**Predicting Bike Rental Count**

Aditya Kanungo  
*18 August 2018*

**Contents**

**1 Introduction**  
 1.1 Problem Statement   
 1.2 Data

**2 Methodology**

2.1 Pre-Processing

2.1.1 Missing value-analysis  
 2.1.2 Outlier analysis

2.1.3 Feature selection  
 2.1.6 Outlier Analysis  
 2.1.7 Feature Scaling

2.2 Modeling  
 2.2.1 Model Selection  
 2.2.2 Regression  
 2.2.3 Random forest  
 2.2.4 Decision tree  
 2.2.5 Linear regression

**3 Conclusion**

3.1 Model Evaluation

**Appendix A – Python Code  
Appendix B – R Code  
References**

**Chapter 1**

**Introduction**

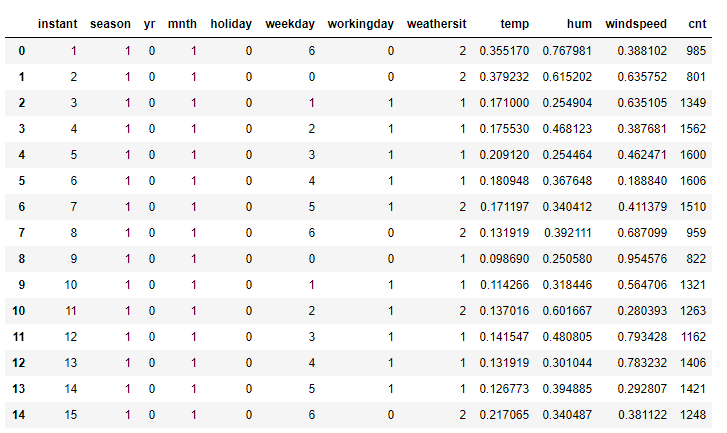
1.1 Problem Statement

The objective of this Case is to Predict the daily rental count with the help of historical bike rental data which consist of environmental and season settings.

1.2 Data

Our task is to build regression models which will predict the number of bike rentals each day which depend on multiple factors. Given below is a sample of the data set that we are using to predict the daily bike rentals.

# The details of data attributes in the dataset are as follows –

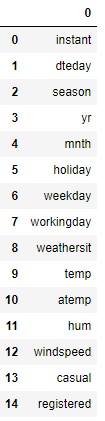


# Target variable -



# Predictor variables –

As you can see in the table below we have the following 15 variables, using which we have to correctly predict the daily bike rentals.



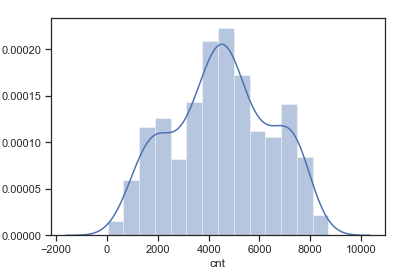
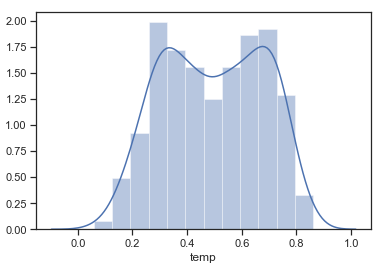
**Chapter 2**

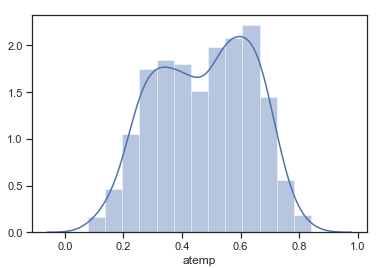
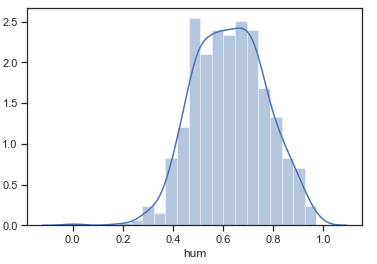
**Methodology**

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. To start this process, we will first try and look at all the probability distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the probability distributions or probability density functions of the variable.

In Figure 2.1 we have plotted the distribution of all the continuous predictors, So as you can see in the figure most variables either very closely, or somewhat imitate the normal distribution.

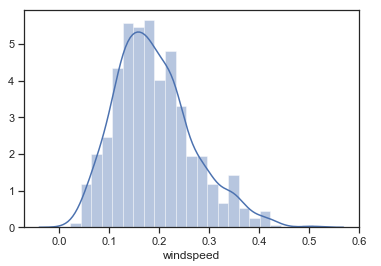


Fig. 2.1

2.1.1 Missing value analysis

We will first check for any missing values in the data to check if any variable is null or empty. During which we found that there are no missing values in our dataset.

2.1.2 Outliers analysis

In [statistics](https://en.wikipedia.org/wiki/Statistics), an outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the [data set](https://en.wikipedia.org/wiki/Data_set). An outlier can cause serious problems in statistical analyses.

Initial Box-plot analysis confirms few outliers in the numeric data. So, we will use boxplot method and inter quartile range (IQR) to find outliers in our dataset and replace it with null value or NA. Then we can impute the NA or the replaced outlier values using mean, median or knn-imputation method.

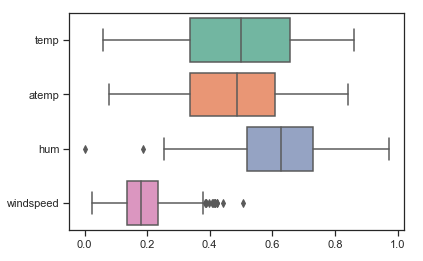


Fig. 2.2

2.1.3 Feature Selection

Before performing any type of modeling, we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. Below we have used Random Forests to perform features selection.

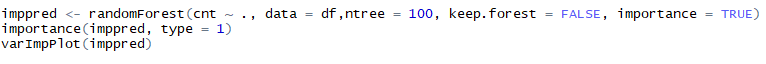
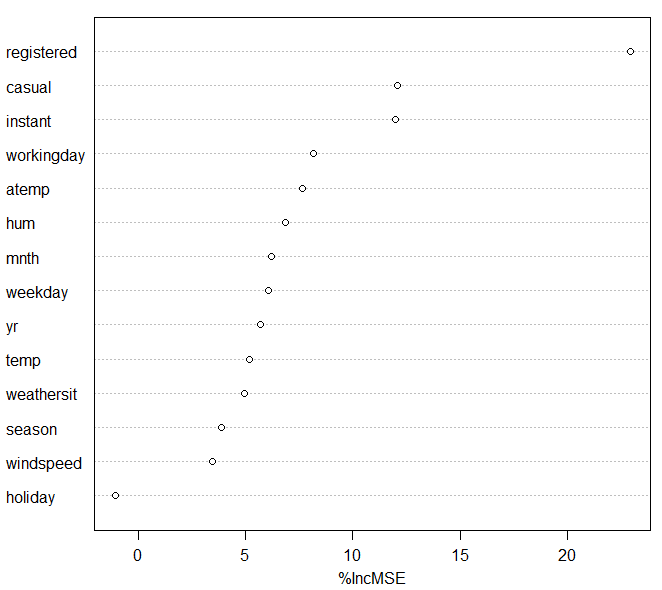
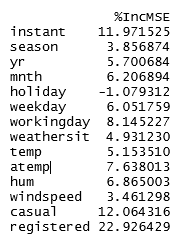


Fig. 2.3

Another step of Exploratory Data Analysis is to look for highly correlated variables in the data. A very simple way of looking at correlations in the data is shown below using co. Without much detail and at a glance you can see that **temp** and **atemp** are highly correlated.

# Correlation between variables

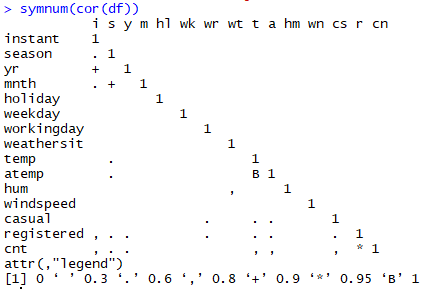
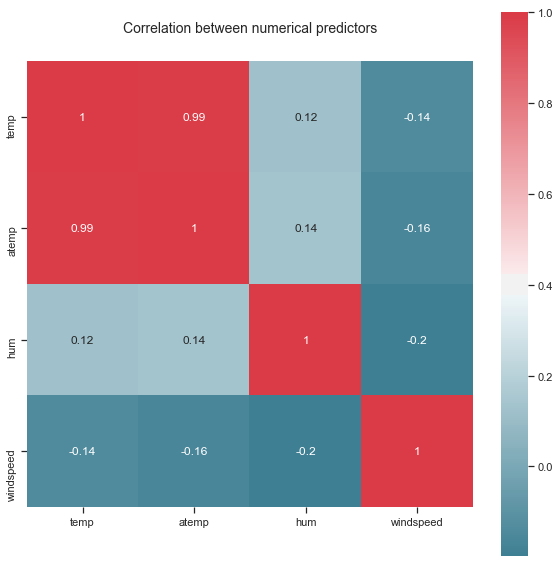


Fig. 2.4

# Heatmap between continuous variables



**# Wrapper method:**

Backward Elimination: In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

Using wrapper method, we were significantly able to improve the performance of our mode

2.1.3 Modeling

2.1.3.1 Model Selection

In our early stages of analysis during pre-processing we have come to understand that we have to predict the count of bike rentals every day. Therefore, we can use combinations of all the environment and season settings to predict the target variable. The dependent variable can fall in either of the four categories:

1. Nominal

2. Ordinal

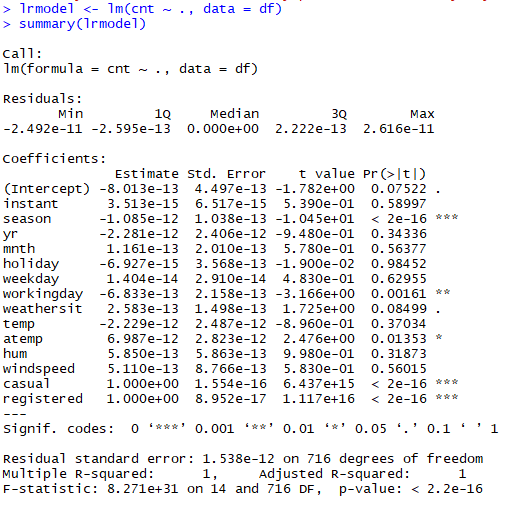
3. Interval

4. Ratio

The dependent variable we are dealing with is interval, for which regression can be done. You always start your model building from the simplest to more complex. Therefore, we use Multiple Linear Regression.

2.1.3.2 Multiple Linear regression

#Firstly we will run our model with all the variables except dteday



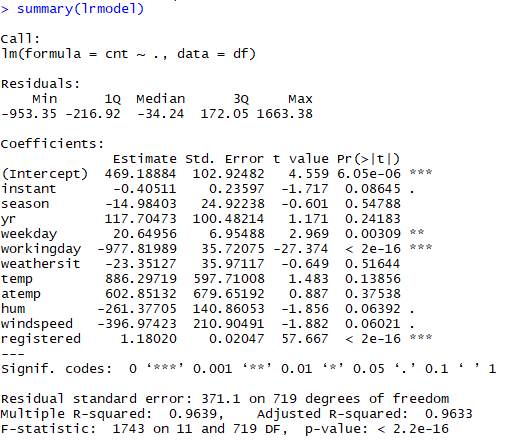
In summary.lm(lrmodel) : essentially perfect fit: summary may be unreliable

We saw that over model was overfitting, to avoid that we will drop certain features from our model based on our study using statistical models and from our observations of visualization.

# Wrapper method:

By using wrapper method, we observed that dropping few variables significantly improved our model.

# Model summary after feature selection



2.1.3.2 Random Forest Regressor

A random forest regressor.

A random forest is a meta estimator that fits a number of classifying decision trees on various sub samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

2.1.3.3 Decision Tree Regressor

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

**Chapter 3**

**Conclusion**

3.1 Model Evaluation

Any predictive Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance

2. Interpretability

3. Computational Efficiency

In our case we will use Predictive performance as the criteria to compare and evaluate models.

# Random forest:

test ->

R^2 : 0.987864054823

MAPE : 3.840853054437163

RMSE : 208.71123284

train ->

R^2 : 0.996583021642

MAPE : 3.5212136570946213

RMSE : 113.760993521

# Decision Tree:

test ->

R^2 : 0.968981993558

MAPE : 6.547376989765103

RMSE : 333.66907808

train ->

R^2 : 0.973047093277

MAPE : 9.58313523542184

RMSE : 319.503092721

# Liner regression:

test ->

R^2 : 0.966350501553

MAPE : 8.641254171187425

RMSE : 347.534815585

train ->

R^2 : 0.963200457452

MAPE : 12.72352126781848

RMSE : 373.330500246

3.2 Model Selection

We can see that random forest is having best performance, so we will use random forest.

**References**

[**https://www.analyticsvidhya.com/**](https://www.analyticsvidhya.com/)

[**https://machinelearningmastery.com**](https://machinelearningmastery.com)

[**https://docs.python.org/3/**](https://docs.python.org/3/)