so first, we apply the normalization, then we are using Standardization (Z - Score) technique to scale the Numeric Variable.



The above one is the formula to normalize all the data.



The above one is the formula to Standardization all the data.

**Chapter 6**

**Modelling**

**6.1 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 we built are:**

**6.1.1 Decision Trees:**

Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

**Advantages:** Decision Tree is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data.

**Disadvantages:** Decision tree can create complex trees that do not generalize well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

In Python:

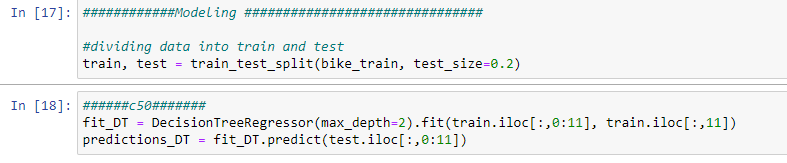


Figure 6.1 Python code for Decision Tree Regression

In RStudio:



Figure 6.2 RStudio code for Decision Tree Regression

MAPE of Decision tree regression in Python is 35.90188069313161

**6.1.2 Random Forest Classifier:**

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

**Advantages:** Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

**Disadvantages:** Slow real time prediction, difficult to implement, and complex algorithm.

In Python:

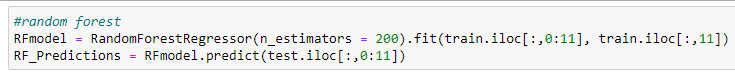


Figure 6.3 Python code for Random Forest

In RStudio:

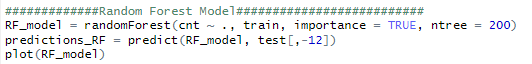


Figure 6.4 RStudio code for Random Forest

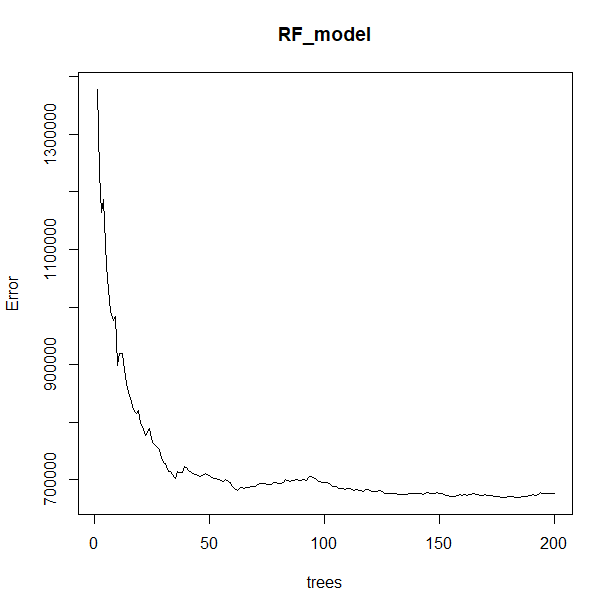


Figure 6.5 RStudio output of Random Forest

MAPE of random forest regression Python is 21.18620576885162

**6.1.3 Linear Regression:**

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

**Advantages:** Linear regression is great when the relationship to between covariates and response variable is known to be linear. This is good as it shifts focus from statistical modeling and to data analysis and preprocessing. It is great for learning to play with data without worrying about the intricate details of the model.

**Disadvantages:** A clear disadvantage is that Linear Regression over simplifies many real-world problems. More often than not, covariates and response variables don’t exhibit a linear relationship.

In Python:

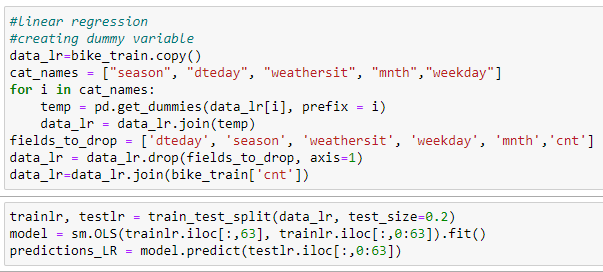


Figure 6.6 Python code for Linear Regression

In RStudio:

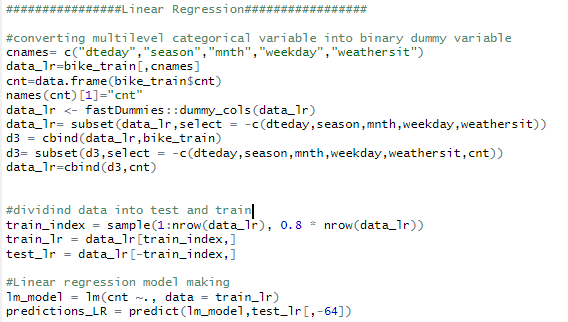


Figure 6.7 RStudio code for Linear Regression

MAPE of linear regression Python is 21.990433437865846

**Conclusion**

By using all the above modal here, the results we get this table of results, so we will go with Random forest.

|  |  |
| --- | --- |
| **Classification Algorithms**  **(Models)** | **MAPE** |
| Decision Tree | 35.90188069 |
| Random Forest | 21.18620576 |
| Linear Regression | 21.99043343 |