Employee Absenteeism

Aaron Rebello

13 december 2018

Contents

Chapter 1

Introduction

1.1 Problem Statement

Employee Absenteeism is the absence of an employee from work. Its a major problem faced by almost all employers of today. Employees are absent from work and thus the work suffers. Absenteeism of employees from work leads to back logs, piling of work and thus work delay.
XYZ is a courier company. As we appreciate that human capital plays an important role in collection, transportation and delivery. The company is passing through genuine issue of Absenteeism. The company has shared it dataset and requested to have an answer on the following areas:
1. What changes company should bring to reduce the number of absenteeism?
2. How much losses every month can we project in 2011 if same trend of absenteeism continues?

1.2 Data

Our task is to build Regression models which will predict the absenteeism depending on multiple employee characteristics. Given below is a sample of the data set that we are using to predict the absenteeism of employee:
As you can see in the table below we have the following 11 variables, using which we have to correctly predict the quality of the wines:

Dataset Details: Dataset
Characteristics: Timeseries Multivariant
Number of Attributes: 21
Missing Values : Yes

Attribute Information:
1. Individual identification (ID)
2. Reason for absence (ICD).
Absences attested by the International Code of Diseases (ICD) stratified into 21 categories
(I to XXI) as follows:
I Certain infectious and parasitic diseases
II Neoplasms
III Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
IV Endocrine, nutritional and metabolic diseases
V Mental and behavioural disorders
VI Diseases of the nervous system
VII Diseases of the eye and adnexa
VIII Diseases of the ear and mastoid process
IX Diseases of the circulatory system
X Diseases of the respiratory system
XI Diseases of the digestive system
XII Diseases of the skin and subcutaneous tissue
XIII Diseases of the musculoskeletal system and connective tissue
XIV Diseases of the genitourinary system
XV Pregnancy, childbirth and the puerperium
XVI Certain conditions originating in the perinatal period

XVII Congenital malformations, deformations and chromosomal abnormalities
XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX Injury, poisoning and certain other consequences of external causes
XX External causes of morbidity and mortality
XXI Factors influencing health status and contact with health services. And 7 categories without (CID) patient follow-up (22), medical consultation (23), blood donation (24), laboratory examination (25), unjustified absence (26), physiotherapy (27), dental consultation (28).

3. Month of absence
4. Day of the week (Monday (2), Tuesday (3), Wednesday (4), Thursday (5), Friday (6))
5. Seasons (summer (1), autumn (2), winter (3), spring (4))
6. Transportation expense
7. Distance from Residence to Work (kilometers)
8. Service time
9. Age
10. Work load Average/day
11. Hit target
12. Disciplinary failure (yes=1; no=0)
13. Education (high school (1), graduate (2), postgraduate (3), master and doctor (4))
14. Son (number of children)
15. Social drinker (yes=1; no=0)
16. Social smoker (yes=1; no=0)
17. Pet (number of pet)
18. Weight
19. Height
20. Body mass index
21. Absenteeism time in hours (target)

Chapter 2

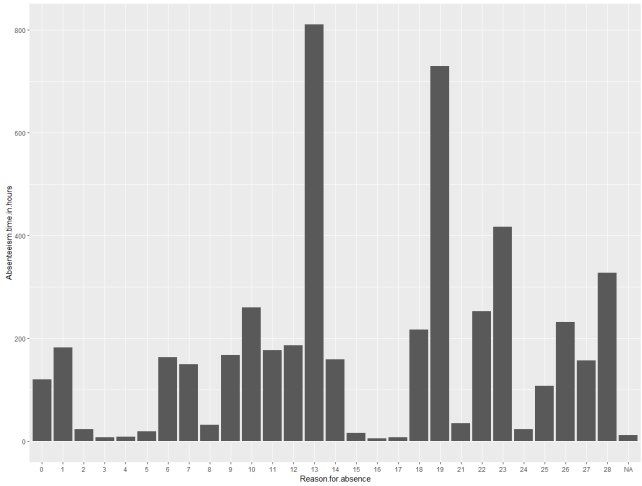
Methodology

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**, followed by preproceesing.

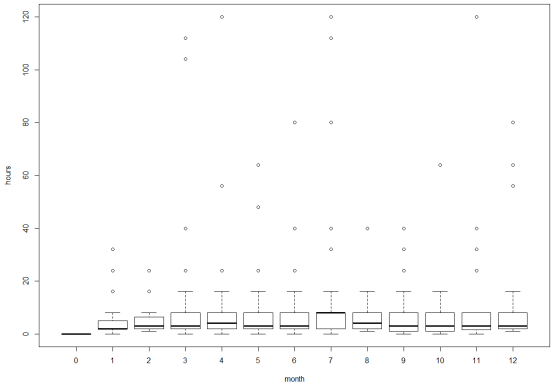
2.1.1 Univariate and Bivariate analysis:

1)Let us see if our hypothesis of reason of absence affecting abseentism :-



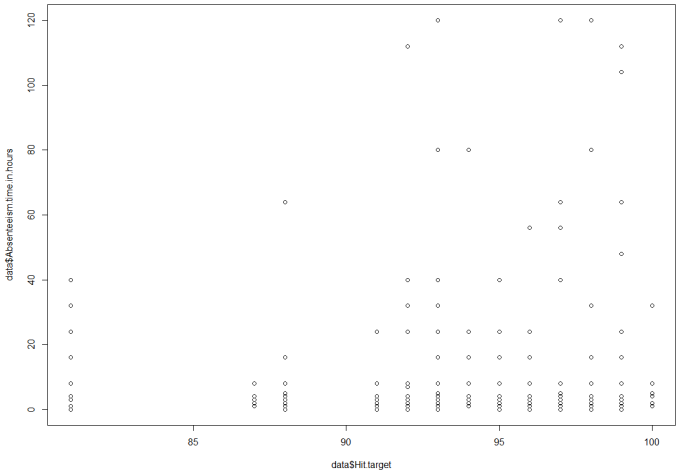
From the above fig we see that, main reason for abseentism is **13(Diseases of the musculoskeletal system and connective tissue)** which means that the employee might have to do heavy work which results in connective tisues or bone issues. So, there should be a bone checkup and first aid maintained for the employeers
Another reason is **19(Injury, poisoning and certain other consequences of external causes)** which means same that because of work the employee are suffering and there are not enough medication or aid maintained to heal them. So, the employee has to take external treatment which results in absenteeism.
So, the workload should be reduced and also a team should be to look after the health if the employee gets any injury and should be treated.

2)Let us see if month causes any effect on absenteeism:-



From the above fig, we can infer that the winter time i.e. January, february has least absenteeism

3) let us plot a scatter plot between hit target and absenteeism:-



We see that, more number of absenteeism comes under 90-100, which means that once an employee hits their target, they remain absent.
So, there should a system assigning a work once the employee finishes its assigned work. So, that he doesn't have to be absent thinking that he has no target to complete.

2.1.2 Missing value Analysis and Outlier Analysis

Missing values occur when no data value is stored for the variable in an observation. A missing value can signify a number of different things in your data. Perhaps the data was not available or not applicable or the event did not happen. It could be that the person who entered the data did not know the right value, or missed filling in. Missing values are a common occurrence, and you need to have a strategy for treating them. Typically, ignore the missing values, or exclude any records containing missing values, or replace missing values with the mean, or infer missing values from existing values. We check for missing values in our data. We saw that the target variable has most number of missing value which seems to be less than 30% of the total data. And since it the target data we need to impute data. Here, we use KNN imputation and impute the data

Now, we plotted box plot to see if there are any outliers in the data. An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. There are numerous impacts of outliers in the data set. It increases the error variance and reduces the power of statistical tests. If the outliers are non-randomly distributed, they can decrease normality. They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions. After plotting box plot, we got the figures as below:-

From the figure 1,2,3 we can see that height has most outliers followed by some other variables.
So, we replace them by NA and later impute this missing values using KNN imputation.

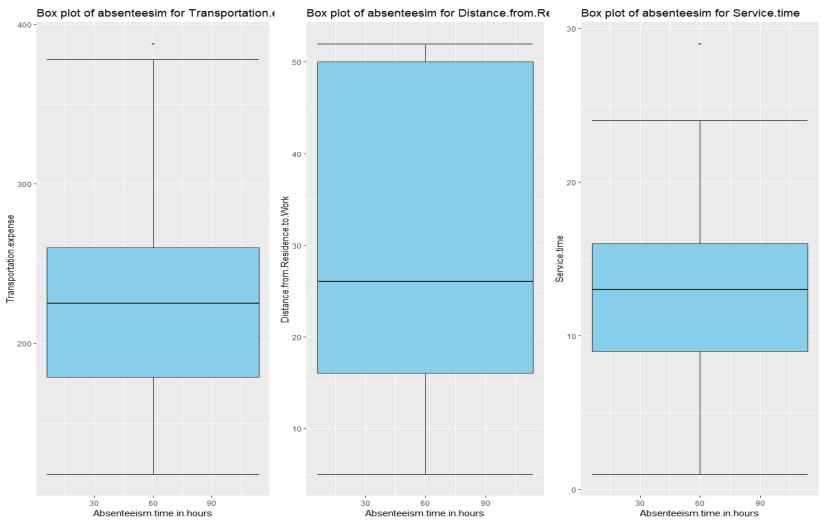


Fig 1

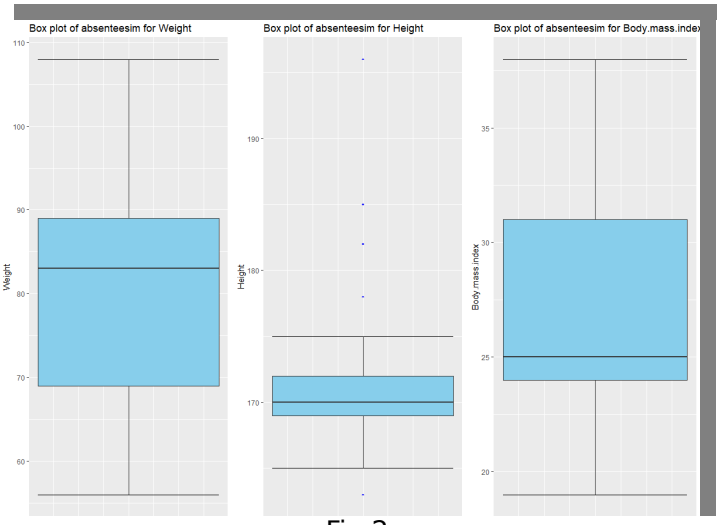


Fig 2

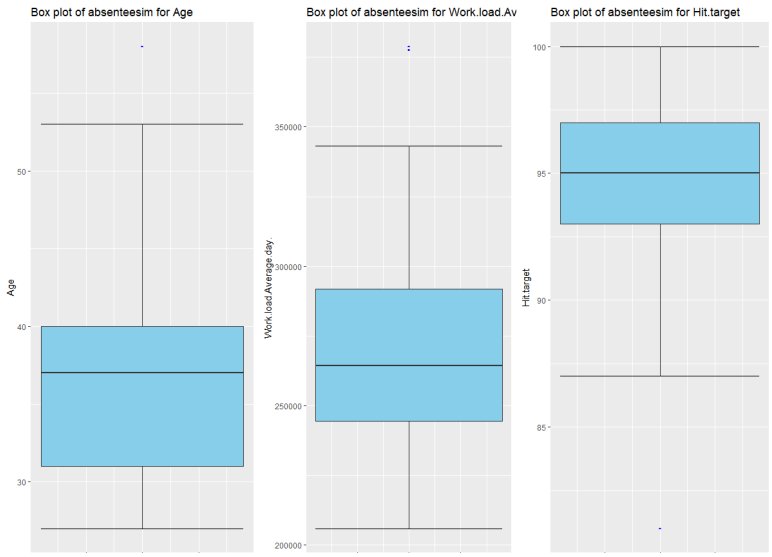
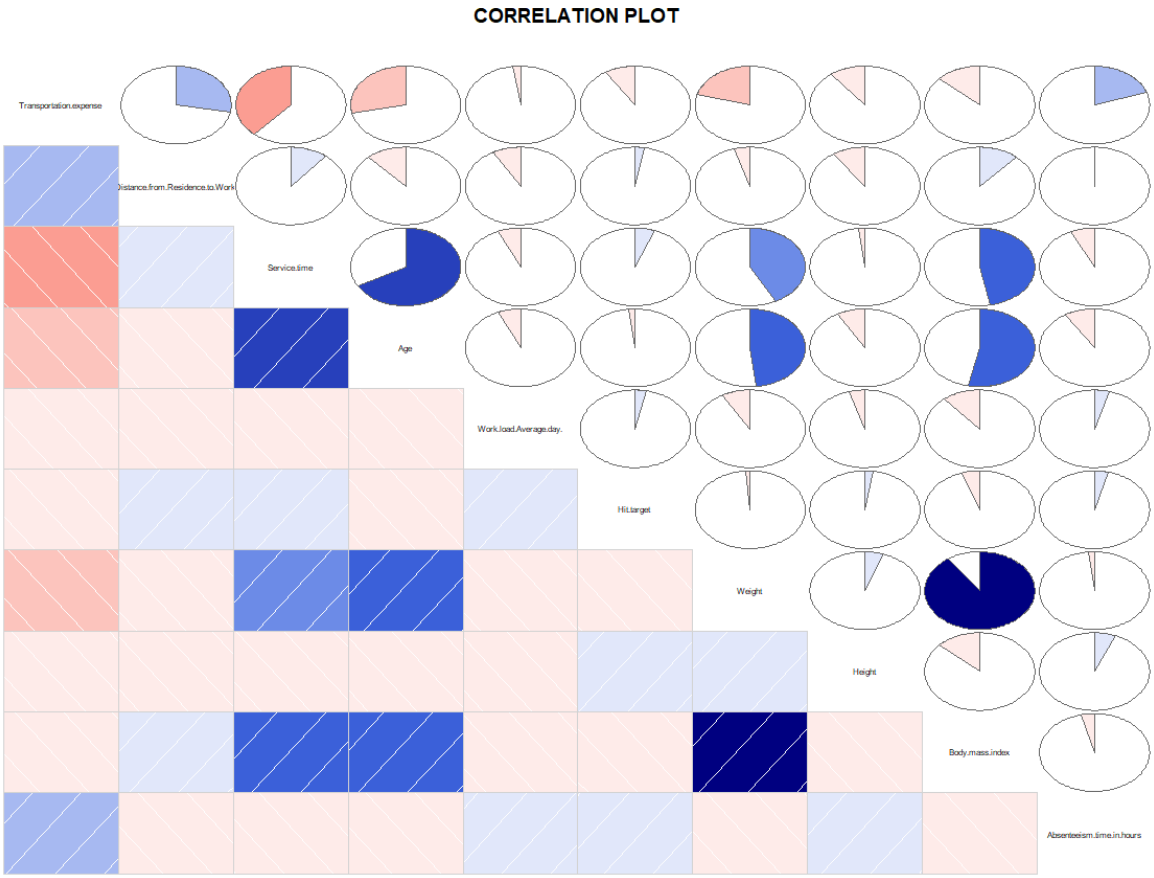


Fig 3

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. We are using correlation matrix to identify most correlated numeric variables and anova test for categorical variables.

2.1.3.1 Correlation Matrix



From the above figure we can see that weight is highly correlated to body mass index. Also, service time is slightly correlated to age. So, we drop weight and service time.

2.1.3.2 Anova Test

```
summary(anova_test)
                  Df Sum Sq Mean Sq F value Pr(>F)
ID                 35   1385   39.57   5.908 <2e-16 ***
Reason.for.absence 27   2397   88.76  13.252 <2e-16 ***
Month.of.absence   12     78    6.49   0.970  0.477
Day.of.the.week     4     14    3.62   0.541  0.706
Seasons             3     19    6.26   0.934  0.424
Disciplinary.failure  1     15   14.60   2.180  0.140
Education           1      0    0.17   0.025  0.875
Son                 3      7    2.24   0.335  0.800
Social.drinker      1      8    7.88   1.177  0.278
Pet                 1      1    0.77   0.116  0.734
Residuals         651   4361    6.70
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

After performing anova test, we see that reason for absence and ID is less than 0.05 so we consider these variables but ID does not explain much about the target variable so we consisder only reason for absence and remove all other categorical vaiables.

2.1.4 Feature Scaling

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing steps. If training an algorithm using different features and some of them are off the scale in their magnitude, then the results might be dominated by them. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. We use normalization here for feature scaling.
Normalization brings all of the variables into proportion with one another. It transforms data into a range between 0 and 1. We have to see the variables that are scattered highly and apply normalization. We normalize the following variables in our data so that we can process to the modeling phase. Normality check for variables is in appendix
Formulae used for normalization is

$$Value_{new} = \frac{Value - minValue}{maxValue - minValue}$$

So, now as our data is pre processed, we are ready to feed it to the models.

2.2 Modeling

As we know that our model is regression model,we first divide the data into train and test and then apply the train data on the following models, which gives us the metrics.

2.2.1 Decision tree

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.

2.2.2 Random Forest

The Random Forest is one of the most effective machine learning models for predictive analytics, making it an industrial workhorse for machine learning.

2.2.3 Linear Regression

In statistics, linear regression is a linear approach to modelling 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.

The code for models is in the appendix .

Chapter 3

Conclusion

3.1 Model Evaluation

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 RMSE value

RMSE: Root Mean Square Error is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit Lower the value of RMSE the better the model.

1) **Decision tree**

After implementing the model we get the rmse value as #RMSE=0.1958891

2) **Random Forest**

After implementing random forest model we get the value as #RMSE=0.1783666

3) **Linear Regression**

After implementing linear regression model we get the value as #RMSE=0.18

3.2 Model Selection

We can see that Random Forest has the lowest value of RMSE i.e. 0.17837

3.3 Answers to the problem statement

1. What changes company should bring to reduce the number of absenteeism?

-> As we saw from the analysis that the main reason for absence is , main reason for abseentism is **13(Diseases of the musculoskeletal system and connective tissue)** which means that the employee might have to do heavy work which results in connective tisues or bone issues. So, there should be a bone checkup and first aid maintained for the employeers
Another reason is **19(Injury, poisoning and certain other consequences of external causes)** which means same that because of work the employee are suffering and there are not enough medication or aid maintained to heal them. So, the employee has to take external treatment which results in absenteeism.
So, the workload should be reduced and also a team should be to look after the health if the employee gets any injury and should be treated.
Also, We see that, more number of absenteeism comes under 90-100, which means that once an employee hits their target, they remain absent.
So, there should a system assigning a work once the employee finishes its assigned work. So, that he doesn't have to be absent thinking that he has no target to complete.

Appendix A – R code

```
rm(list=ls())

setwd("C:/Users/ARON/Desktop/edwisor projects/employee absenteesim")
getwd()

Load Libraries
# x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies", "e1071",
"Information",
#       "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees')
#
# install.packages(x)
# lapply(x, require, character.only = TRUE)
# rm(x)

library(xlsx)
library(rlang)
library(ggplot2)


data_original=read.xlsx("Absenteeism_at_work_Project (6).xlsx",sheetIndex = 1, header = TRUE)
data=data_original

########analysing dataset#################################################

#let us sse the structure
str(data)

colnames(data)
dim(data)

class(data)
```

```
#no of unique values in each variables


apply(data, 2,function(x) length(table(x)))

# we can see that the number if ID's are 36, which means we have data of 36 employees

###understanding the unique values and depending on which converting the categorical interger into factor
unique(data$ID)
data$ID=as.factor(as.integer( as.character(data$ID)))

unique(data$Reason.for.absence)
#we see that there are 28 categories so we convert this num into factor
data$Reason.for.absence=as.factor(as.integer( data$Reason.for.absence))
#data$Month.of.absence[data$Reason.for.absence %in%"0"]= NA



unique(data$Month.of.absence)
#we see that there are 12 months so it should be categorised
data$Month.of.absence=as.factor(as.integer(data$Month.of.absence))
#data$Month.of.absence[data$Month.of.absence %in%"0"]= NA

unique(data$Day.of.the.week)
#we have 5 days given in week so it should be categorised
data$Day.of.the.week=as.factor(as.character(data$Day.of.the.week))

unique(data$Seasons)
#we have been given 4 seasons so we need to convert it into factor
data$Seasons=as.factor(as.character(data$Seasons))


unique(data$Transportation.expense)

unique(data$Distance.from.Residence.to.Work)

unique(data$Disciplinary.failure)
#we have two categories, so we need to convert it into categorical
data$Disciplinary.failure=as.factor(as.character(data$Disciplinary.failure))


unique(data$Education)
#we have four categories, so we need to convert it into categorical
data$Education=as.factor(as.character( data$Education))

unique(data$Son)
data$Son=as.factor(as.character(data$Son))



unique(data$Social.drinker)
#we have two categories, so we need to convert it into categorical
data$Social.drinker  =as.factor(as.character(data$Social.drinker))

unique(data$Social.smoker)
#we have two categories, so we need to convert it into categorical
data$Social.smoker=as.factor(as.character(data$Social.smoker))

unique(data$Pet)
#we have categories, so we need to convert it into categorical
data$Pet=as.factor(as.character(data$Pet))

unique(data$Absenteeism.time.in.hours)


unique(data$Work.load.Average.day)


#now the dtype has been changed, so lets look at the structure of the data
```

```
str(data)


# #####analysing data (univariate)############################
#
round(prop.table(table(data$Absenteeism.time.in.hours))*100,2)
#we see that the percentage of employee remaining absenteesim is more between 0-24 hours



library(ggplot2)
ggplot(data = data,aes(x =Absenteeism.time.in.hours))+
geom_bar() +  labs(y='', title = ' ')




boxplot(data$Absenteeism.time.in.hours~data$ID,xlab="id",ylab="hours",mail="emplyess remainig absent")
#we see that employee id 9 has been absent for most of the time

boxplot(data$Absenteeism.time.in.hours~data$ID,xlab="id",ylab="hours",mail="emplyess remainig absent")


ggplot(data=data, aes(x=Reason.for.absence, y=Absenteeism.time.in.hours)) + geom_bar(stat="Identity")


#bivariate

#let us check if the reason of absense cause any effect on the absenteesim using boxplot

library(ggplot2)
ggplot(data = data,aes(x =Reason.for.absence))+
  geom_bar() +  labs(y='', title = ' ')
ggplot(data=data, aes(x=Reason.for.absence, y=Absenteeism.time.in.hours)) + geom_bar(stat="Identity")
#13,19,23,28 are the main reasons for absenteesim


library(ggplot2)
ggplot(data = data,aes(x =Month.of.absence))+
  geom_bar() +  labs(y='', title = ' ')
boxplot(data$Absenteeism.time.in.hours~data$Month.of.absence,xlab="month",ylab="hours",mail="hours vs
month")
#january has less absentee

library(ggplot2)
ggplot(data = data,aes(x =Day.of.the.week))+
  geom_bar() +  labs(y='', title = ' ')
boxplot(data$Absenteeism.time.in.hours~data$Day.of.the.week,xlab="day",ylab="hours",mail="hours vs day")
#day doesnt explain beacuse it is uniformly distributed


ggplot(data = data,aes(x =Seasons))+
  geom_bar() +  labs(y='', title = ' ')
  boxplot(data$Absenteeism.time.in.hours~data$Seasons,xlab="Seasons",ylab="hours",mail="hours vs Seasons")
#uniformly distributed


ggplot(data = data,aes(x =Disciplinary.failure))+
  geom_bar() +  labs(y='', title = ' ')
boxplot(data$Absenteeism.time.in.hours~data$Disciplinary.failure,xlab="Disciplinary.failure",ylab="hours",mail="
hours vs df")
#we see that the disciplinary failure can cause 0 absenteesim

boxplot(data$Absenteeism.time.in.hours~data$Education,xlab="Education",ylab="hours",mail="hours vs
Education")

boxplot(data$Absenteeism.time.in.hours~data$Son,xlab="Son",ylab="hours",mail="hours vs Son")
```

```
 boxplot(data$Absenteeism.time.in.hours~data$Social.drinker,xlab="Social.drinker",ylab="hours",mail="hours vs Social.drinker")

boxplot(data$Absenteeism.time.in.hours~data$Social.smoker,xlab="Social.smoker",ylab="hours",mail="hours vs Social.smoker")

plot(data$Transportation.expense,data$Absenteeism.time.in.hours)

plot(data$Distance.from.Residence.to.Work,data$Absenteeism.time.in.hours)

plot(data$Age,data$Absenteeism.time.in.hours)

plot(data$Service.time,data$Absenteeism.time.in.hours)
#service time 5-20 hrs has more absenteesim

plot(data$Hit.target,data$Absenteeism.time.in.hours)
#more number of abseentism has hit more target. so the employee who hit thier target before time might stay absent

plot(data$Body.mass.index,data$Absenteeism.time.in.hours)

plot(data$Pet,data$Absenteeism.time.in.hours)

plot(data$Body.mass.index,data$Absenteeism.time.in.hours)









######EDA

#getting all numeric varaibles together
num_index = sapply(data, is.numeric)
num_data = data[,num_index]
num_col = colnames(num_data) #storing all the column name

#getting all categorical variables together

cat_ind=sapply(data, is.factor)
cat_data=data[,cat_ind]
cat_col= colnames(cat_data)

str(data)

num_col
cat_col


####### missing value analysis and outlier analysis###############


#checking missing value
apply(data,2,function(x){sum(is.na(x))})

library(DMwR)
library(lattice)
library(grid)

# missing_val = data.frame(apply(data,2,function(x){sum(is.na(x))}))
# missing_val$Columns = row.names(missing_val)
# names(missing_val)[1] =  "Missing_percentage"
# missing_val$Missing_percentage = (missing_val$Missing_percentage/nrow(data)) * 100
# missing_val = missing_val[order(-missing_val$Missing_percentage),]
# row.names(missing_val) = NULL
```

```
#data=knnImputation(data,k=3)

#let us first check outliers

library(ggplot2)

   for (i in 1:length(num_col))
{
 assign(paste0("gn",i),
       ggplot(aes_string(y = (num_col[i]), x = 'Absenteeism.time.in.hours'),data = data) +
         stat_boxplot(geom = "errorbar", width = 0.5) +
         geom_boxplot(outlier.colour="blue", fill = "skyblue",
                      outlier.shape=18,outlier.size=1, notch=FALSE) +
         labs(y=num_col[i],x="Absenteeism.time.in.hours")+
         ggtitle(paste("Box plot of absenteesim for",num_col[i])))
}

#gn1-gn11 are all the numerical columns

## Plotting plots together


gridExtra::grid.arrange(gn1,gn2,gn3,ncol=3)


gridExtra::grid.arrange(gn4,gn5,gn6,ncol=3)


gridExtra::grid.arrange(gn7,gn8,gn9,ncol=3)

gridExtra::grid.arrange(gn10,ncol=1)







#we see that some variables has got outliers let us remove them

# #Removing oulier by replacing with NA and then impute
for(i in num_col){
  print(i)
  outv = data[,i][data[,i] %in% boxplot.stats(data[,i])$out]
  print(length(outv))
  data[,i][data[,i] %in% outv] = NA
}
#
# #checking all the missing values
 library(DMwR)
  sum(is.na(data))
data = knnImputation(data, k=3)  #as it gives error so we going via mean or median

# let us check missing values left
apply(data,2,function(x){sum(is.na(x))})
dim(data)


#######feature selection
  library(corrgram)


  corrgram(data[,num_index],
           order = F,  #we don't want to reorder
           upper.panel=panel.pie,
           lower.panel=panel.shade,
           text.panel=panel.txt,
```

```
        main = 'CORRELATION PLOT')
 #We can see var the highly corr related var in plot marked dark blue.
 #Dark blue color means highly positive cor related
 # We se that service.time is highly correlated with age so we remove service time
 # Also, weight is highly correlated to body mass index so we remove weight



 ##---------anova ----------------------------------

 colnames(cat_data)


 #Anova test
 library("lsr")

anova_test=aov(Absenteeism.time.in.hours~ID+Reason.for.absence+Month.of.absence+Day.of.the.week+Seasons+
               Disciplinary.failure+Education+Son+Social.drinker+Social.smoker+Pet,data = data)


 summary(anova_test)

  ##------------------Removing Highly Corelated and Independent var----------------------
 data = subset(data, select = -c(Weight,Day.of.the.week,Seasons,
                                 Disciplinary.failure,Education,Son,Social.drinker,Social.smoker,Pet))

colnames(data)
str(data)

 #####feature scaling

 #Checking Data of Continuous Variable
num_index = sapply(data, is.numeric)
num_data = data[,num_index]
num_col = colnames(num_data)

 ################  Histogram   ###################
qqnorm(data$Transportation.expense)
hist(data$Transportation.expense)

#normalization

 for (i in num_col){
  print(i)
  data[,i]=(data[,i]-min(data[,i]))/(max(data[,i]-min(data[,i])))
}


 #Most of the data is uniformally distributed

 #Using data Standardization/Z-Score here
 # for(i in num_col){
 #   print(i)
 #   data[,i] = (data[,i] - mean(data[,i]))/sd(data[,i])
 # }


str(data)

############model development####

####decision tree########

library(MASS)
library(rpart)

train_index= sample(1:nrow(data),0.6*nrow(data))
train= data[train_index,]
test= data[-train_index,]
```

```
regression=rpart(Absenteeism.time.in.hours ~.,data=train,method="anova")

summary(regression)

reg_predict=predict(regression,test[,-12])


#evaluate
View(test[,12])

#install.packages("DMwR")

library(DMwR)
regr.eval(test[,12],reg_predict,stats = c("mae","mape","rmse"))

# rmse=0.19

####random forest######

library(randomForest)

rf_model= randomForest(Absenteeism.time.in.hours~.,train,importance=TRUE,ntree=100)

summary(rf_model)

rf_predict=predict(rf_model,test[,-12])


regr.eval(test[,12],rf_predict,stats = c("mae","mape","rmse"))

#rmse=0.17

####linear regression#######


library(usdm)


lm_model= lm(Absenteeism.time.in.hours~.,data=train)

summary(lm_model)

lm_predict=predict(lm_model,test[,-12])


regr.eval(test[,19],lm_predict,stats = c("mae","mape","rmse"))

#rmse=3.26







#
###############################################################################################
######
# ###            MODEL BUILDING USING K_FOLD
#
###############################################################################################
######

##Uing k-fold


library(caret)
library(data.table)
##We will use k-fold cross validation method in all the models to be trained below.
```

```
train_control <- trainControl(method="cv", number=5)



#####-----DECISION TREE-----#####

dt_model <- train(Absenteeism.time.in.hours~.,data=data,method="rpart",trControl=train_control)
plot(dt_model)
summary(dt_model)
print(dt_model)
##RMSE is used to select the optimal model using the smallest model .i.e. when cp=0.0801
#RMSE=0.1958891  --  Rsquared=0.1455299  --  MAE=0.1422224

#####-----RANDOM FOREST-----#####

tgrid=expand.grid(.mtry=c(3:8),.splitrule ="variance",.min.node.size = c(5,10,15,20))
rf_model <-
train(Absenteeism.time.in.hours~.,data=data,method="ranger",trControl=train_control,tuneGrid=tgrid,num.tree=200
,importance="permutation")
plot(rf_model)
print(rf_model)
#From the plot above we could see that the optimal parameters are mtry=4, min.node.size=5
#RMSE=0.1783666  --  Rsquared=0.2814073  --  MAE=0.1299943

#####-----LINEAR REGRESSION-----#####

lr_model <- train(Absenteeism.time.in.hours~.,data=data,method="lm",trControl=train_control)
print(lr_model)
summary(lr_model)
##The accuracy of linear regression model is as follows
#RMSE=0.18  --  Rsquared=0.27239244357  --  MAE=0.12

####How much losses every month can we project in 2011 if same trend of absenteeism continues


#2nd PART PREDICTION OF LOSS FOR THE COMPANY IN EACH
#absenty monthwise

#to find loss we require month of absense service time absententy hours and work load

lossdata = subset(data, select = c(Month.of.absence, Service.time, Absenteeism.time.in.hours,
                          Work.load.Average.day.))

#Work loss = ((Work load per day/ service time)* Absenteeism hours)

lossdata["loss"]=with(lossdata,((lossdata[,4]*lossdata[,3])/lossdata[,2]))
for(i in 1:12)
{
  di=lossdata[which(lossdata["Month.of.absence"]==i),]
  print(sum(di$loss))

}


View(lossdata)
```