**Predicting Bike Rental Count**

*Sivamani Murugan*

*07th December 2019*

Contents

[Introduction 3](#_Toc26657599)

[1. Problem Statement 3](#_Toc26657600)

[2. Data 3](#_Toc26657601)

[Exploratory Data Analysis 4](#_Toc26657602)

[1. Analysis: 4](#_Toc26657603)

[2. Conclusion on Preliminary Analysis 6](#_Toc26657604)

[3. Pre Processing 6](#_Toc26657605)

[Missing Value Analysis 6](#_Toc26657606)

[Feature Selection 6](#_Toc26657607)

[Outlier Analysis 7](#_Toc26657608)

[Data Sampling 9](#_Toc26657609)

[4. Model Development and Accuracy Metrics 9](#_Toc26657610)

[1. Linear Regression Model 9](#_Toc26657611)

[2. Random Forest Regression Model 10](#_Toc26657612)

[3. Finalize the Model 10](#_Toc26657613)

[5. Test the Sample Data 11](#_Toc26657614)

[1. Existing one sample data 11](#_Toc26657615)

[2. Using Test csv file 11](#_Toc26657616)

[Appendix A – Additional Figures 12](#_Toc26657617)

[Appendix B – Python Code 12](#_Toc26657618)

[Appendix C – R Code 12](#_Toc26657619)

# Introduction

## Problem Statement

Predicting the Bike Rental Count based on the Historical data. This will help Client to understand the Bike Rental Count pattern and based on our prediction, client make necessary arrangements to have bikes available. The data provided to us contains of various attributes to decide the bike rental count like weather, season, etc.

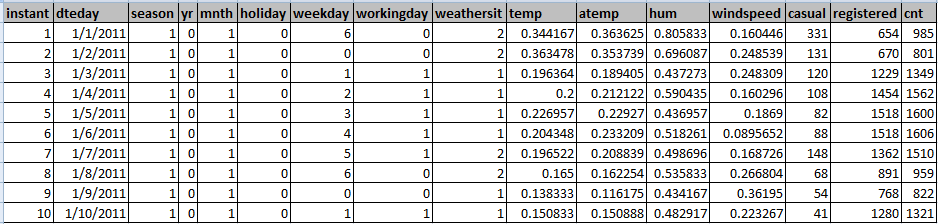
## Data

Historical Data File Name: “*Day.csv*”

*Table 1.1: Data variables*

|  |  |
| --- | --- |
| **Variable Name** | **Explanation** |
| *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) |
| *holiday* | Indicates whether it is holiday or not (0: holiday, 1: Not Holiday) |
| *weekday* | Day of the week (0 to 6) |
| *workingday* | Indicates whether it is working(excluding week end and holiday’s)  0: non- working day, 1: working day |
| *weathersit* | Indicates weather on the given day  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 (Min = -8, Max = +39) |
| *atemp* | Normalized Feeling Temperature in Celsius (Min = -16, Max = +50) |
| *hum* | Normalized Humidity (values are divided to Max= 100) |
| *windspeed* | Normalized WindSpeed (values are divided to Max=67) |
| *casual* | Count of Casual Bike Booking count |
| *registered* | Count of Registered Bike Booking count |
| *cnt* | Count of total Bike Booking count (casual + registered) |

*Table 1.2: Sample Data*



# Exploratory Data Analysis

## Analysis:

1. The given “day.csv” file has **16** **variables with 731 observations**
2. Below are the numerical/continuous variables
   1. *temp*
   2. *atemp*
   3. *hum*
   4. *windspeed*
   5. *casual*
   6. *registered*
   7. *cnt*
3. Below are the categorical variables
   1. season
   2. yr
   3. mnth
   4. holiday
   5. weekday
   6. workingday
   7. weathersit
4. Based on the Understanding “cnt” target variable with respect to other categorical variables, we can understand the pattern of Bike rental.
5. Below the graphs which explains sum of the “cnt” against all the categorical variables

*Figure 1.1: Bike Count based on different Independent variables (refer Python/R code)*

|  |  |
| --- | --- |
| E:\Data Analytics\Project2\YearWise_BikeCount.png | E:\Data Analytics\Project2\SeasonWise_BikeCount.png |
| E:\Data Analytics\Project2\WeatherWise_BikeCount.png | E:\Data Analytics\Project2\TempWise_BikeCount.png |
| E:\Data Analytics\Project2\HumidityWise_BikeCount.png | E:\Data Analytics\Project2\WindspeedWise_BikeCount.png |
|  | |

## Conclusion on Preliminary Analysis

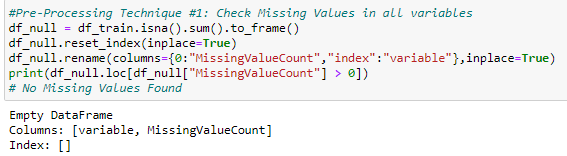
*Table 1.3: Preliminary Prediction*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Season** | **Temperature** | **Humidity** | **Weather** | **Wind speed** | **Bike Count** |
| Fall /summer/Winter | Between 0.4 to 0.7 | Between 0.5 to 0.8 | Clear/Mist | Between 0.1 to 0.3 | High Bookings |
| Other seasons | Between 0 to 0.3  Or Between 0.8 to 1 | Between 0 to 0.3  Or Between 0.9 to 1 | Other weather | Between 0.35 to 0.6 | Very Low Bookings |

## Pre Processing

### Missing Value Analysis

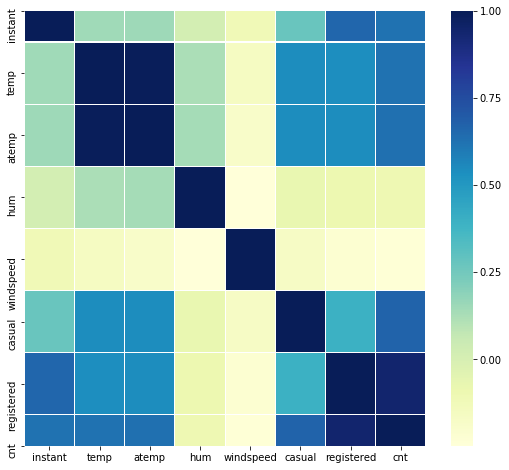
1. Based on the given data, we don’t see any Missing Value in any of the given variables



### Feature Selection

1. For Numerical variables, we can check Co linearity between the variables to understand the dependency
2. Since multiple variables with same information causes model performance, we need to remove one of the variable among two variable which are dependent
3. I have used the Correlation Graph to identify the Multi-Co linearity between Numerical Independent variables

*Figure 2.1: Correlation Graph (refer Python/R code)*



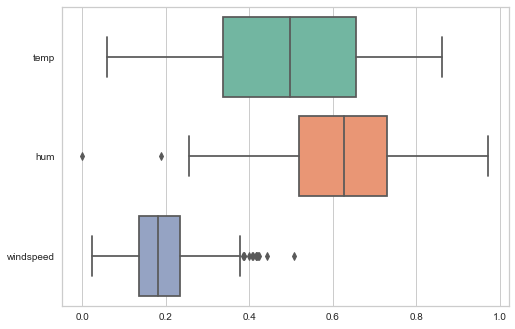
1. Based on the Correlation graph, we observe that “atemp” and “temp” are collinear to each other.
2. So, we can drop the “atemp” variable from further analysis
3. Also, we can drop “casual” and “registered” variables, since we considered “cnt” as target variable which is sum of “casual” and “registered”
4. For Categorical Test, we can identify Multi-Co linearity based on either ANOVA test or Chi-Square test.
5. We have used ANOVA test to verify the Multi-Co linearity between categorical variables
6. And as per the ANOVA, all the categorical variables carry meaningful information about the Target variable.
7. In addition, we can drop “dteday” variable, since the required information is split across “yr”,”mnth”,”weekday”.
8. Below are the final Variables after Feature Selection

*'season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday', 'weathersit', 'temp', 'hum', 'windspeed', 'cnt'*

### Outlier Analysis

1. The Outlier analysis on Numerical variables is important to have good data for model prediction.
2. We have “temp”,”hum” and “windspeed” numerical variables and out of which, we see “hum” and “windspeed” has Outliers as below

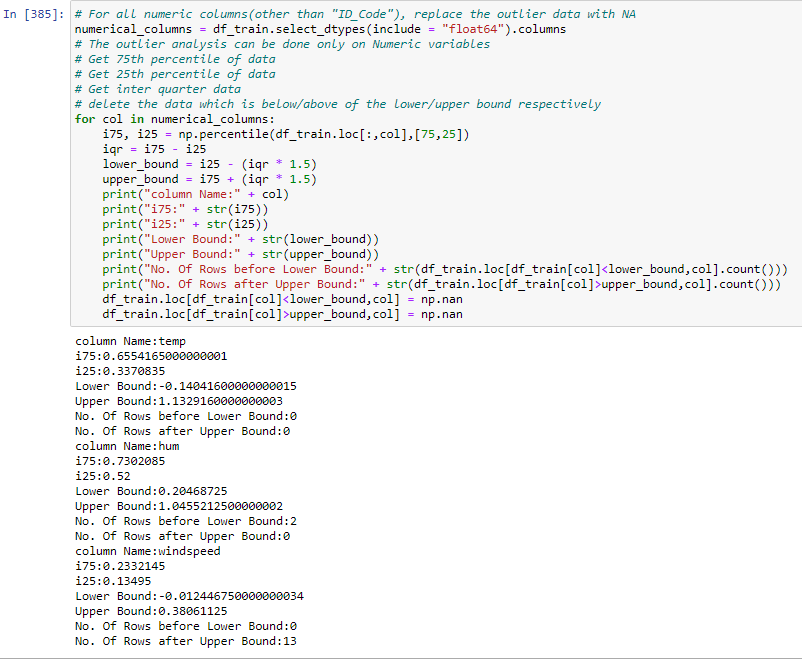
*Figure 2.2: Outlier Data for Numerical variables (refer Python/R code)*



|  |  |
| --- | --- |
| *Figure 2.3: Histogram of “hum” with Outlier Data* | *Figure 2.4: Histogram of “hum” without Outlier Data* |
| E:\Data Analytics\Project2\WithOutlier_hum.png | E:\Data Analytics\Project2\WithOutOutlier_hum.png |
|  |  |
| *Figure 2.5: Histogram of “windspeed” with Outlier Data* | *Figure 2.6: Histogram of “windspeed” without Outlier Data* |
| E:\Data Analytics\Project2\WithOutlier_windspeed.png | E:\Data Analytics\Project2\WithOutOutlier_windspeed.png |

1. To remove the Outlier, we have updated the outlier data with “NA” and imputed the “NA” with Mean of the corresponding columns

*Figure 2.7: Python Code to check the Outlier and Update with NA*



### Data Sampling

1. Sampling in necessary to train and test the model to understand the Model performance
2. As a standard practice, we should take **80% of sample** and **20% of test** data.
3. I have used simple sample, to get the sample

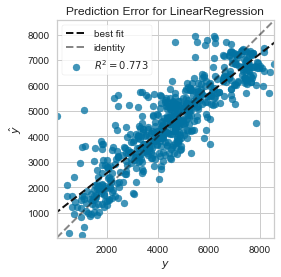
## Model Development and Accuracy Metrics

## Linear Regression Model

1. Train the model with 80% of data with all final list of variables
2. Below are the accuracy metrics

|  |  |  |
| --- | --- | --- |
| **Metrics** | **Python Values** | **R Values** |
| R-Squared | 77.3% | 84.25% |
| Mean Squared Error(MSE) | 838411.88 | 579230.3 |
| Mean Absolute Percentage deviation(MAPE) | 58.31% | 18.52% |
| Min Max Accuracy | NA | 84.67% |
| Correlation Accuracy | NA | 90.2% |

*Figure 4.1: Prediction Error for Linear Regression in Python*

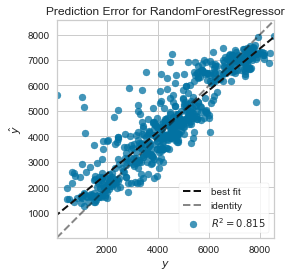


## Random Forest Regression Model

* 1. Train the model with 80% of data with all final list of variables
  2. Below are the accuracy metrics

|  |  |  |
| --- | --- | --- |
| **Metrics** | **Python Values** | **R Values** |
| R-Squared | 81.5% | NA |
| Mean Squared Error(MSE) | 684469.02 | NA |
| Mean Absolute Percentage deviation(MAPE) | 63.94% | 14.44% |
| Min Max Accuracy | NA | 89.21% |
| Correlation Accuracy | NA | 94.3% |

*Figure 4.1: Prediction Error for Random Forest Regression in Python*



## Finalize the Model

* 1. Based on the accuracy metrics, in both R and Python, the “**Random Forest**” algorithm performs better when compared to Linear Regression model.
  2. Save the model to Disk

## Test the Sample Data

## Existing one sample data

1. We can provide the below sample data for the Random Forest Model to check the output

*Table 5.1: Sample Value to predict*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Season** | **Year** | **Month** | **Holiday** | **Weekday** | **Working day** | **Weather** | **Temp** | **Hum** | **Wind Speed** |
| Springer | 2011 | January | No | Wednesday | Yes | Clear, Few Clouds | 0.226957 | 0.436957 | 0.1869 |

Actual and Predicted Values

*Table 5.2: Predicted values using Python and R*

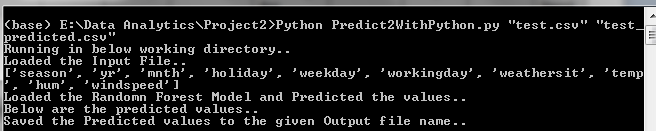
|  |  |  |
| --- | --- | --- |
| **Actual Bikes Booked** | **Python Predicted Bike Count** | **R Predicted Bike Count** |
| 1600 | 1642 | 1324 |

## Using Test csv file

1. I have developed Python and R script to provide input as Test csv file and get the predicted values based on Saved Random Forest Model
2. Below are the steps to predict using saved Random forest model using Python
   1. Place the test csv file in any folder
   2. Place the “rf\_final\_model\_py.sav” model file and “Predict2withPython.py” file in the same folder
   3. From Command Prompt, navigate to the above placed folder
   4. Run the below command

Python Predict2withPython.py <input\_csv\_filename> <output\_csv\_filename>

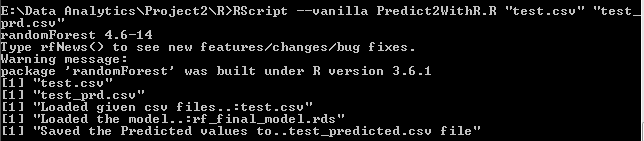
*Figure 5.1: Sample Command for Python Prediction*



1. Below are the steps to predict using saved Random forest model using R
   1. Place the test csv file in any folder
   2. Place the “rf\_final\_model.rds” model file and “Predict2WithR.R” file in the same folder
   3. From Command Prompt, navigate to the above placed folder
   4. Run the below command

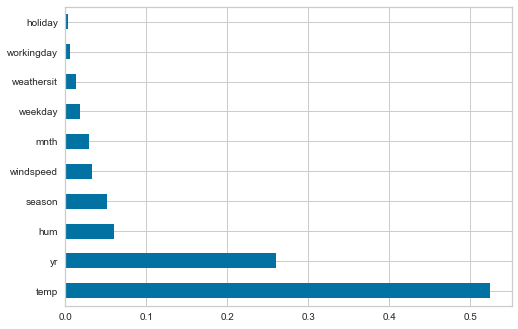
RScript --vanilla Predict2withR.R <input\_csv\_filename> <output\_csv\_filename>

*Figure 5.1: Sample Command for R Prediction*



## Appendix A – Additional Figures

*Figure 6.1: Feature Importance in Random Forest Model*



## Appendix B – Python Code

1. Refer attached “***Project2WithPython.ipynb***” Jupyter Notebook file for the complete code for model development
2. Refer attached “***Predict2WithPython.py***” python file to Predict Test data from Command Prompt(as mentioned 5.2 “Using Test csv file)

## Appendix C – R Code

1. Refer attached “***Project2WithR.R***” R file for the complete code for model development
2. Refer attached “***Predict2WithR.R***” R file to Predict Test data from Command Prompt(as mentioned 5.2 “Using Test csv file)