Predicting Daily bike counts on Environmental conditions

Saurabh Gautam

14/8/2019

CHAPTER 1

Introduction:

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic. Through these systems, user is able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousands bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

Apart from interesting real world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research. Opposed to other transport services such as bus or subway, the duration of travel, departure and arrival position is explicitly recorded in these systems. This feature turns bike sharing system into a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that most of important events in the city could be detected via monitoring these data.

Attribute Information:

- 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)
- hr : hour (0 to 23)
- holiday : weather day is holiday or not (extracted from [Web Link])
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.

+ weathersit :

- 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. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
- atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50 (only in hourly scale)
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

CHAPTER 2

METHODOLOGY

APPROACH: CRISP-DM PROCESS

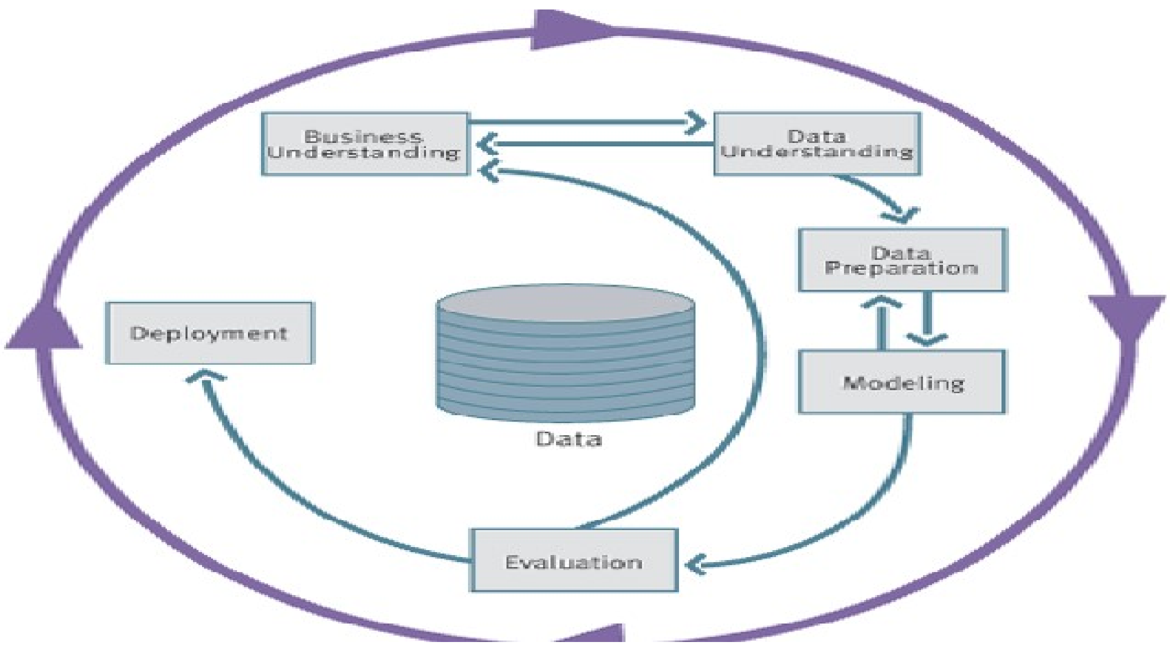


Fig1.0 To show CRISP-DM Approach

CRISP-**DM**, which stands for Cross-Industry Standard **Process** for **Data Mining**, is an industry-proven way to guide your **data mining** efforts. As a methodology, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks.

STEP1: DATA UNDERSTANDING

Understanding the **Data** in **Data Science**. The most time-consuming aspect of any **data science** project is the transformation of **data** to a format that an analyst can use to build models. Summarize the **data** by identifying key characteristics, such as **data** volume and total number of variables in the **data**

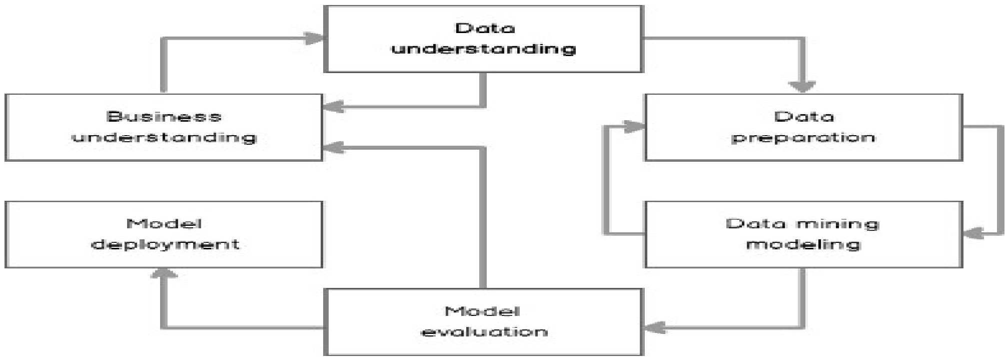


Fig 1.1 To show Data Understanding

STEP 2: DATA PREPROCESSING

As in the project data is converted to the particular data type needed to build the model

bike$weathersit = as.factor(bike$weathersit)

bike$season = as.factor(bike$season)

bike$dteday = as.character(bike$dteday)

bike$mnth = as.factor(bike$mnth)

bike$weekday = as.factor(as.character(bike$weekday))

bike$workingday = as.factor(as.character(bike$workingday))

bike$yr = as.factor(bike$yr)

bike$holiday = as.factor(bike$holiday)

STEP2.A) CHECKING FOR ANY MISSING VALUE

There is no missing value in the given data set.

check for missing values

missing_value=data.frame(apply(bike,2,function(x) sum(is.na(x))))

no missing value

write.csv(missing_value,"C:/Users/Saurabh Gautam/Desktop/project/missingvalue.csv")

STEP2.b) Analyzing categorical data using bar graph

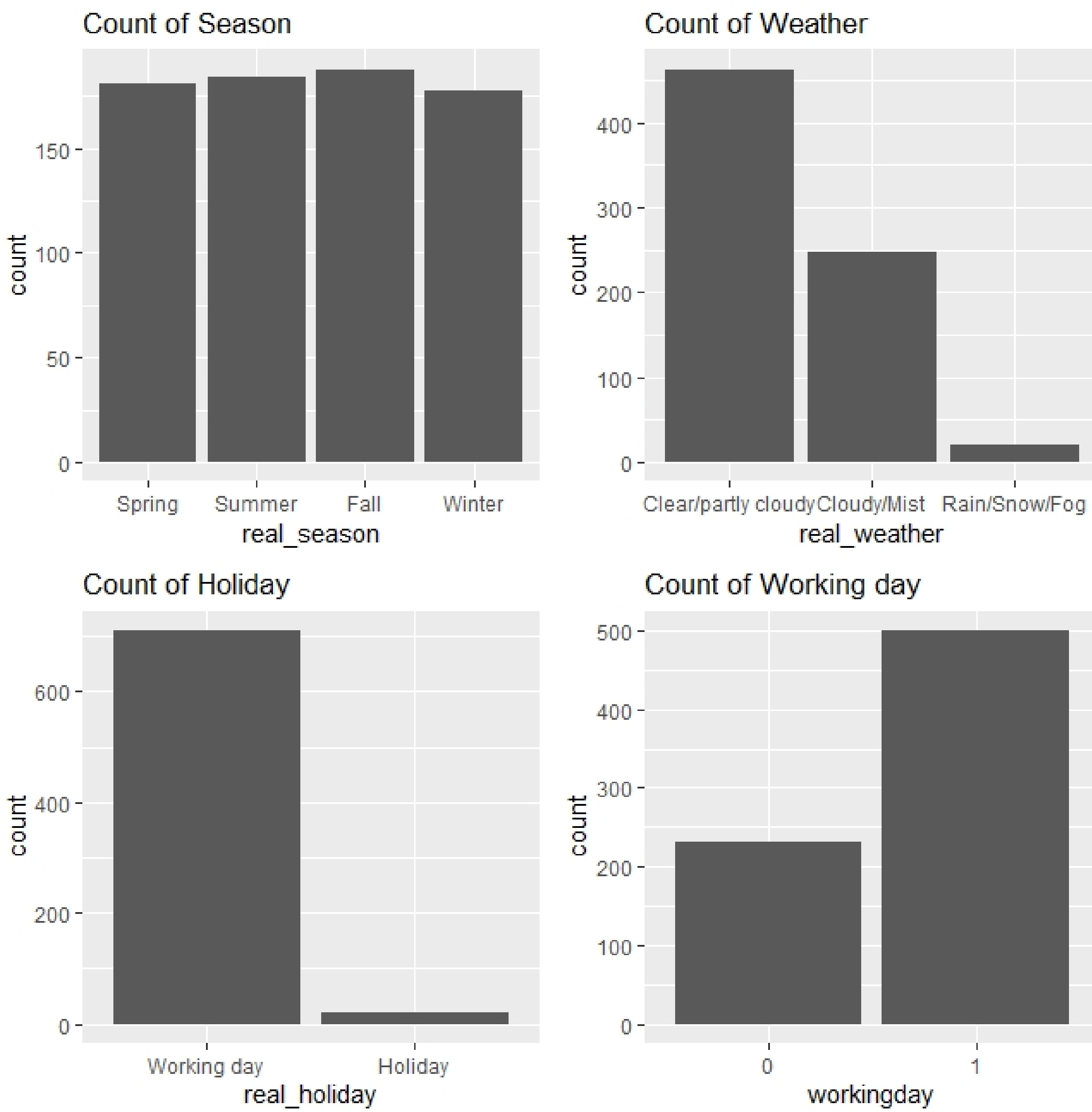


Fig 2.0 Categorical Data analyzing

Check the distribution of numerical data using histogram

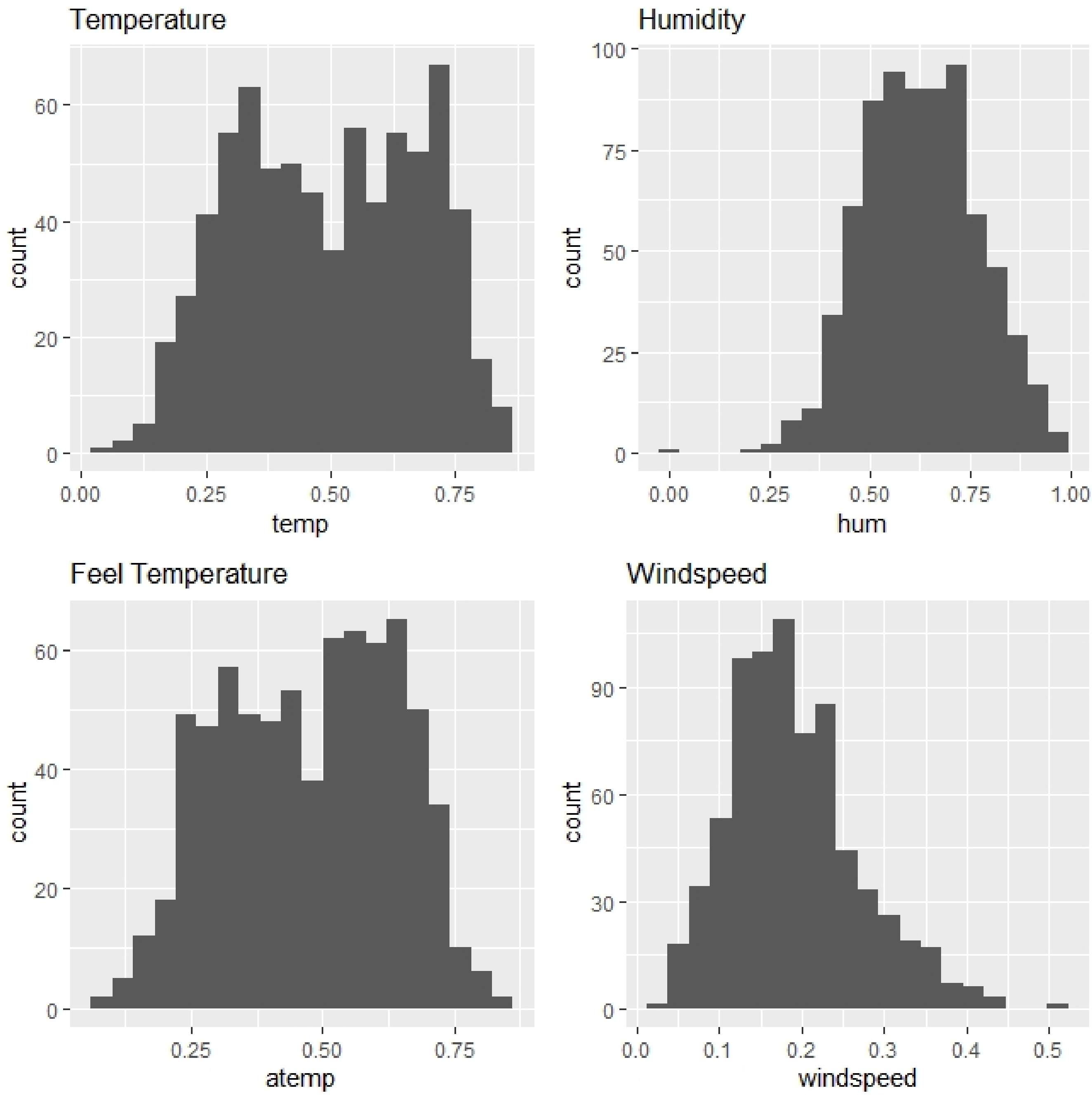


Fig 2.1 Numerical Data analyzing

Check distribution using scatter plot

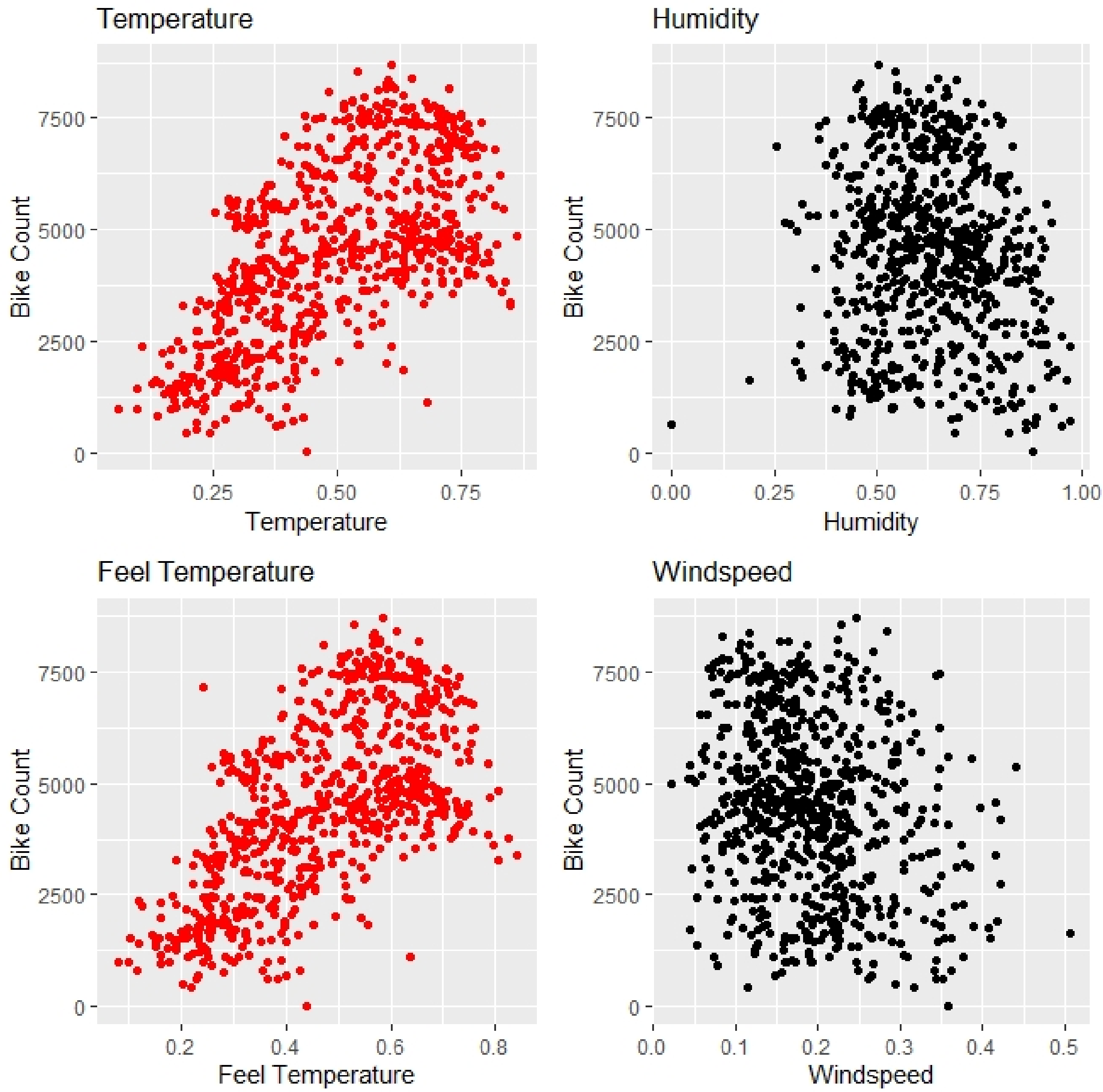
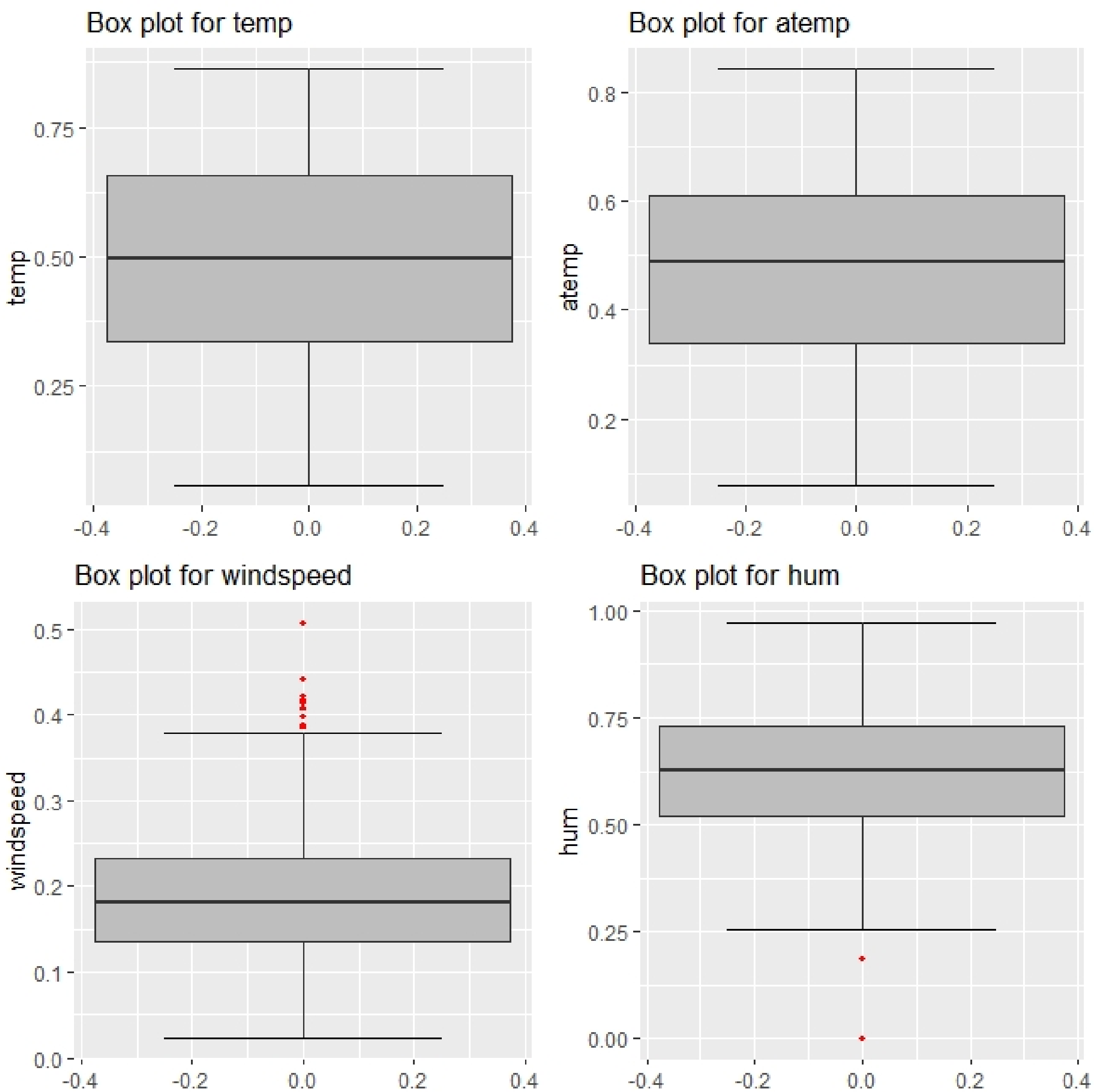


Fig 2.2 Scatter plot distribution

STEPT 2.C) Outlier analysis Using box plot on numeric values



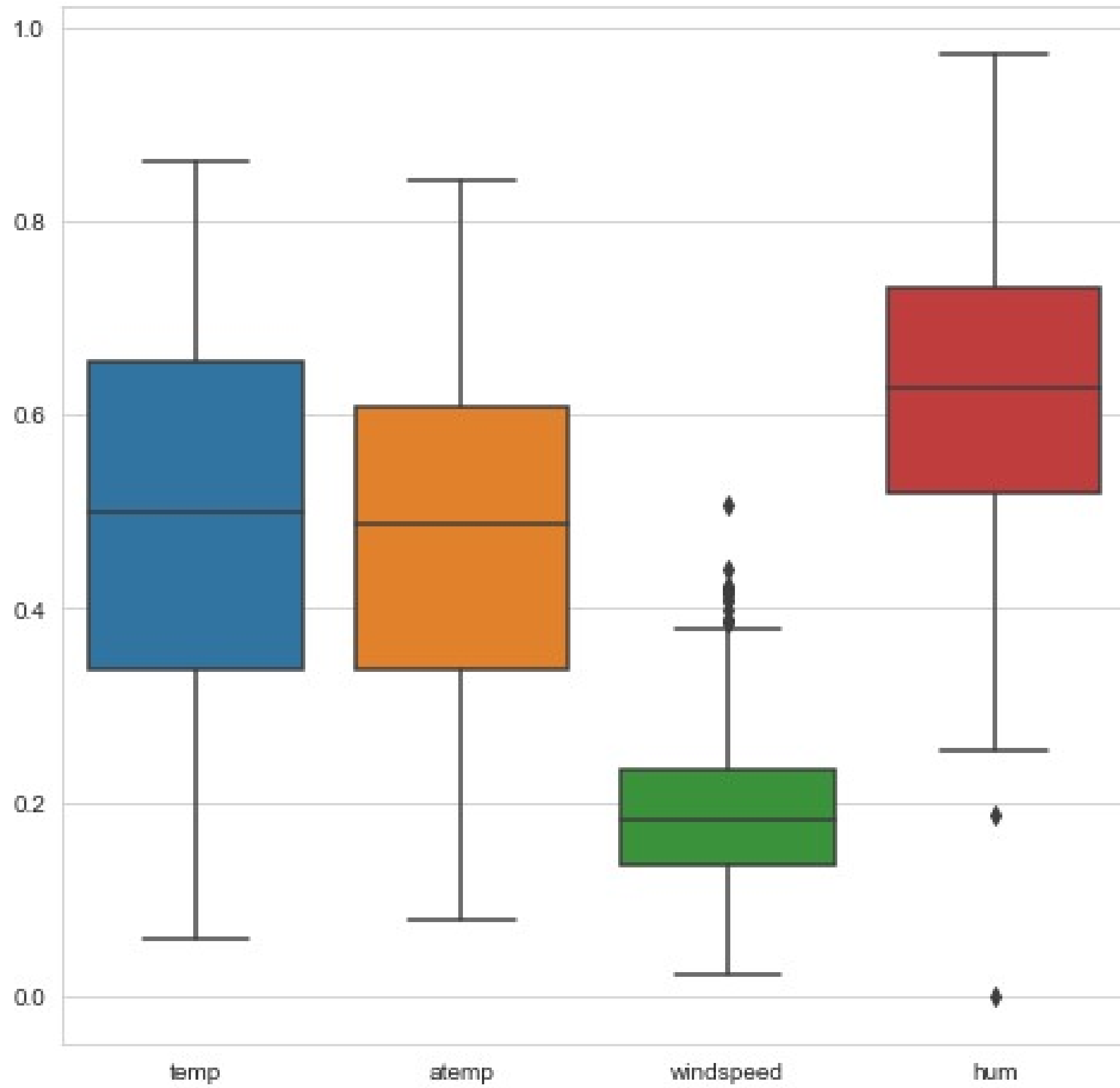


Fig 2.3 Outliers using Boxplot

As we can see in the above figure that there are outliers in windspeed (dotted point above the line)

So we need to remove the outliers otherwise they will affect the result of the mode

REMOVE OUTLIERS

val=bike$windspeed[bike$windspeed %in% boxplot.stats(bike$windspeed)$out]

bike=bike[which(!bike$windspeed %in% val),]

View(bike)

So observation reduced to 718 observations after removing outliers

Remove outliers in humidity(python)

q75, q25 = np.percentile(df['hum'], [75 ,25])

print(q75,q25)

iqr = q75 - q25

print(iqr)

min = q25 - (iqr*1.5)

max = q75 + (iqr*1.5)

print(min)

print(max)

df = df.drop(df[df.iloc[:,12] < min].index)

df = df.drop(df[df.iloc[:,12] > max].index)

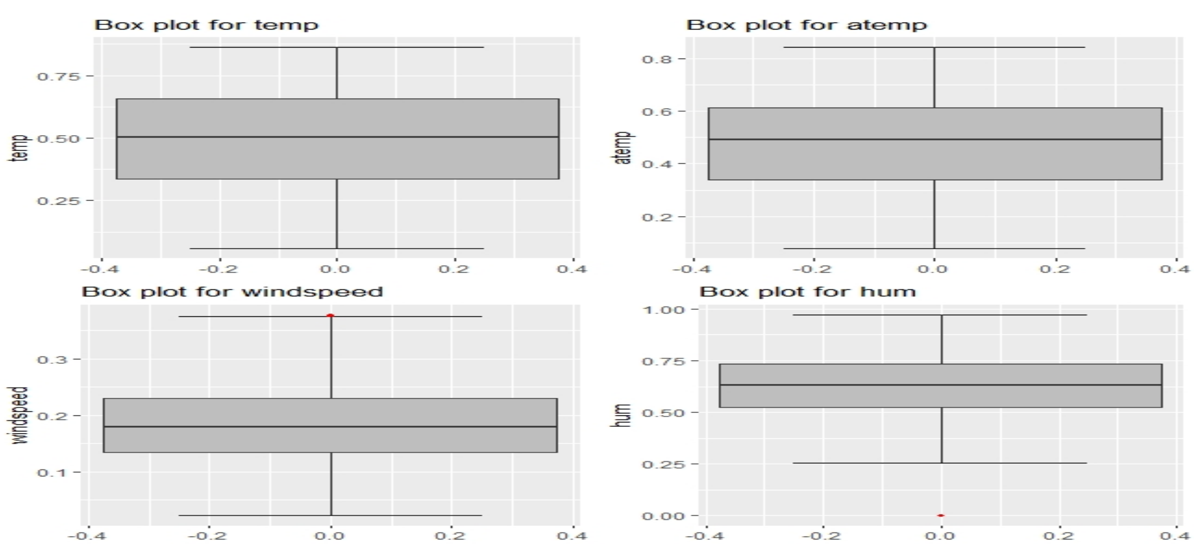


Fig 2.4 removed outliers from windspeed

STEP 3.Feature Selection

By using Correlation plot only on numerical data we can neglect variables which are highly correlated to each because deletion of highly correlated variable does not affect the result of the model

1 variables from the 4 input variables have collinearity problem:

atemp

After excluding the collinear variables, the linear correlation coefficients ranges between:

min correlation (hum ~ temp): 0.1162369

max correlation (windspeed ~ hum): -0.2080313

---------- VIFs of the remained variables --------

Variables VIF

1 temp 1.028541

2 hum 1.053726

3 windspeed 1.060541

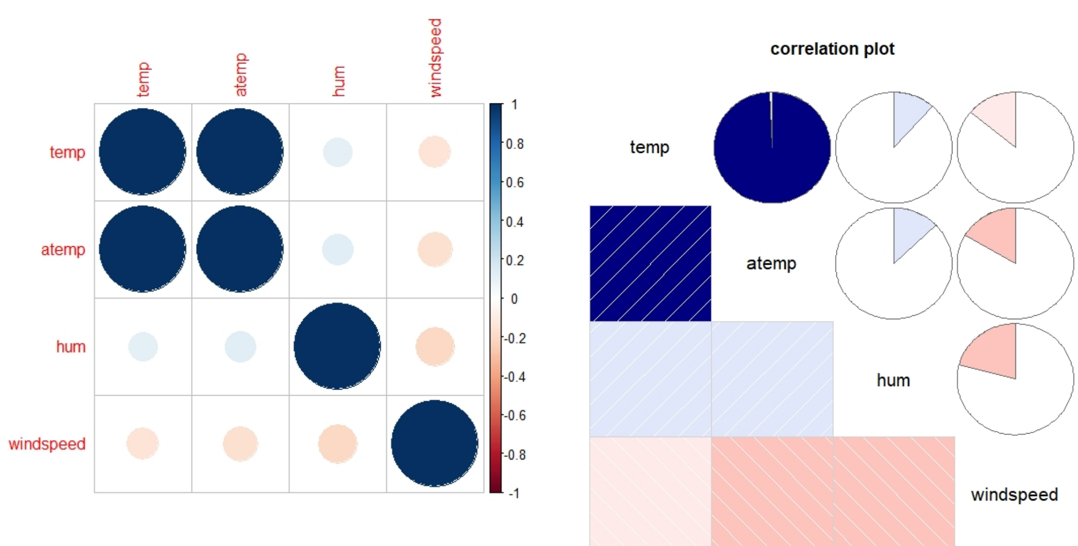
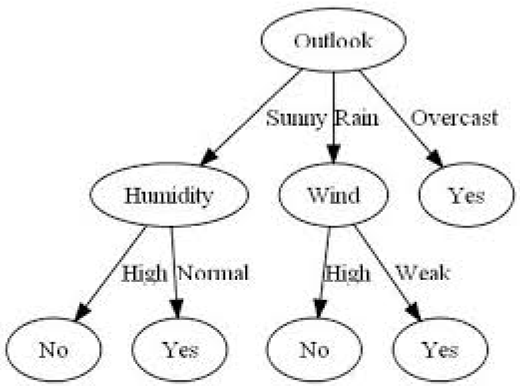


Fig 3.0 Correlation plots

Step 4. MODEL DEVELOPMENT

Using Decision Tree Machine Learning Algorithm for Regression because target variable (cnt) is continous

Decision Tree algorithm belongs to the family of supervised learning **algorithms**. ... The **decision tree algorithm** tries to solve the problem, by using **tree** representation. Each internal node of the **tree**corresponds to an attribute, and each leaf node corresponds to a class label.



set.seed(123)

train_index=sample(1:nrow(real_bike),0.8*nrow(real_bike))

train=real_bike[train_index,] #574 observation

test=real_bike[-train_index,]

rpart for regression

fit=rpart(cnt~.,data=train,method='anova')

write rule into disk

write(capture.output(summary(fit)),"rules.txt")

rpart(formula = cnt ~ ., data = train, method = "anova")

n= 574

```
CP nsplit rel error    xerror      xstd
1  0.39755531      0 1.0000000 1.0014249 0.04482121
2  0.07156473      1 0.6024447 0.6181418 0.03128687
3  0.05448738      2 0.5308800 0.5849415 0.03262530
4  0.02464143      3 0.4763926 0.5394778 0.02758118
5  0.01481200      4 0.4517511 0.4923767 0.02512372
6  0.01426961      5 0.4369391 0.5150380 0.02643032
7  0.01335812      7 0.4083999 0.5171445 0.02652735
8  0.01108524      8 0.3950418 0.5145968 0.02646190
9  0.01101758      9 0.3839566 0.5101314 0.02612933
10 0.01097902     10 0.3729390 0.5101314 0.02612933
11 0.01000000     11 0.3619600 0.4970278 0.02491894

Variable importance
     temp     season hum  windspeed weathersit
       48        31        11         6         4

Node number 1: 574 observations,    complexity param=0.3975553
  mean=4503.709, MSE=3905922
  left son=2 (234 obs) right son=3 (340 obs)
  Primary splits:
      temp       < 0.432373  to the left,  improve=0.39755530, (0 missing)
      season     splits as  LRRR, improve=0.31092610, (0 missing)
      weathersit splits as  RRL, improve=0.06788193, (0 missing)
      hum        < 0.8222915 to the right, improve=0.06497156, (0 missing)
      windspeed  < 0.184196  to the right, improve=0.05359049, (0 missing)
```

```
  Surrogate splits:
      season    splits as  LRRL, agree=0.829, adj=0.581, (0 split)
      hum       < 0.5385415 to the left,  agree=0.611, adj=0.047, (0 split)
      windspeed < 0.249379  to the right, agree=0.610, adj=0.043, (0 split)

Node number 2: 234 observations,    complexity param=0.07156473
  mean=3001.632, MSE=2319815
  left son=4 (154 obs) right son=5 (80 obs)
  Primary splits:
      season     splits as  LL-R, improve=0.29557330, (0 missing)
      temp       < 0.2748915 to the left,  improve=0.28692430, (0 missing)
      weathersit splits as  RRL, improve=0.07568364, (0 missing)
      hum        < 0.7725    to the right, improve=0.07381437, (0 missing)
      windspeed  < 0.184196  to the right, improve=0.04514850, (0 missing)
  Surrogate splits:
      windspeed < 0.10745   to the right, agree=0.744, adj=0.250, (0 split)
      hum       < 0.88      to the left,  agree=0.671, adj=0.037, (0 split)
      temp      < 0.3472285 to the left,  agree=0.667, adj=0.025, (0 split)

Node number 3: 340 observations,    complexity param=0.05448738
  mean=5537.491, MSE=2376011
  left son=6 (23 obs) right son=7 (317 obs)
  Primary splits:
      hum        < 0.8485415 to the right, improve=0.151218200, (0 missing)
      weathersit splits as  RRL, improve=0.112400400, (0 missing)
      temp       < 0.5133335 to the left,  improve=0.052213770, (0 missing)
      windspeed  < 0.154231  to the right, improve=0.040756440, (0 missing)
```

season splits as LLRL, improve=0.006536296, (0 missing)

Surrogate splits:

weathersit splits as RRL, agree=0.962, adj=0.435, (0 split)

windspeed < 0.3526145 to the right, agree=0.938, adj=0.087, (0 split)

Node number 4: 154 observations, complexity param=0.02464143

mean=2404.812, MSE=1568600

left son=8 (55 obs) right son=9 (99 obs)

Primary splits:

temp < 0.262953 to the left, improve=0.22870120, (0 missing)

hum < 0.680652 to the right, improve=0.08334158, (0 missing)

weathersit splits as RRL, improve=0.05695076, (0 missing)

season splits as LR--, improve=0.03862730, (0 missing)

windspeed < 0.1235335 to the left, improve=0.02331489, (0 missing)

Surrogate splits:

windspeed < 0.1298855 to the left, agree=0.688, adj=0.127, (0 split)

Node number 5: 80 observations

mean=4150.512, MSE=1760303

Node number 6: 23 observations

mean=3312.174, MSE=1730332

Node number 7: 317 observations, complexity param=0.014812

mean=5698.95, MSE=2037493

left son=14 (104 obs) right son=15 (213 obs)

Primary splits:

```
    hum       < 0.6947915 to the right, improve=0.05141547, (0 missing)
    windspeed  < 0.154231  to the right, improve=0.04323369, (0 missing)
    temp      < 0.5133335 to the left,  improve=0.04108706, (0 missing)
    weathersit splits as  RL-, improve=0.02701426, (0 missing)
    season     splits as  LLRR, improve=0.01476313, (0 missing)
  Surrogate splits:
    weathersit splits as  RL-, agree=0.767, adj=0.288, (0 split)
    windspeed  < 0.065     to the left,  agree=0.685, adj=0.038, (0 split)

Node number 8: 55 observations
  mean=1601.236, MSE=360209.3

Node number 9: 99 observations,    complexity param=0.01426961
  mean=2851.242, MSE=1681888
  left son=18 (38 obs) right son=19 (61 obs)
  Primary splits:
    hum       < 0.678125  to the right, improve=0.16923480, (0 missing)
    temp      < 0.3455075 to the left,  improve=0.10463320, (0 missing)
    weathersit splits as  RLL, improve=0.08714218, (0 missing)
    windspeed  < 0.3056645 to the right, improve=0.02191374, (0 missing)
    season     splits as  LR--, improve=0.00387856, (0 missing)
  Surrogate splits:
    weathersit splits as  RLL, agree=0.808, adj=0.500, (0 split)
    temp      < 0.40572   to the right, agree=0.667, adj=0.132, (0 split)
    windspeed  < 0.1276115 to the left,  agree=0.657, adj=0.105, (0 split)
```

```
Node number 14: 104 observations,    complexity param=0.01335812
  mean=5235.75, MSE=1814785
  left son=28 (48 obs) right son=29 (56 obs)
  Primary splits:
      windspeed  < 0.1744375 to the right, improve=0.158680000, (0 missing)
      temp       < 0.5108335 to the left,  improve=0.094521370, (0 missing)
      season     splits as  RLRR, improve=0.031895540, (0 missing)
      hum        < 0.810625  to the right, improve=0.027476190, (0 missing)
      weathersit splits as  RL-, improve=0.004015014, (0 missing)
  Surrogate splits:
      weathersit splits as  RL-, agree=0.587, adj=0.104, (0 split)
      season     splits as  RLRR, agree=0.577, adj=0.083, (0 split)
      temp       < 0.502989  to the left,  agree=0.577, adj=0.083, (0 split)
      hum        < 0.810625  to the right, agree=0.577, adj=0.083, (0 split)

Node number 15: 213 observations,    complexity param=0.01108524
  mean=5925.113, MSE=1990325
  left son=30 (36 obs) right son=31 (177 obs)
  Primary splits:
      temp       < 0.7591665 to the right, improve=0.058624180, (0 missing)
      windspeed  < 0.1194045 to the right, improve=0.045273530, (0 missing)
      season     splits as  LLLR, improve=0.030677110, (0 missing)
      hum        < 0.54      to the right, improve=0.017023100, (0 missing)
      weathersit splits as  RL-, improve=0.005898356, (0 missing)
```

Node number 18: 38 observations

mean=2175.289, MSE=869023.7

Node number 19: 61 observations,   complexity param=0.01426961

mean=3272.328, MSE=1726315

left son=38 (39 obs) right son=39 (22 obs)

Primary splits:

temp   < 0.3408335 to the left, improve=0.3400224000, (0 missing)

season   splits as LR--, improve=0.0567148400, (0 missing)

hum   < 0.451875 to the left, improve=0.0291496700, (0 missing)

windspeed < 0.1829545 to the right, improve=0.0261273500, (0 missing)

weathersit splits as LRL, improve=0.0003453363, (0 missing)

Surrogate splits:

hum   < 0.6563675 to the left, agree=0.689, adj=0.136, (0 split)

season   splits as LR--, agree=0.672, adj=0.091, (0 split)

weathersit splits as LLR, agree=0.656, adj=0.045, (0 split)

windspeed < 0.26135   to the left, agree=0.656, adj=0.045, (0 split)

Node number 28: 48 observations

mean=4656.125, MSE=1669692

Node number 29: 56 observations

mean=5732.571, MSE=1404350

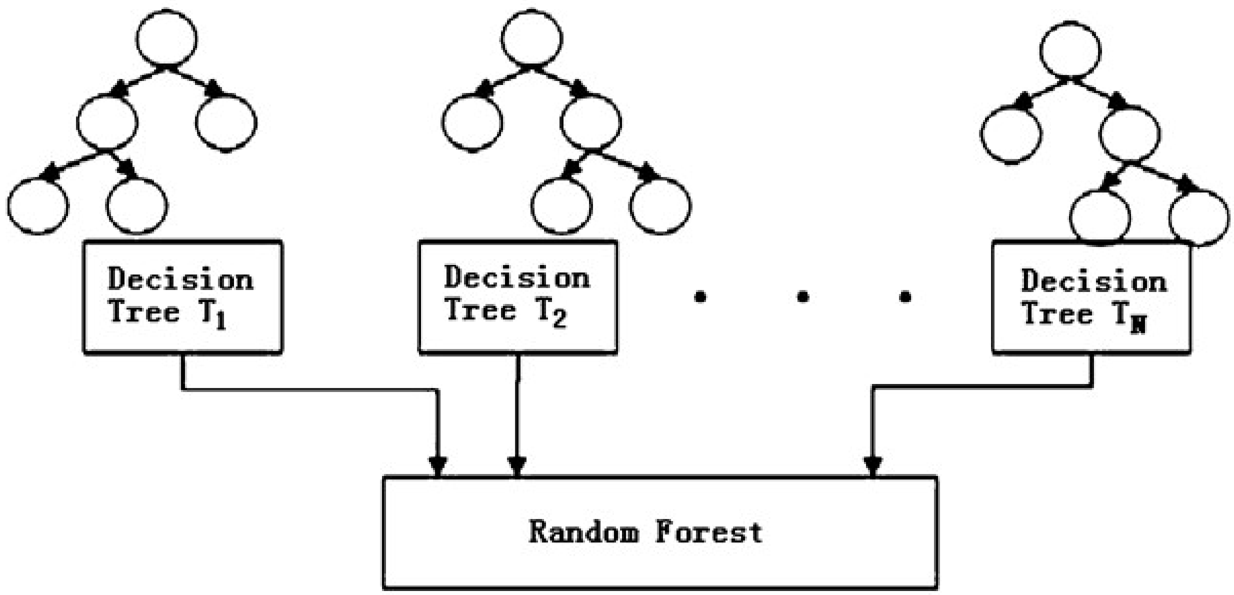
Lets predict test data

bike_predictions=predict(fit,test[-6])

= c("mae","rmse","mape"))

Using Random Forest Machine Learning Algorithm for Regression because target variable (cnt) is continous

Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.[1][2] Random decision forests correct for decision trees' habit of overfitting to their training set



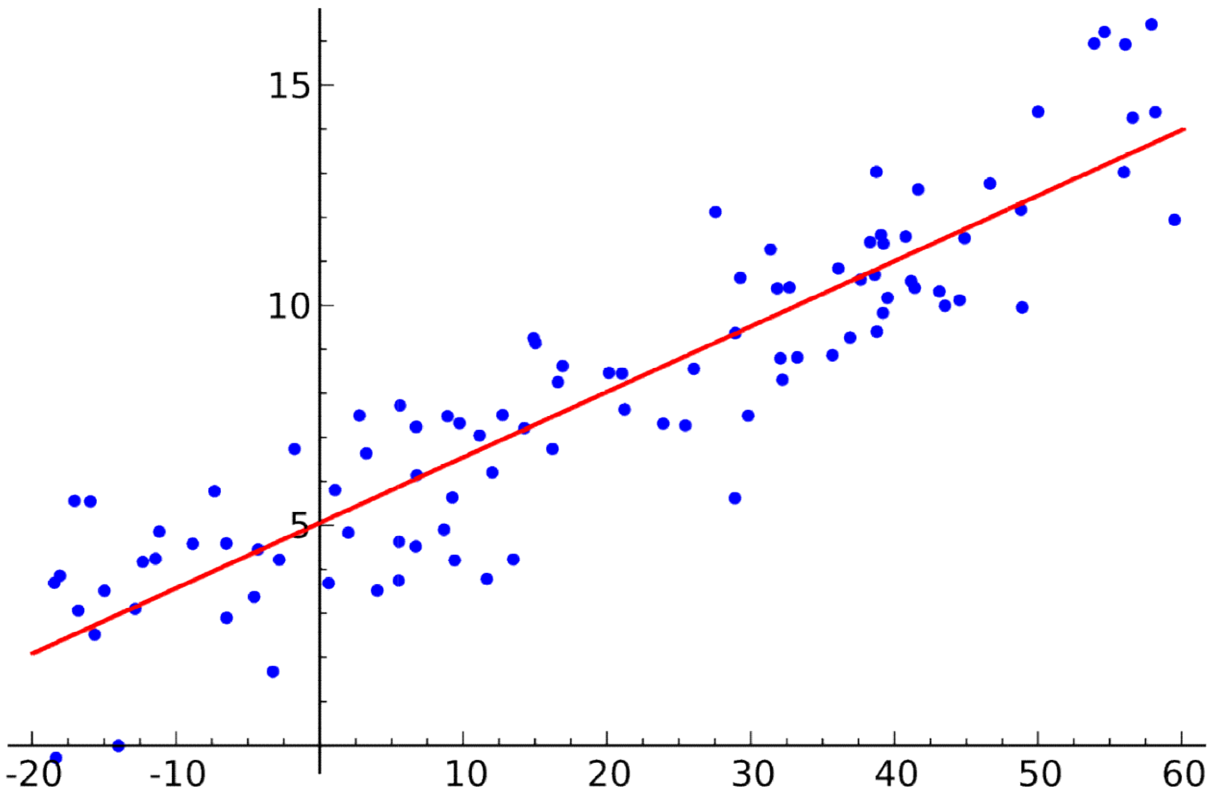
rf_model = randomForest(cnt~., data = train,importance=TRUE, ntree = 200)

#Predict the test cases

rf_predictions = predict(rf_model, test[,-6])

Using Linear Regression Machine Learning Algorithm

In statistics, **linear regression** is a **linear** approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple **linear regression**.



lr_model=lm(formula = cnt~., data = train)

Check the summary of the model

summary(lr_model)

#Multiple R-squared: 0.5517,   Adjusted R-squared: 0.5453

#F-statistic: 86.9 on 8 and 565 DF, p-value: < 2.2e-16

#Predict the test cases

lr_predictions = predict(lr_model, test[,-6])

CHAPTER 3

CONCLUSION

Calculate MAPE Mean Absolute Percentage Error Loss

(DECISION TREE)

```
MAPE = function(actual, pred){

print(mean(abs((actual - pred)/actual)) * 100)

}

MAPE(test[,6],bike_predictions)

# MAPE 26.05408 %

# ACCURACY 73.94%

# MAE 1018.3953691

# RMSE 1246.8818104


regr.eval(test[,6],bike_predictions, stats = c("mae","rmse","mape"))
```

(Random Forest)

```
rf_predictions = predict(rf_model, test[,-6])

MAPE(test[,6],rf_predictions)


regr.eval(test[,6],rf_predictions, stats = c("mae","rmse","mape"))
```

MAPE 27.40%   ACCURACY 72.60%   MAE 1004.6482397 RMSE 1142.0250063

(Linear regression)

regr.eval(trues = test[,6], preds = lr_predictions, stats = c("mae","rmse","mape"))

MAPE(test[,6], lr_predictions)

#MAPE 25.28217%

#ACCURACY 74.72%