Project:

Prediction of Bike Rental Count

NAME: JOEL BRAGANZA

Email: joelbraganza@yahoo.in

INDEX:

Chapter 1- Introduction

Overview of the dataset

Chapter 2 – Exploratory Data Analysis

Missing values analysis

Histograms

Boxplots

Chapter 3 – Data preprocessing

Outlier removal

Data-imputation

Feature selection

Feature Scaling

Chapter 4 - Machine Learning Models

Train-test splitting

Decision-tree regression

Random-Forest regression

Linear-regression

Comparision of results & conclusion

Chapter 1:

Introduction

Problem statement - The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

Below is the sample dataset of day.csv:

instant dteday season yr mnth holiday weekday workingday \

0 1 2011-01-01 1 0 1 0 6 0

1 2 2011-01-02 1 0 1 0 0 0

2 3 2011-01-03 1 0 1 0 1 1

3 4 2011-01-04 1 0 1 0 2 1

4 5 2011-01-05 1 0 1 0 3 1

weathersit temp atemp hum windspeed casual registered \

0 2 0.344167 0.363625 0.805833 0.160446 331 654

1 2 0.363478 0.353739 0.696087 0.248539 131 670

2 1 0.196364 0.189405 0.437273 0.248309 120 1229

3 1 0.200000 0.212122 0.590435 0.160296 108 1454

4 1 0.226957 0.229270 0.436957 0.186900 82 1518

cnt

0 985

1 801

2 1349

3 1562

4 1600

This is a regression problem, where we have to predict the bike rental count.

Let’s move forward to exploring the data.

Chapter 2

Exploratory-Data Analysis:

This is the part where we explore the data, we try to understand the different variable/factors that contribute to the target variable which we’re trying to predict and maybe even bring about some changes to make it more presentable. We look at the different variables, their continuous – distributions, look for the proportion of missing-data and presence of outliers.

We have the variables:

instant (contionus, but acts as simply a serial no.)

dteday (categorical, needs alteration to make it work as such)

season (categorical)

yr (categorical)

mnth (categorical)

holiday (categorical)

weekday (categorical)

workingday (categorical)

weathersit (categorical)

temp (continuous)

atemp (continuous)

hum (continuous)

windspeed (continuous)

casual (continuous)

registered (continuous)

cnt (continuous, target variable)

This is (731 x 16) dataframe.

In our data, the variable ‘dteday’ could be considered a categorical variable, so we changed the format of this variable from: Y-M-D (2011-01-01) to simply D (01), i.e, the day data, as the yr(year) and mnth(month) variables give the remaining information.

Also we must look for any missing data in our dataset.

Missing Values Analysis:

Both Python and R give the same results as below:

varaibles percentage

0 instant 0.0

1 dteday 0.0

2 season 0.0

3 yr 0.0

4 mnth 0.0

5 holiday 0.0

6 weekday 0.0

7 workingday 0.0

8 weathersit 0.0

9 temp 0.0

10 atemp 0.0

11 hum 0.0

12 windspeed 0.0

13 casual 0.0

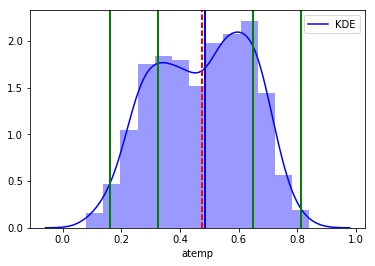
14 registered 0.0

15 cnt 0.0

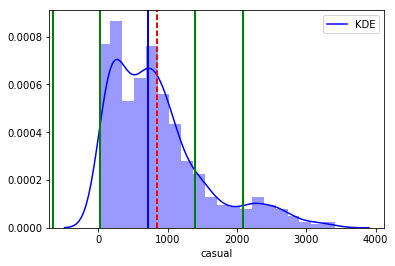
So there is no missing data till now.

Now let’s look at the data distributions(histograms) of the continuous variables.

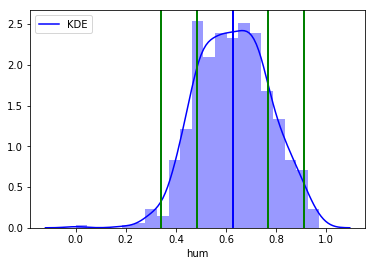
HISTOGRAMS:



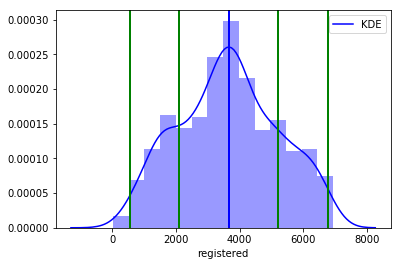
The data in ‘atemp’ seems normally distributed as the mean and median lines are very close. And most of the data too appears to be within 2 standard deviations.



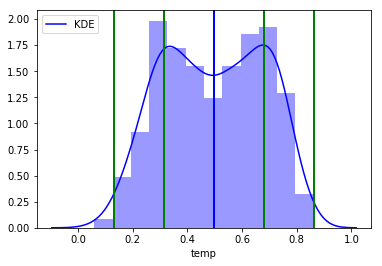
Here the data is skewed, with many being outliers also, the data is not normally distributed, based on the difference between the mean and median numerically



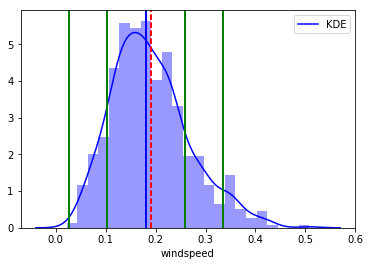
hum: Here the data appears normally distributed with mean & median line coinciding. There might be some outliers on the left(lower side).



Registered: also is a normally distributed variable



temp: also looks like a normally distributed variable



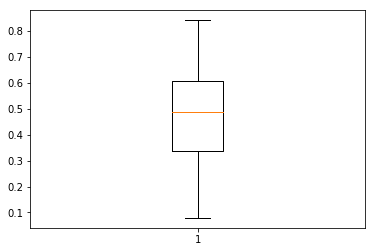
windspeed: looks like a normally distributed data with mean & median lines closeby, but there are many outliers on the right(upper) side of the data.

To make sure of the presence of outliers, let’s do an outlier analysis –

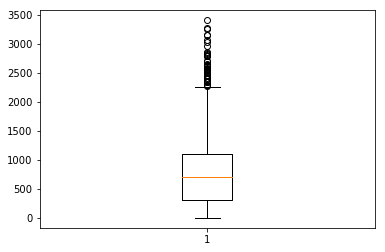
of the these continuous variables.

BOXPLOTS:

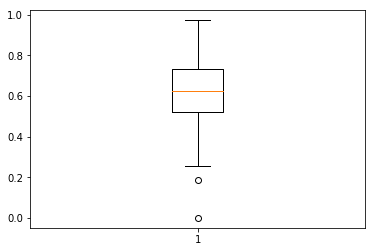
Outlier Analysis is a very important part of the data-preprocessing. We need to be sure that there are no abnormal (incorrectly recorded or exceptional data) in our dataset variables as this might affect our models and observations. To analyze the outliers we do boxplot.



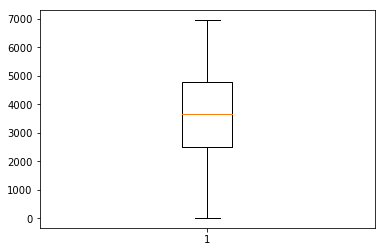
atemp: has no outliers



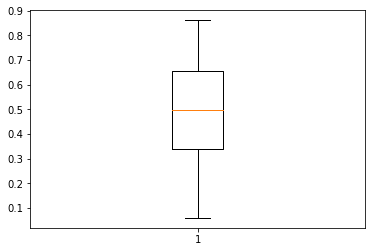
casual: like we saw in our histogram, the boxplot confirms that there are outliers in the ‘casual’ variable



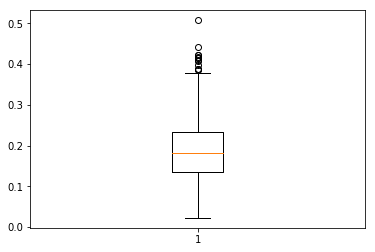
hum: the ‘humidity’ variable too has few outliers on the lower side(as we saw in the histogram).



registered: this variable has no outliers.



temp: this too has no outliers



Windspeed: this has outlier, many, on the upper-side.

The conclusions for the boxplot outlier-analysis are same in both R and Python.

So, from the boxplot, we can conclude that ‘hum’,’windspeed’ and ‘casual’ variables need an outlier removal.

Chapter 3

Data-preprocessing:

This is the part where we prepare the data for the necessary models of prediction. We do outlier-removal based on the conclusions of outlier-analysis, we do correlation-tests and ANOVA tests when the target is a continuous-variable.

Outlier-removal-

After doing the outlier removal, the percentages of missing-values are below:

varaibles percentage

12 casual 6.019152

11 windspeed 1.778386

10 hum 0.273598

0 dteday 0.000000

1 season 0.000000

2 yr 0.000000

3 mnth 0.000000

4 holiday 0.000000

5 weekday 0.000000

6 workingday 0.000000

7 weathersit 0.000000

8 temp 0.000000

9 atemp 0.000000

13 registered 0.000000

14 cnt 0.000000

With only ‘hum’,’casual’ and ‘windspeed’ variables having missing-values and none greater than 30%, therefore these values can be imputed for.

The missing-value count results are same in both R and Python. Let’s impute them.

KNN-imputation-

We have used the KNN-imputation method to impute the missing-values from the outlier-removal.

Feature selection:

Feature selection is the part where we select only those features/variables which carry the necessary information needed for the prediction of the target variable.

We might remove correlated variables, we may see if a particular variable explains the necessary variance in the target variable, if not then we remove that variable to reduce dimensionality for making the data easier, faster and simpler to work on the models. This is called ‘Dimensionality-reduction’.

ANOVA-test:

Because our variable being a continuous variable, we will do one-way-ANOVA test in both R and Python to see if the categorical-variables have any relationship(any variance explained by different classes to the mean of the target-continuous- variables)

In Python,

dteday P-value: 0.0 f-statistic: 3924.44375175

There is a relationship between col: dteday and the target variable

season P-value: 0.0 f-statistic: 3947.71375699

There is a relationship between col: season and the target variable

yr P-value: 0.0 f-statistic: 3951.216006

There is a relationship between col: yr and the target variable

mnth P-value: 0.0 f-statistic: 3940.64963675

There is a relationship between col: mnth and the target variable

holiday P-value: 0.0 f-statistic: 3952.0443759

There is a relationship between col: holiday and the target variable

weekday P-value: 0.0 f-statistic: 3946.83276815

There is a relationship between col: weekday and the target variable

workingday P-value: 0.0 f-statistic: 3950.89441239

There is a relationship between col: workingday and the target variable

weathersit P-value: 0.0 f-statistic: 3949.64633794

There is a relationship between col: weathersit and the target variable

In R,

dteday # pVal: 1>0.05 (not significant)

season # pVal: <2e-16 \*\*\*<0.05 (significant)

yr # pVal: <2e-16 \*\*\* <0.05 (significant)

mnth # pVal: <2e-16 \*\*\* <0.05 (significant)

holiday # 0.0648 >0.05 (not significant)

weekday # pVal: 0.583>0.05 (not significant)

workingday # pVal: 0.0985>0.05 (not significant)

weathersit # pVal: <2e-16 \*\*\* <0.05 (significant)

Both Python and R gave different results on the one-way-ANOVA test,

Python says that all the categorical – variables have a relationship(explains the variance in the target), as their p-value <0.05.

R on the other hand is showing here season, yr, mnth, weathersit variables are showing same p < 0.05 so are significant. However, dteday, holiday, weekday, workingday variables have p-value > 0.05 so they are not significant.

In Python, ANOVA analysis has been performed on the data saved after outlier-analysis and imputation done in R and we have the same results as would be in Python, similar results received in the vice versa process.

So we remove those insignificant variables based on the ANOVA test in R.

Correlation test:

We do it on the continuous variables to see how much correlated they are to each other so that extra variables could be removed.

R & Python show the same numeric values of correlation-test:

temp atemp hum windspeed casual registered \

temp 1.000000 0.991702 0.122413 -0.144417 0.583479 0.540012

atemp 0.991702 1.000000 0.136029 -0.169814 0.583432 0.544192

hum 0.122413 0.136029 1.000000 -0.204279 -0.085479 -0.114885

windspeed -0.144417 -0.169814 -0.204279 1.000000 -0.172089 -0.205270

casual 0.583479 0.583432 -0.085479 -0.172089 1.000000 0.414638

registered 0.540012 0.544192 -0.114885 -0.205270 0.414638 1.000000

cnt 0.627494 0.631066 -0.124448 -0.218978 0.680205 0.945517

cnt

temp 0.627494

atemp 0.631066

hum -0.124448

windspeed -0.218978

casual 0.680205

registered 0.945517

cnt 1.000000

We can see that the ‘atemp’ and ‘temp’ are highly correlated, so we remove the ‘atemp’ variable. We can also see that 'registered' has a very strong linear-relationship with the target 'cnt' variable. Also 'casual' variable has a decent linear-relationship with the target. On exploring the 2 variables: casual & registered, we realize that : casual + registered=cnt.

We can assume that therefore linear-regression might play a big role in predicting the target variable accurately.

Feature Scaling:

Normalizing: We now normalize the continuous-variables in the dataset to avoid the influence of one continuous-variable over the other. We do it by bringing each variable value in a range between 0 to 1.

Chapter 4

Machine Learning Models

We now split the data into train and test. We apply the machine learning models over the training data to train those models over, then we use apply those models over the test data and use the error metrics to see how well those models predict the target variable values.

**Decision Tree**

The decision tree algorithm accept continuous and categorical variables as independent variables .A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utilityThe branches connect nodes with “and” and multiple branches are connected by “or”. It can be used for classification and regression. It is a supervised machine learning algorithm. Extremely easy to understand by the business users. Split of decision tree is seen in the below tree.

**Random Forest**

Random Forest is an ensemble technique that consists of many decision trees. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. It is called random forest as we are building n no. of trees randomly. In other words, to build the decision trees it selects randomly n no of variables and n no of observations to build each decision tree. It means to build each decision tree on random forest we are not going to use the same data.

**Linear Regression**

We use Linear-regression when the target variables is continuous. Here, the model uses the formula y= b0 +b1x1+…..+bnxn, where b0, b1 till bn are the values which the model evaluates based on the x and y values, then using those b0,b1 till bn to use them over the test data to predict the target values.

These are the models applied in both Python and R in this project. Along with these in Python I have applied K-Neighbours regressor model based on the KNN methods and also SVR(Support-Vector-regression) based on the SVM( support-vector-machines).

The error metrics:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **R-squared** | **MAPE** | **RMSE** |
| **Decision Tree** | Python: 0.983  R: 0.93 | Python: 3.57%  R: 11.32% | Python: 224.11  R: 500.18 |
| **Random Forests** | Python: 0.994  R: 0.994 | Python: 2.54%  R: 2.75% | Python: 143.18  R: 143.26 |
| **Linear-Regression** | Python: 0.993  R: 0.993 | Python: 1.68%  R: 2.28% | Python: 143.71  R: 159.64 |

From the above error metrics: R-squared(also called as co-efficient of Determination) which checks how well the model captures the variance of the test data and MAPE( Mean-absolute-Percentage-Error) with the RMSE(Root-Mean-Square-Error), we can say that Linear-regression would be the best model in the prediction of this problem because of less error and better variance capturing.