Project –

BIKE RENTING PREDICTION MODEL

BY

NILANJAN SINGHA MAHAPATRA

DATE-09-JULY-2018

Introduction :

The aim of the project is to predict the bike rental count on daily based on the environmental and seasonal settings. This will help the companies to know the rental count depending on various environmental and seasonal settings and help them to grow their business better.

DATA

Our aim is to predict the count of the bikes based on provided variables ranging from day to humidity . We have continuous variable and there is no categorical variable in our dataset.

Below is the sample of our data set.

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	1/1/2011	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	1/2/2011	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
3	1/3/2011	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
4	1/4/2011	1	0	1	0	2	1	1	0.2	0.212122	0.590435	0.160296	108	1454	1562

METHODOLOGY

1.Pre-processing the data.

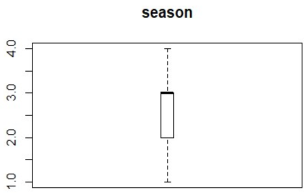
Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data. This is often called Exploratory Analysis of data.

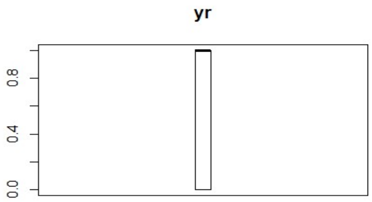
Missing Values :

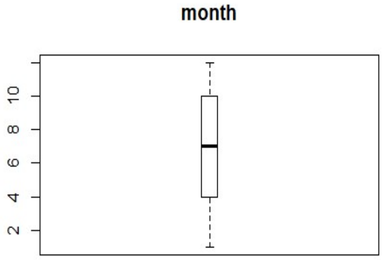
Missing values in data can change the result we expect from our models. Missing values can be different variables of the data . If Missing values is found in any observation of a particular variable we can either remove the variable if it is not that important to our model or we can input the Missing value of that observation by using different method like Mean and KNN Imputation. In our present situation there is no Missing data available in any of the variables. (See Apendix for the code on Missing Value Analysis)

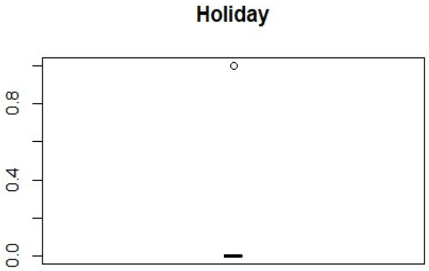
Outliers:

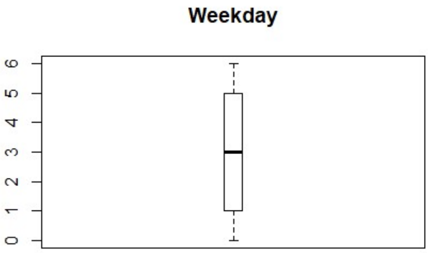
An **outlier** is an observation point that is at unusual distant from other observations.
An **outlier** may be due to variability in the measurement or it may indicate experimental error;
In case of experimental error we can exclude it from the data set.
Here we have used box plot to determine the outlier analysis of all predictor variables.
Below are the box plots of every variable for Outlier Analysis.
(See Apendix for the code on Outlier Analysis)

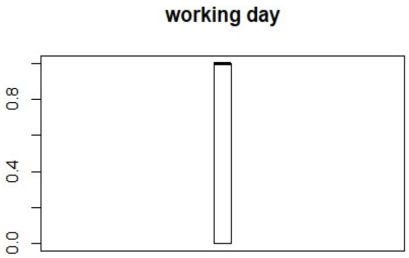

season
1.0
2.0
3.0
4.0

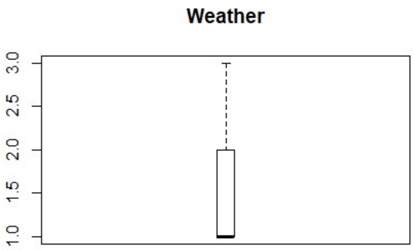

yr
0.0
0.4
0.8

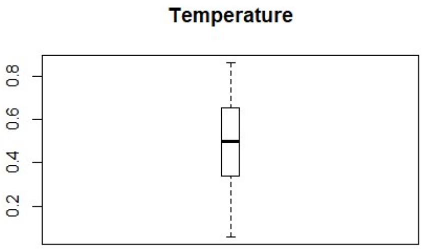

month
2
4
6
8
10

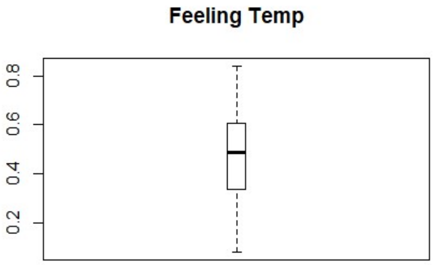

Holiday
0.0
0.4
0.8

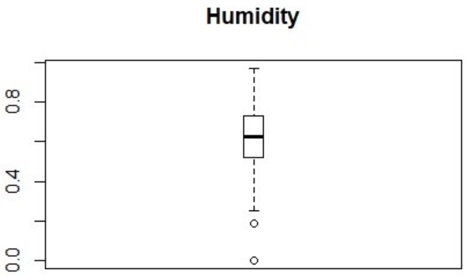

Weekday
0
1
2
3
4
5
6

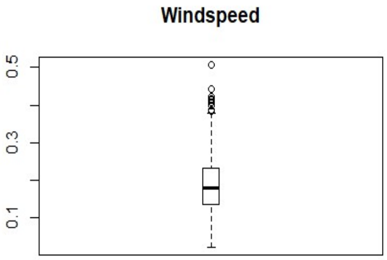

working day
0.0
0.4
0.8

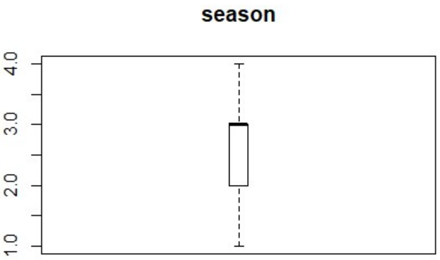

Weather

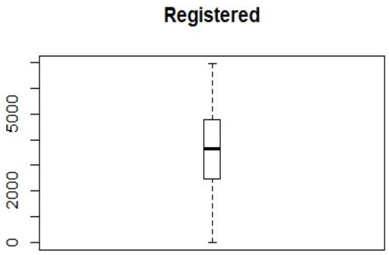

Temperature

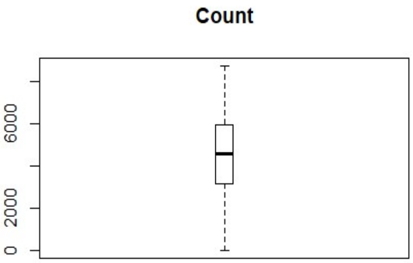

Feeling Temp


Humidity


Windspeed


season

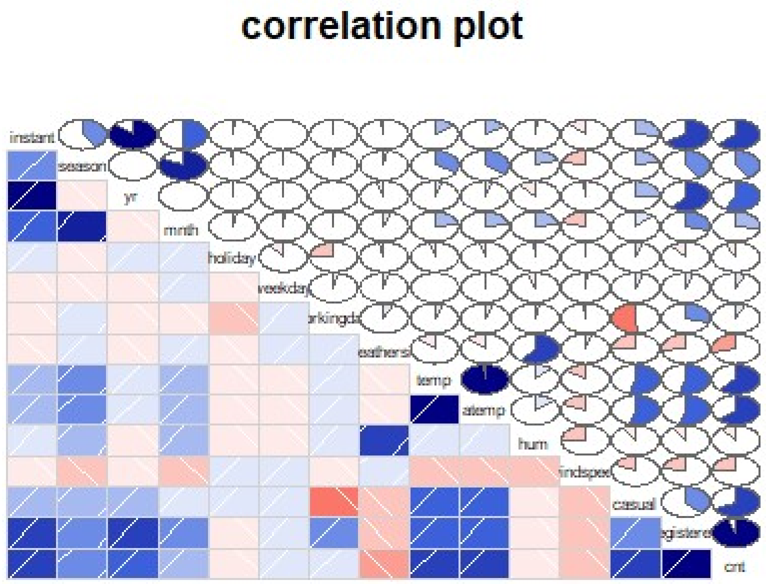

Registered

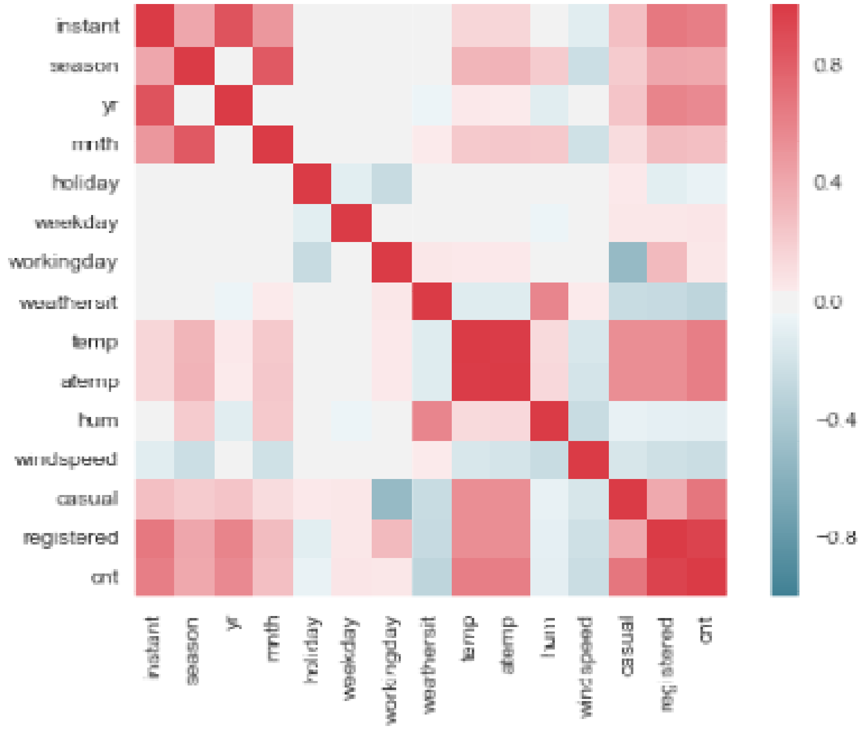

Count

Feature Selection

Feature Selection is a very important aspect of Data Pre-processing . Here We will check whether there is multicolinearity between the predictor variables. To have the best data fitted into the model there should be no correlation among independent variables . Ideally the correlation should be zero between independent variables and high among independent and dependent variables.

We will use Correlation graph and heat map to check the Correlation among the variables. If there is high correlation between two independent variables we can remove any one of them .





(Fig: Heat Map as per the correlation)

As per the above two plots/figures we will drop the below variables due to high correlation

atemp, instant, season, humidity, registered, casual

Hence dropping 6 variables out of 14 variables

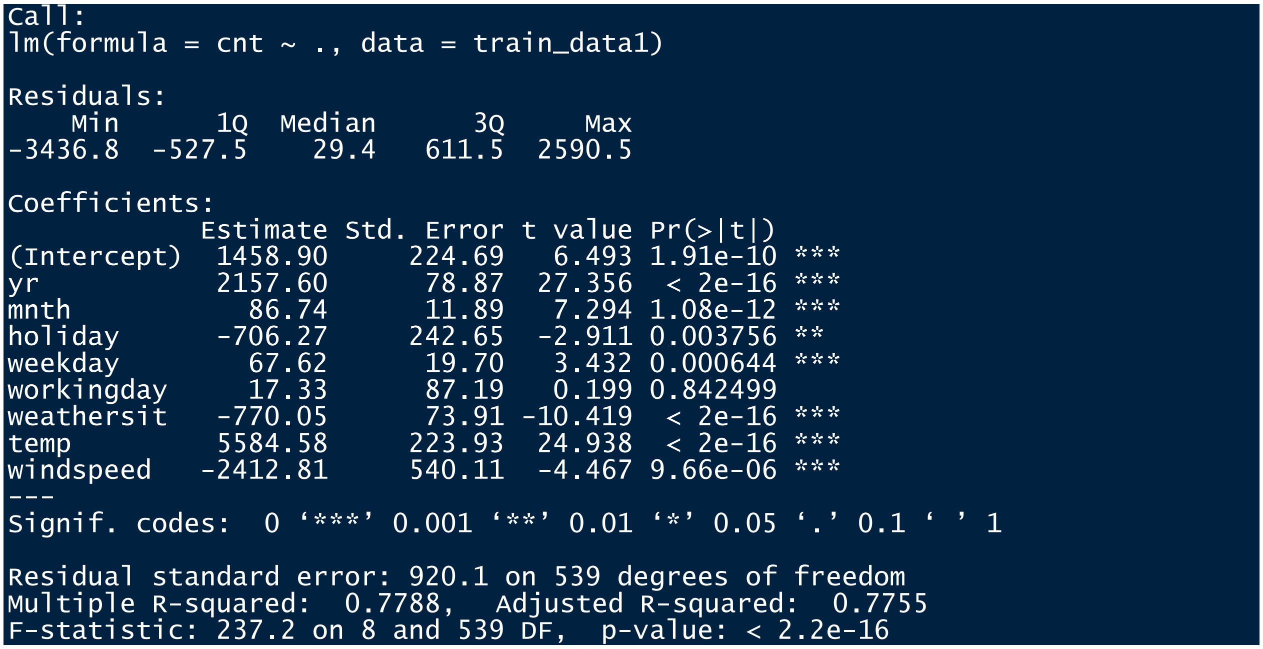
Variable ‘dteday’ is also dropped since it does not have much significance overall to the model

Model Development

After Data Pre-processing comes the Model Development part. Since we have only continuous variables with us we will go for Statistical models such as Multiple Linear Regression and KNN Regression

Multiple Linear Regression

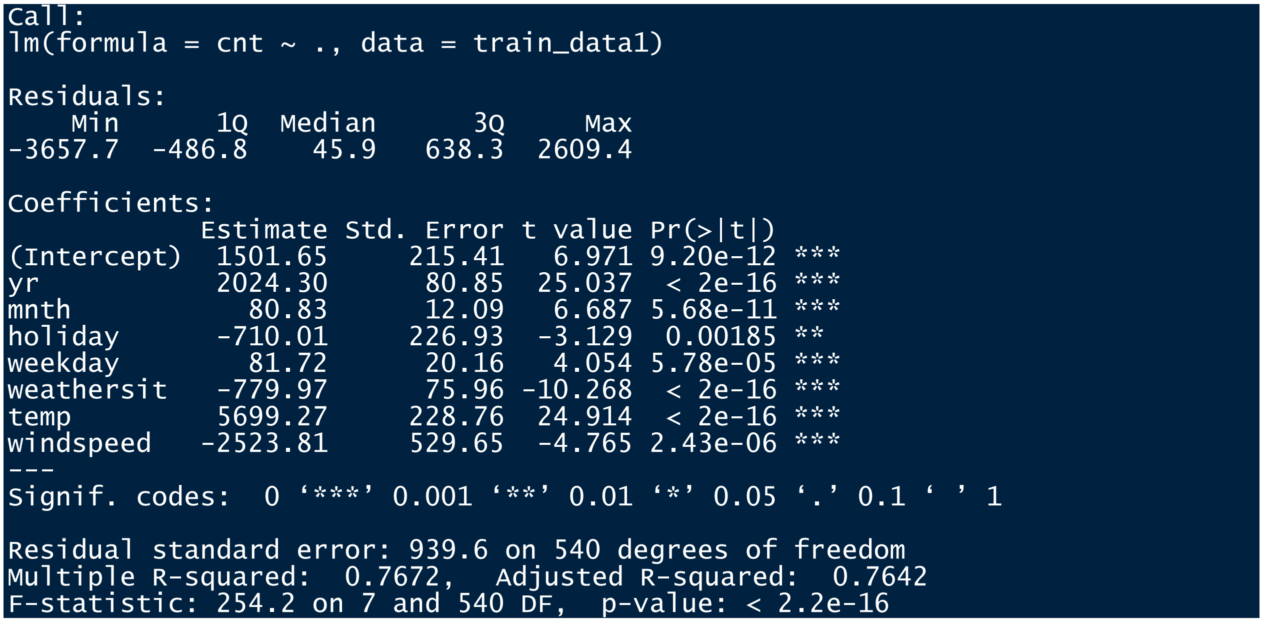
In Multiple Linear regression We will divide the pre-processed data into two parts i.e train data and test data and then fit the train data into the Multiple Linear Regression . We then check the various factors from the model to get details about the model.



```
Call:
lm(formula = cnt ~ ., data = train_data1)

Residuals:
    Min      1Q  Median      3Q     Max 
-3436.8  -527.5    29.4   611.5  2590.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1458.90     224.69   6.493 1.91e-10 ***
yr           2157.60      78.87  27.356  < 2e-16 ***
mnth           86.74      11.89   7.294 1.08e-12 ***
holiday      -706.27     242.65  -2.911 0.003756 ** 
weekday        67.62      19.70   3.432 0.000644 ***
workingday     17.33      87.19   0.199 0.842499    
weathersit   -770.05      73.91 -10.419  < 2e-16 ***
temp         5584.58     223.93  24.938  < 2e-16 ***
windspeed   -2412.81     540.11  -4.467 9.66e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 920.1 on 539 degrees of freedom
Multiple R-squared:  0.7788,	Adjusted R-squared:  0.7755 
F-statistic: 237.2 on 8 and 539 DF,  p-value: < 2.2e-16
```

As per the above figure we can see the significance level of various variables which suggests their influence on the data. The important factor Adjusted -R^2 is 77.25% which suggests we can explain 77.25% of the data which is okay. The F-statistic is 237.2 and p value is2.2e-16 which means we can reject the null hypothesis that target variable does not depend on any of the predictor.
If we look at the significance of the independent variables we can say that working day has the least significance among the variables . So we can drop it.

After dropping the working day let’s check the factors determining our model again



```
Call:
lm(formula = cnt ~ ., data = train_data1)

Residuals:
    Min      1Q  Median      3Q     Max 
-3657.7  -486.8    45.9   638.3  2609.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1501.65     215.41   6.971 9.20e-12 ***
yr           2024.30      80.85  25.037  < 2e-16 ***
mnth           80.83      12.09   6.687 5.68e-11 ***
holiday      -710.01     226.93  -3.129  0.00185 ** 
weekday        81.72      20.16   4.054 5.78e-05 ***
weathersit   -779.97      75.96 -10.268  < 2e-16 ***
temp         5699.27     228.76  24.914  < 2e-16 ***
windspeed   -2523.81     529.65  -4.765 2.43e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 939.6 on 540 degrees of freedom
Multiple R-squared:  0.7672,	Adjusted R-squared:  0.7642 
F-statistic: 254.2 on 7 and 540 DF,  p-value: < 2.2e-16
```

There has not been much change in the average R^2 value . Therefore, this is the maximum accuracy that we can get from this model.

KNN Regression

KNN Regression is a simple algorithm that stores all available cases and predicts the numerical target based on a similarity measure (e.g., distance functions).

Before implementing the KNN Regression, We will do Feature Scaling so that the data is normalized so that the output is not heavily influenced by any one set of variables.

After Normalization we will fit the data into the model and apply the trained model on test data to predict the model.

MODEL EVALUATION

For Model Evaluation we will calculate MAPE (Mean Absolute Percentage Error) for both the models :

Multiple Linear Regression – MAPE is 17.69 %

KNN Regression – MAPE is 5.86 %

CONCLUSION

As per the MAPE result We can conclude the KNN regressor has the lowest MAPE of around 6% Hence we will go for KNN as our model.

Appendix

a) Outliers

```
boxplot(data['yr'], main='year', boxwex=0.1)

boxplot(data['mnth'], main='month', boxwex=0.1)

boxplot(data['Holiday'], main='Holiday', boxwex=0.1)

boxplot(data['season'], main='Season', boxwex=0.1)

boxplot(data['weekday'], main='Weekday', boxwex=0.1)

boxplot(data['workingday'], main='Workingday', boxwex=0.1)

boxplot(data['weathersit'], main='Weathersit', boxwex=0.1)

boxplot(data['temp'], main='Temperature', boxwex=0.1)

boxplot(data['atemp'], main='Feeling Temp', boxwex=0.1)
boxplot(data['hum'], main='Humidity', boxwex=0.1)

boxplot(data['windspeed'], main='WindSpeed', boxwex=0.1)

boxplot(data['registered'], main='Registered', boxwex=0.1)

boxplot(data['cnt'], main='Count', boxwex=0.1)
```

b)Correlation Matrix and HeatMap

```
library(corrgram)

corrgram(data[,numeric_index], order=F ,upper.panel = panel.pie, text.panel = panel.txt,
main='correlation plot')
```

HeatMap

```
sms.heatmap(corr, mask=py.zeros_like(corr, dtype=py.bool),
cmap=sms.diverging_palette(220, 10, as_cmap=True),square=True, ax=ax)

plt.show()
```

Both R and Python Code are shared separately.