Project Report On

EMPLOYEE ABSENTEEISM

NIKET RAMESH PATIL

Contents

CHAPTER 1

INRODUCTION

1.1 Problem statement

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

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 behavioral 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)

1.3 Data Understanding

As we can see in the below table we have 21 variables in which 20 variables are independent and 1 variable is dependent & it is also known as target variable. In above data set independent variables have 11 categorical and 9 continuous variables. As we can see our target variable i.e. **Absenteeism time in hours** is a continuous variables so above data belongs to regression problem.

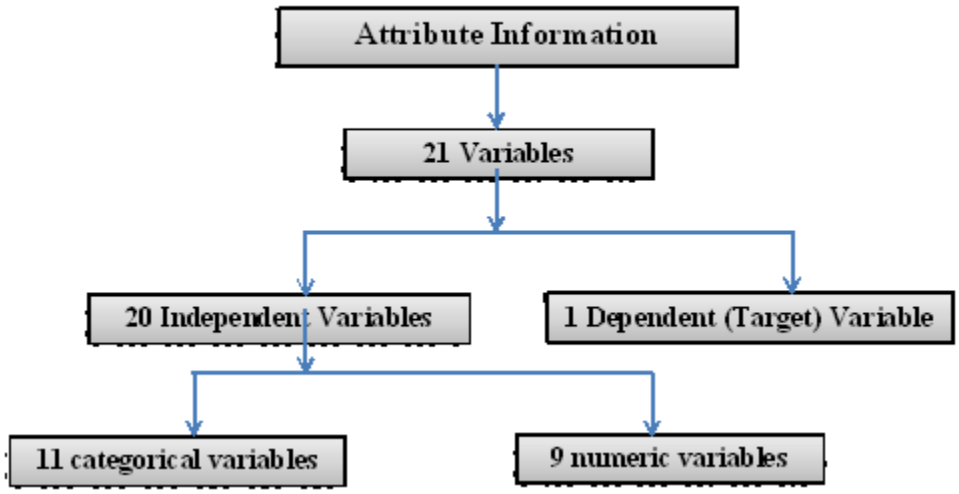


Fig 1.1 Attribute distribution tables

```
ID                                int64
Reason for absence              float64
Month of absence                float64
Day of the week                   int64
Seasons                           int64
Transportation expense          float64
Distance from Residence to Work float64
Service time                    float64
Age                             float64
Work load Average               float64
Hit target                      float64
Disciplinary failure            float64
Education                       float64
Son                             float64
Social drinker                  float64
Social smoker                   float64
Pet                             float64
Weight                          float64
Height                          float64
Body mass index                 float64
Absenteeism time in hours       float64
dtype: object
```

Fig 1.2 Data types of all 21 variables

CHAPTER 2

DATA PREPROCESSING

As we know when we received any data it is in unstructured format we cannot pass the same to our machine learning models because our models only understand the structure data. So, to convert our data into the desire shape we must have pass it from data preprocessing to remove the impurity from the same. As discuss below.

2.1 Missing Value Analysis

Missing value analysis plays a vital role in data preparing. There are many reasons to occur missing values. In statistics while calculating missing values if it is more than 30% we just drop the particular attribute because it does not carry much information to predict our target variables. As we can see in the below Fig. 2.1 highest percentage of missing value is 4.189. So, here we consider all the attributes.

	Variables	Missing_percentage
0	Body mass index	4.189189
1	Absenteeism time in hours	2.972973
2	Height	1.891892
3	Work load Average	1.351351
4	Education	1.351351
5	Transportation expense	0.945946
6	Son	0.810811
7	Disciplinary failure	0.810811
8	Hit target	0.810811
9	Social smoker	0.540541
10	Age	0.405405
11	Reason for absence	0.405405
12	Service time	0.405405
13	Distance from Residence to Work	0.405405
14	Social drinker	0.405405
15	Pet	0.270270
16	Weight	0.135135
17	Month of absence	0.135135
18	Seasons	0.000000

Fig. 2.1 Missing value percentage table

There are many metrologies to remove missing values like mean, median, KNN. So, **here we go for KNN imputation method.**

In below Fig 2.2 we can see the probability distribution function before applying outlines. The attributes are skewed in nature. It means they are not normally distributed.

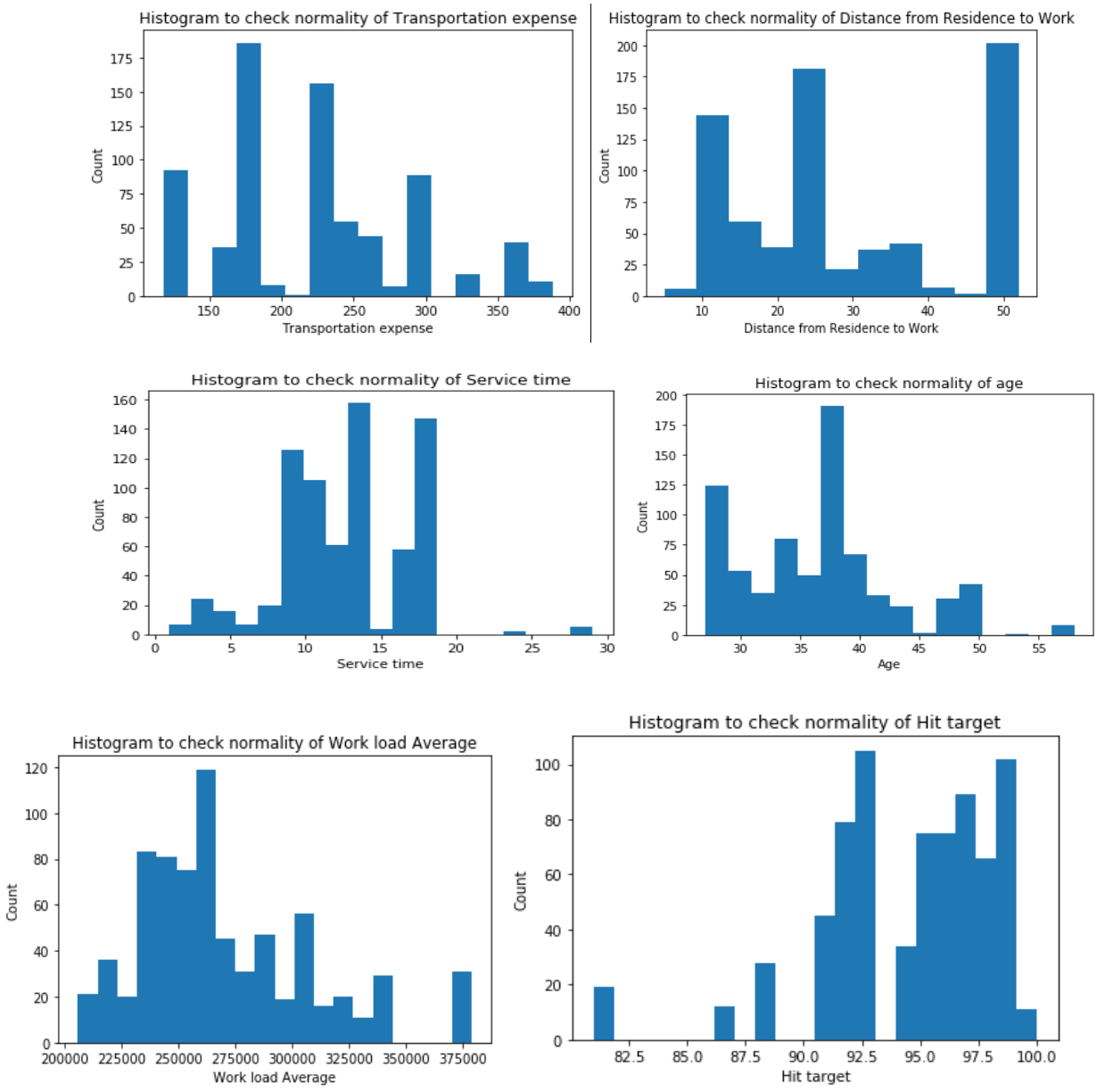
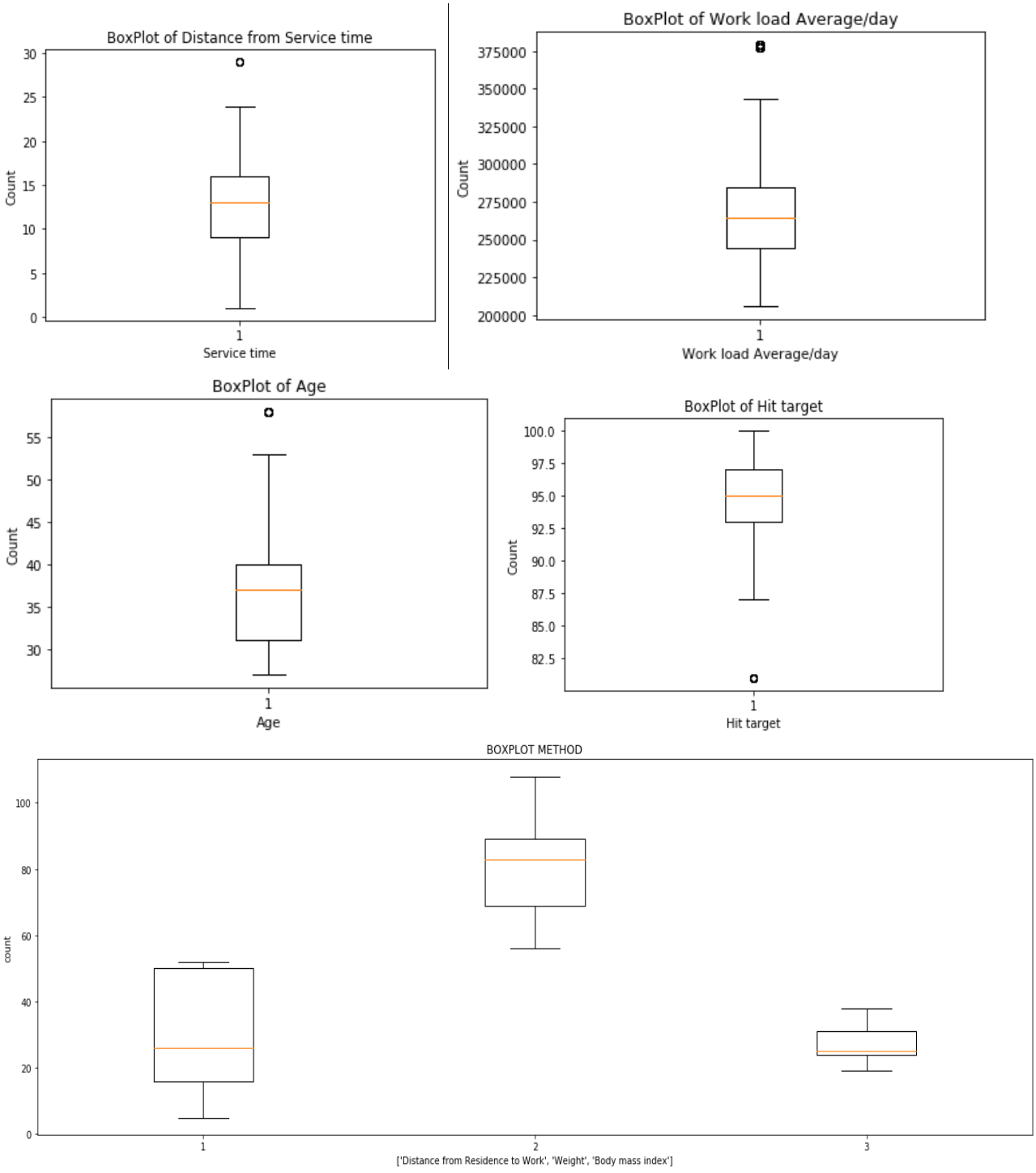


Fig 2.2 To check normality of data

2.2 Outlier Analysis

Outlier is the observation which is inconsistent related with all data set. The values of Outliers are the accurate but it is far away from the set of actual values and it heavily impact on the mean so that we consider it as an outlier. Here we have used box plot method to detect outliers as shown below Fig 2.3.



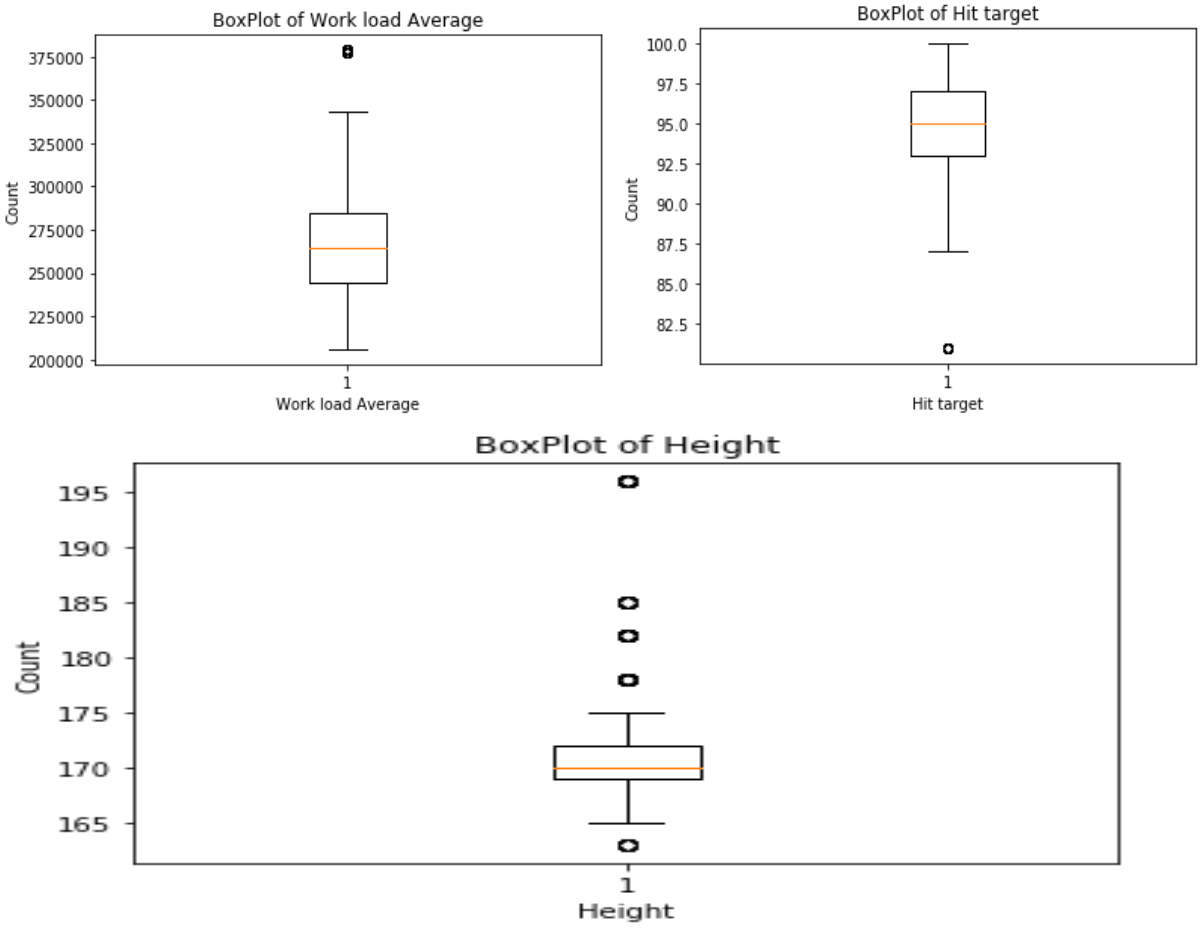


Fig 2.3 Detecting outliers using box plot.

As we can see almost all the variables having outliers except **Distance from residence to work, Weight and Body mass index.** There Two method to remove the outliers direct and KNN. Here we directly remove outlines whose values are above upper 75 percentile and below lower 25 percentiles.

2.3 Feature Selection

As we know, while developing the model if we consider the independent variables which carries the same information to explain the target variables it will affect the problem of multi-collinearity. So to avoid our model from the multi-collinarity problem we need to applied Feature Selection or dimensional reduction on the top of our data set. It helps us to sort out the variables which are highly correlated with each other. In our data set we applied **correlation analysis** for numeric variables and **ANOVA** for the categorical variables

In below Fig.2.4 **weight and body mass index** are highly correlated with each other. It means those variables carry **91%** of same information to explain the target variable. When we applied **ANOVA** test on categorical variables the p value of '**month of absence**' is greater than **0.05** so we reject the null hypothesis and drop the particular variable as shown in **Fig.2.5.**

So, here we drop 2 variables and weight by using Feature Selection techniques.

Fig 2.4 Correlation plot to find the dependency of variables

```
P value for variable ID is 4.306477684811205e-60
P value for variable Reason for absence is 4.535246161888195e-87
P value for variable Month of absence is 0.300131581333309
P value for variable Day of the week is 3.905295006606339e-09
P value for variable Seasons is 6.6063987792330905e-18
P value for variable Disciplinary failure is 1.710062628183748e-40
P value for variable Education is 4.459375563221273e-28
P value for variable Social drinker is 3.622750273781646e-35
P value for variable Social smoker is 2.9883999194395525e-40
P value for variable Pet is 4.559310373902569e-33
P value for variable Son is 1.6788677016060183e-30
```

Fig 2.4 Calculation of P values by using ANOVA test

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Month of	709	4508.973	6.359623	12.28402
Absentee	709	4887.494	6.893504	174.9898

ANOVA

Source of	SS	df	MS	F	P-value	F crit
Between	101.0429	1	101.0429	1.079093	0.299078	3.848034
Within Gr	132589.8	1416	93.63689			
Total	132690.9	1417				

Fig.2.6 ANOVA test using excel for **month of absence**

2.4 Feature Scaling

As we know before passing the data to machine learning algorithm our data should be structure in format. To arrange our data into the desire structure feature scaling technique comes into the picture. In this two methods are playing the important role i.e. normalization and standardization. If our data is normally distributed we go for normalization else for standardization. As we can see in Fig.2.2 our data has skewed in nature. So, here we applied **normalization.**

2.5 Principal Component Analysis

In our project when our data has passed though all preprocessing techniques it is ready for modeling. But, when we used the same data for model development we faced many problems. The model is performing best on my train data but it is not performing well on test data set. It means there is a problem of over-fitting in our model. So, to avoid over-fitting the term PCA comes into the picture. Principal component analysis is a method of selecting important variables from a large set of variables available in a data set. It is also known as dimensional reduction techniques. After Appling dummy variable of categorical variables the shape of our data became **117** columns and **714** observations, more number of columns gives us more impurity. Therefor we have to reduce the dimension of it. From the below graph **Fig 2.7** we

conclude that to get more than 95% of value to explain the target variable here we select 10 variables from 117.

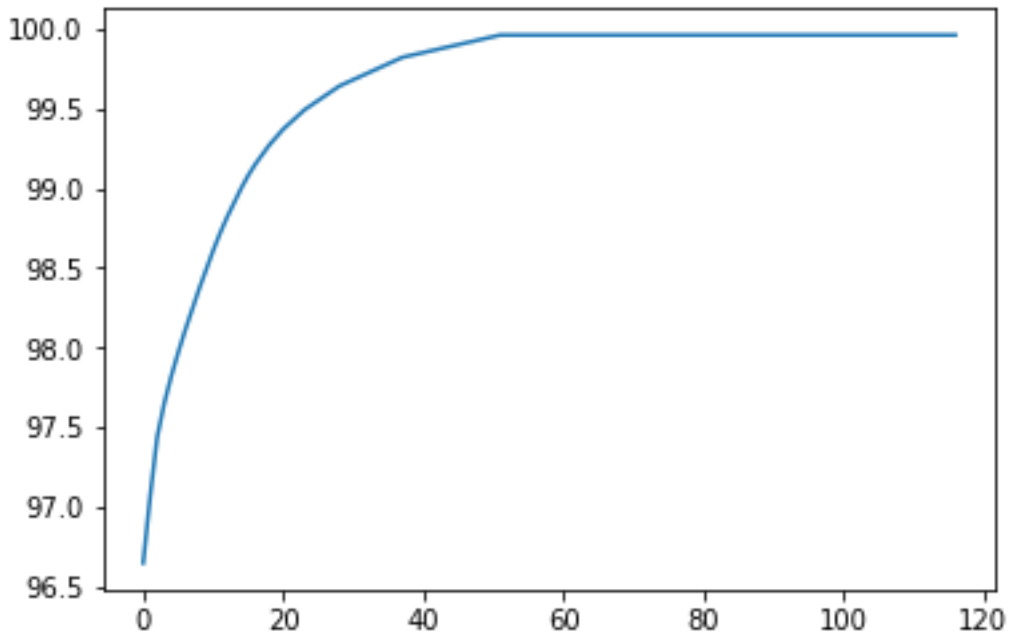


Fig 2.7 PCA graph

CHAPTER 3

Modeling

As we know our target variable is numeric so that here we used regression models on preprocessing data set to predict the target variable.

3.1 Decision Tree

Decision tree is a supervised machine learning algorithm it is applied for regression and classification problem. Decision tree is not only accepted the continuous but also the categorical as independent variables. It is a tree like graph in which each branch connects nodes with "and" & multiple branches are connected by "or" and It is too easy to understand by business user. Below we are calculating RMSE and R^2 before PCA and after PCA.

	Python		R
	Before PCA	**After PCA**	
RMSE Train	12.851874180706115	3.68134298668133	0.4104617
RMSE Test	9.668823710471564	3.70320838447432	0.3629201
R^2 Test	-0.08913222767074	0.917812039695613	0.9838120

3.2 Random Forest

Random forest is the collection of multiple decision trees. While building every single decision tree it selects the random data. It means every single decision tree have different set of data. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. It can handle large no of independent variables without variable deletion and it will give the estimates that what variables are important. The RMSE and R^2mvaalue of our project are shown below.

	Python		R
	Before PCA	**After PCA**	
RMSE Train	5.748489329673611	0.513932453593440	0.3104444
RMSE Test	11.593810930051497	0.247655806947628	0.5246540
R^2 Test	0.04560237818530943	0.999449403746832	0.9700756

3.3 Liner Regression

Linear Regression is one of the statistical methods of prediction. It is applicable only on continuous data. It means the target variables should be continuous in nature. To build any model we have some assumptions to put on data and model. Below we calculated RMSE and R^2 values using liner regression.

	Python		R
	Before PCA	**After PCA**	
RMSE Train	11.340729937074233	2.0770541003584598e-08	0.002999435
RMSE Test	128251.36176736317	2.5002207684027532e-08	0.003919584
R^2 Test	-116788717.04887399	1	0.999999227

3.4 KNN

KNN algorithm is applied for regression and classification problem. It is not storing an y pattern like all algorithms does it just calculate the distance between every test data. So, it takes more time to calculate the same. Therefor the KNN is lazy learning algorithm.

	Python	
	Before PCA	**After PCA**
RMSE Train	11.451786781584904	1.7891982055080662
RMSE Test	12.685707628055733	1.5809981574645886
R^2 Test	-0.1426318269955762	0.9890734422096

3.5 Gradient boosting

Gradient boosting is currently one of the most popular machine learning techniques for efficient modeling of tabular datasets of all sizes. It is a technique which applicable for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. e**X**treme **G**radient **B**oosting.
I.e. XGboost is a very fast.

	Python	R
RMSE Train	0.00900315562275573	0.11334843
RMSE Test	0.7583517760287742	0.2463161
R^2 Test	0.9948372911137446	0.99889710

CHAPTER 4

Conclusion

In above chapters we applied multiple preprocessing to frame our data into the structural format and different machine learning algorithm to check the performance of model. In this chapter we finalize one of them.

4.1 Model Evaluation

In the previous chapter we have applied different algorithms on our dataset and calculate the **Root Mean Square Error (RMSE)** and **R-Squared** Value for all the models. RMSE is the standard deviation of the 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. RMSE is an absolute measure of fit. R-squared is a relative measure of fit. R-squared is basically explains the degree to which input variable explain the variation of the output. In simple words R-squared tells how much variance of dependent variable explained by the independent variable. It is a measure if goodness of fit in regression line. Value of R-squared between 0-1, where 0 means independent variable unable to explain the target variable and 1 means target variable is completely explained by the independent variable. So, Lower values of RMSE and higher value of R-Squared Value indicate better fit of model.

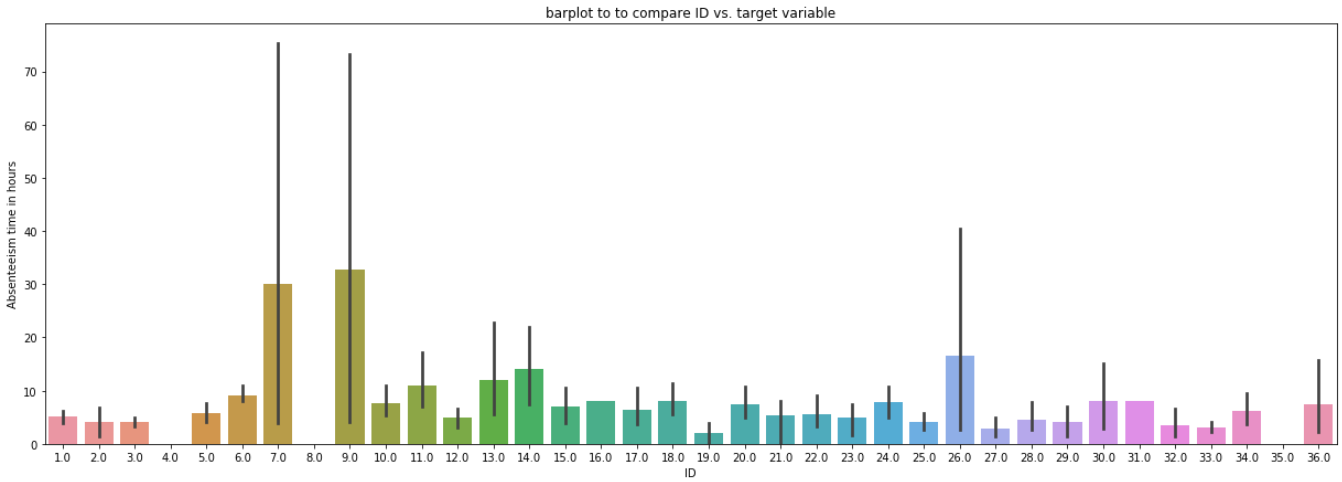
4.2 Model Selection

As we observed on all the model performance the **linear regression** gives us batter out comes as compare to other model. The RMSE of linear regression is less and the value of R^2 is also maximum i.e. 1. The RMSE values difference for train and test data is vary less so there is n o any problem of model over-fitting. So, here we select linear regression is our final model for o ur problem statement.

4.3 Answers of asked questions

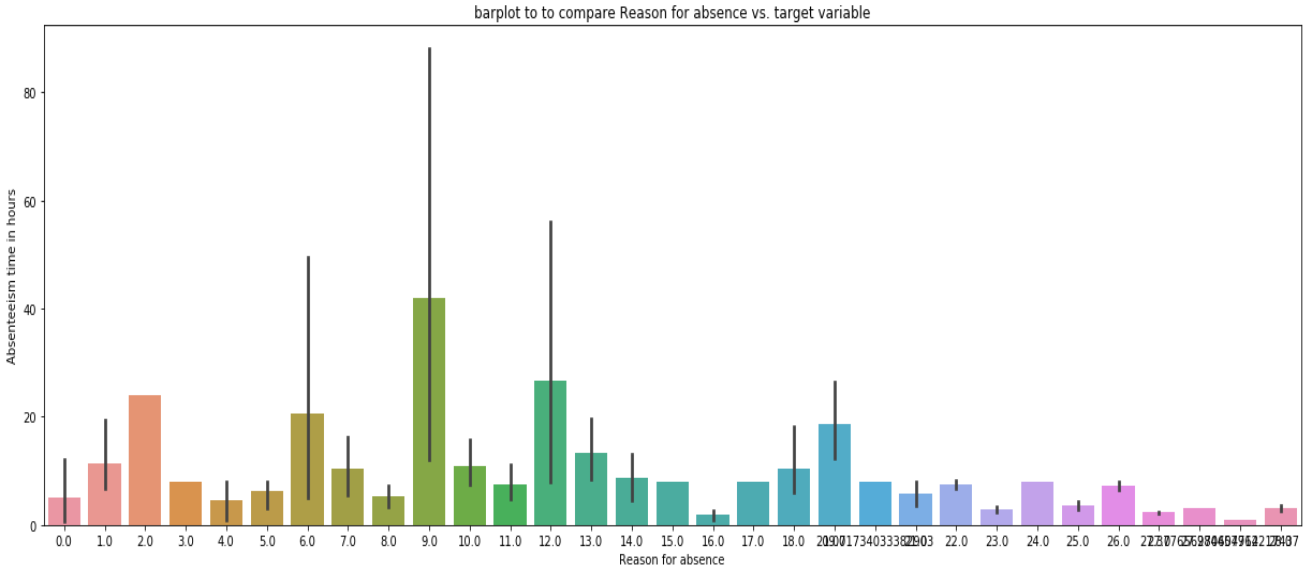
1. **What changes company should bring to reduce the number of absenteeism?**

- **ID**



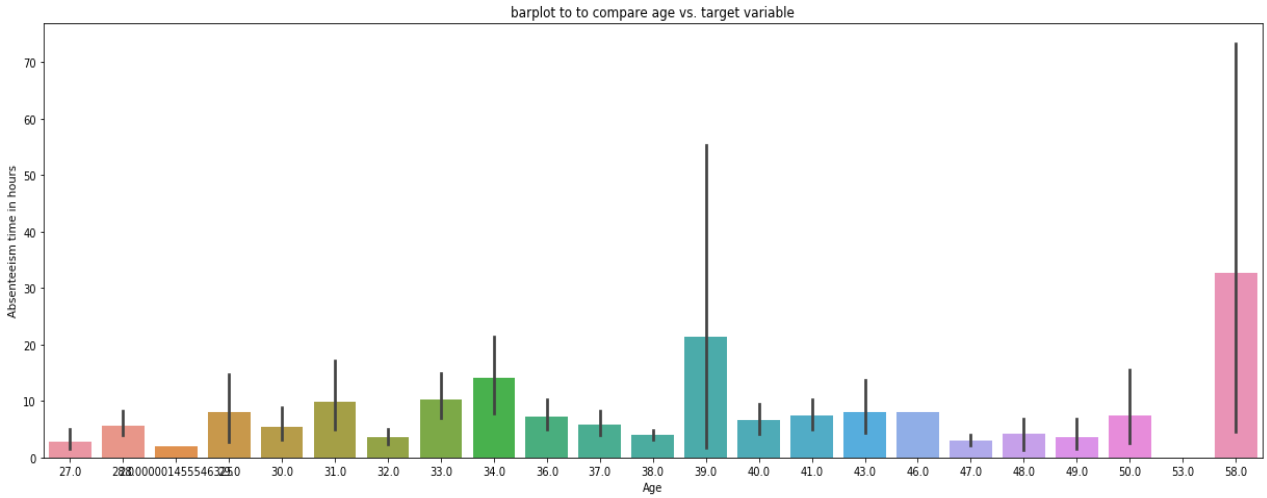
In above plot we have compare ID vs. target variables and here we can see ID 7, 9, 26 are frequently absent. The organization should take an action against them.

- **Reasons of Absent**



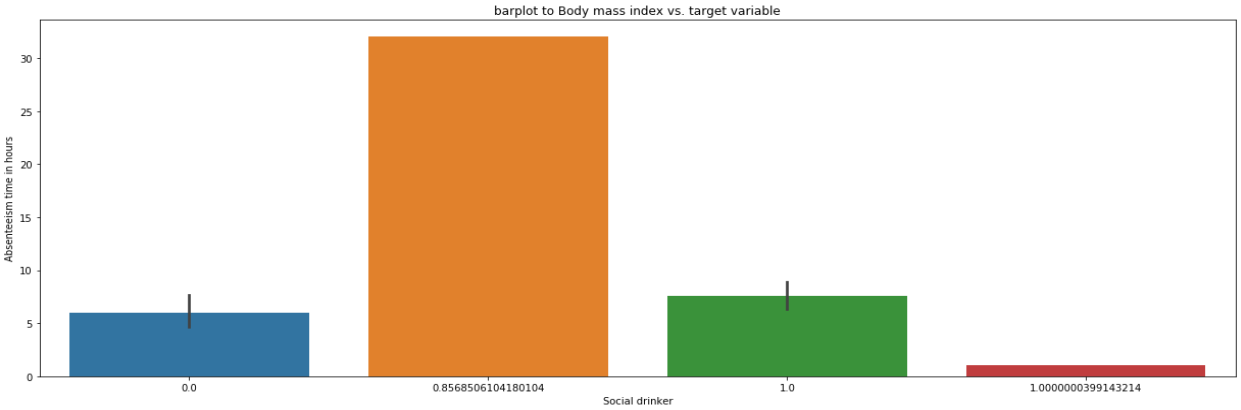
The Most frequent reasons of Absenteeism are Neoplasms, Diseases of the nervous system, Diseases of the circulatory system, Diseases of the skin and subcutaneous tissue. So, the company should think on it or they can conduct the health checkup session for all the employees and try to find the reason for why our workers are suffer from these diseases.

- **Age**



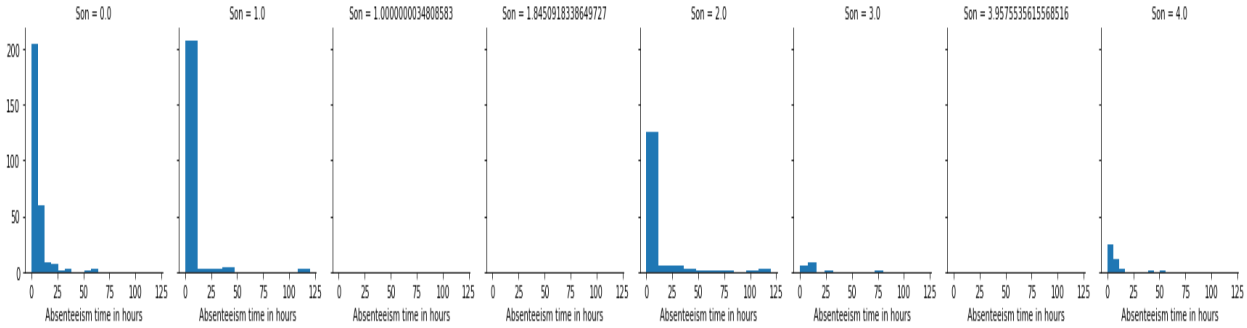
The employee whose age is around 58 they have more absenteeism time.

- **Social Drinker**



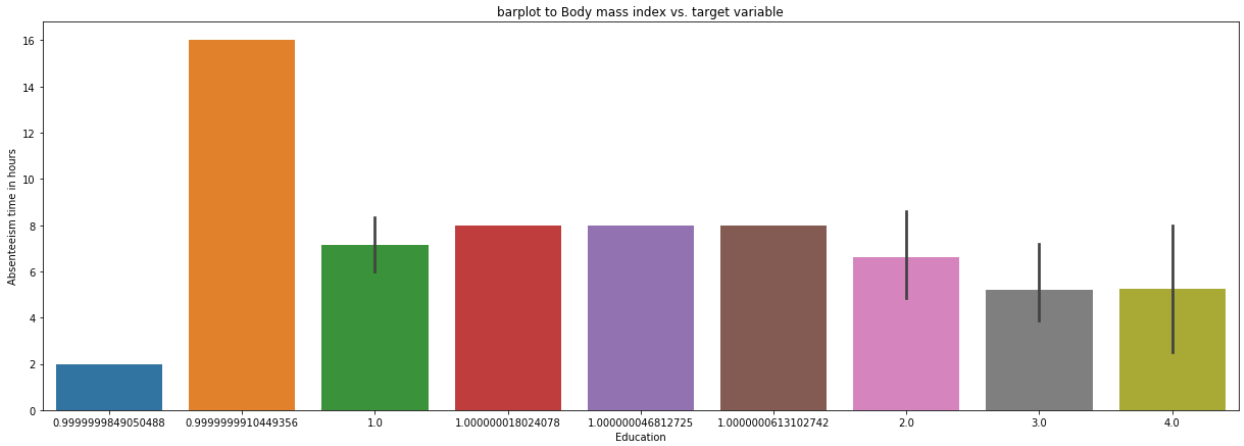
If the worker is social drinker so there is more chance of absenteeism

- **Son**



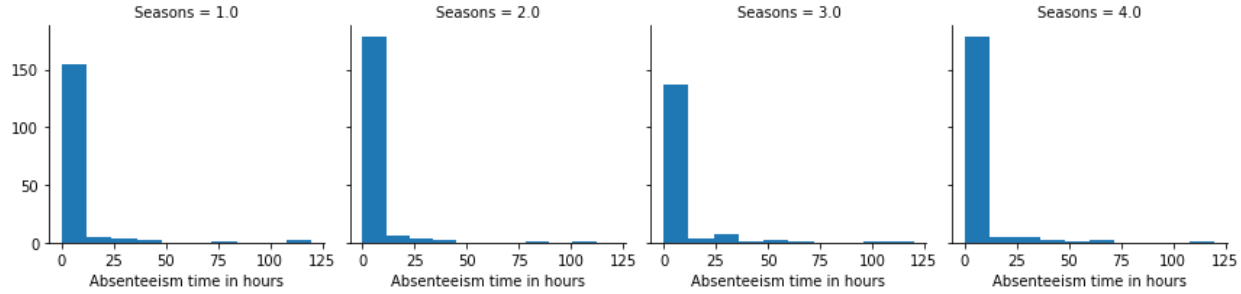
Those employees who have 1 son tend to be absent more. I think because of one of the parson either mother or father should stay at home to pamper there son.

- **Education**



The people those who have completed high school level education tend to be absent more as compare to other people.

- **Seasons**



As we can see there is no huge impact of absenteeism because of seasons.

As mention above these are the main reasons for employee absenteeism company should think on it.

2. How much losses every month can we project in 2011 if same trend of Absenteeism continues?

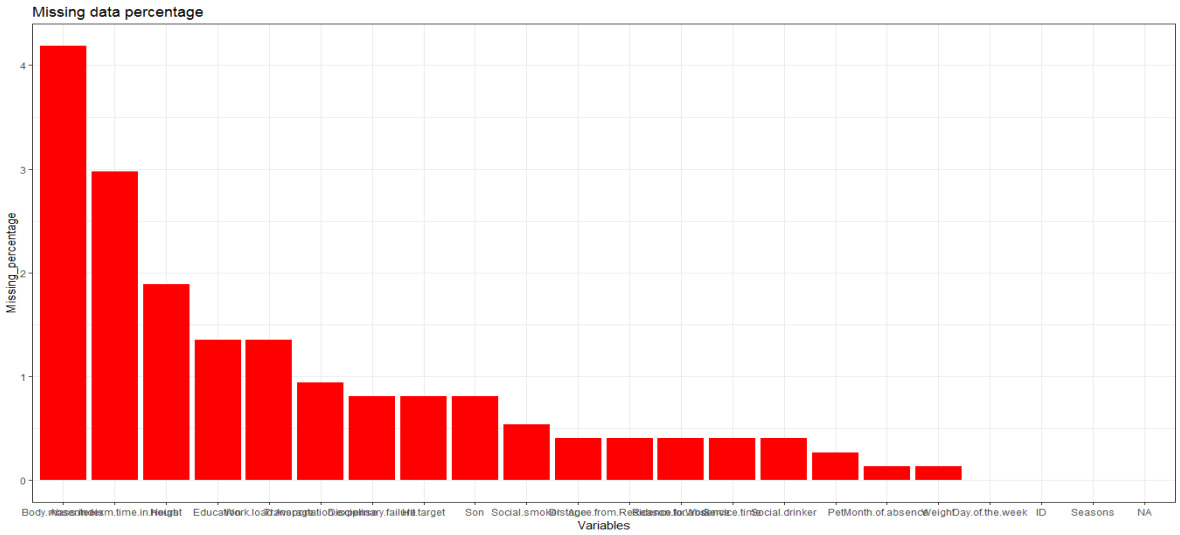
As we have the old data and we want to predict the losses in 2011 if same trend of Absenteeism continues. I think, here we can calculate **loss of work** is in a form of **time in hours.**

Loss of work in hours = (Total sum of month of absence/12)* Absenteeism time in hours

Appendix

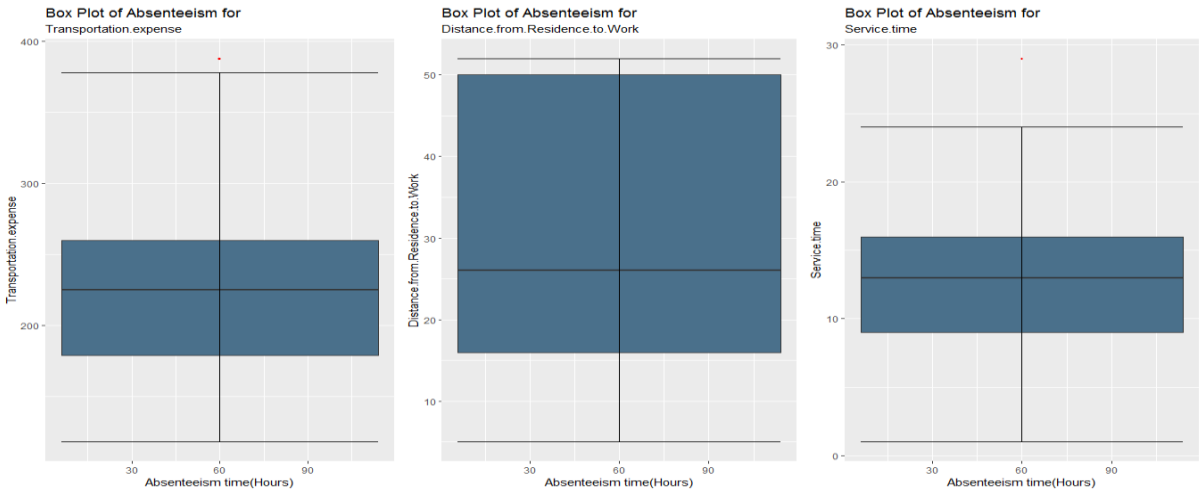
A. Extra Figures: Some visualization in R

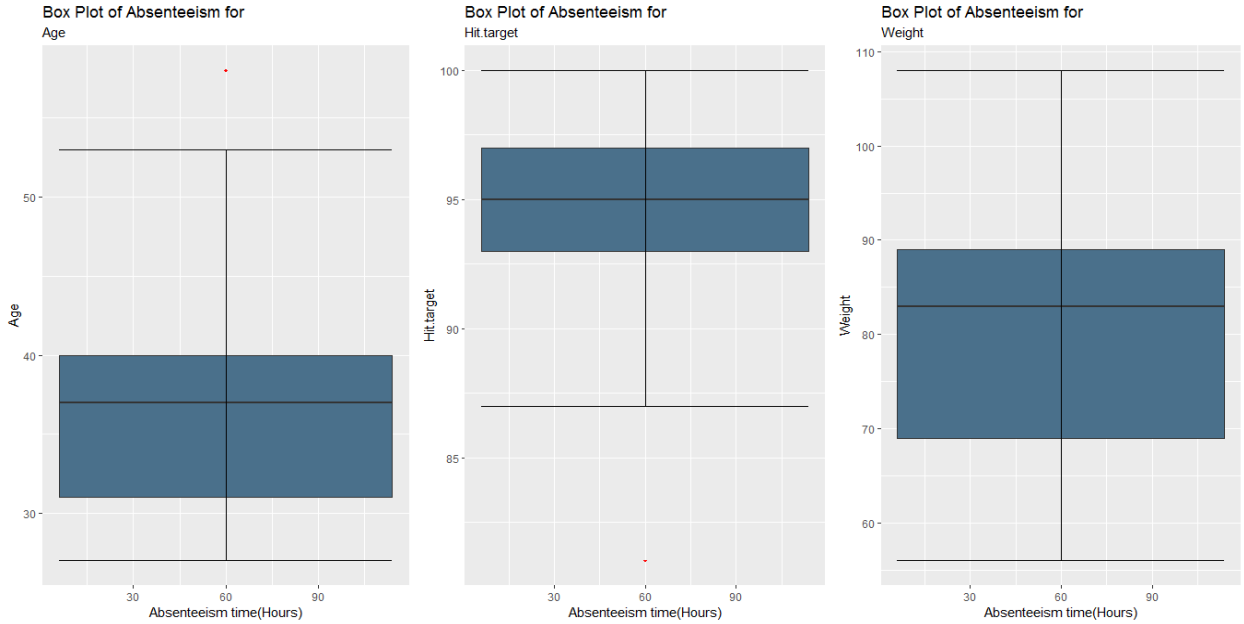
1. Missing value percentage graph

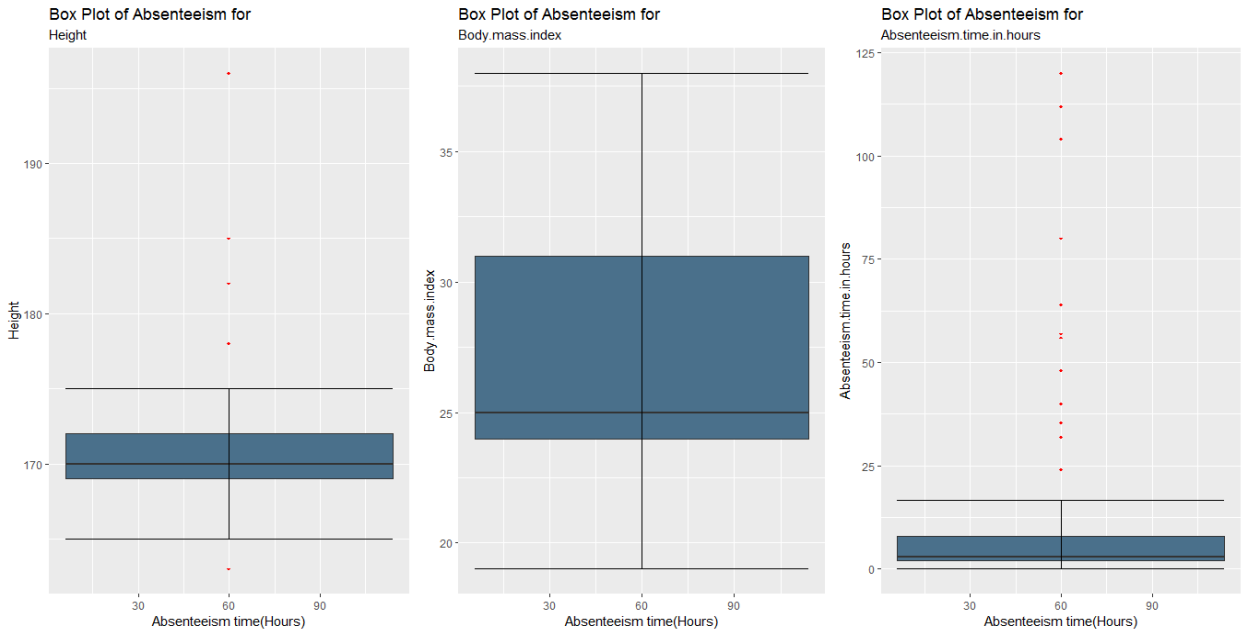


2. Outlier Analysis Using R

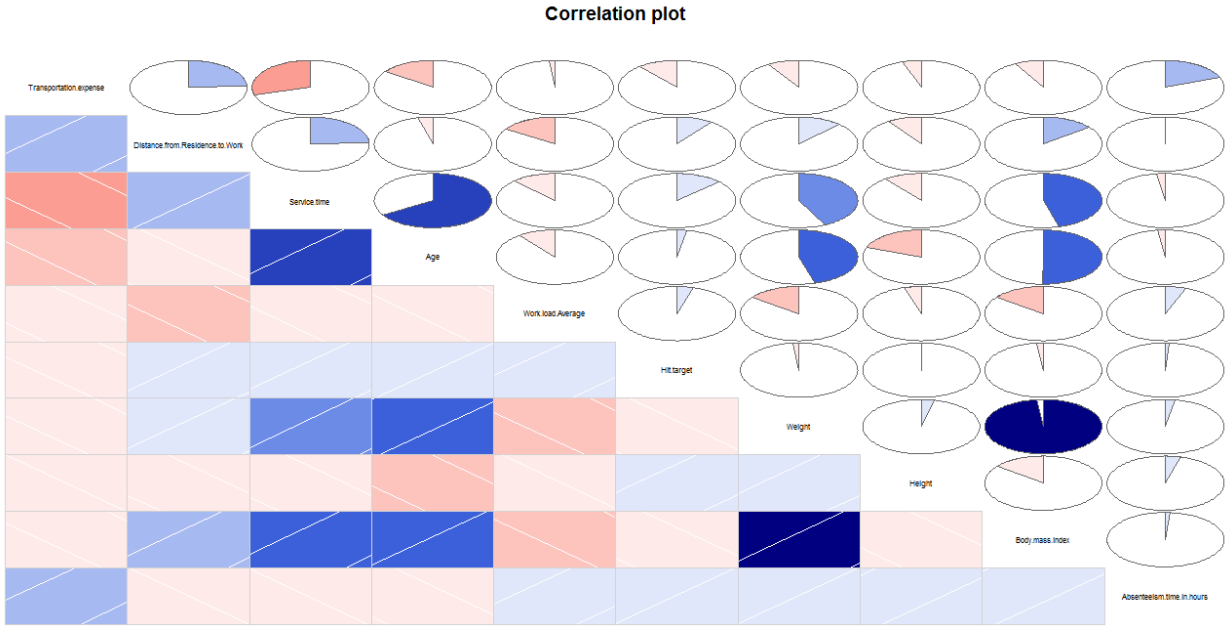
- Boxplot For the continuous variables



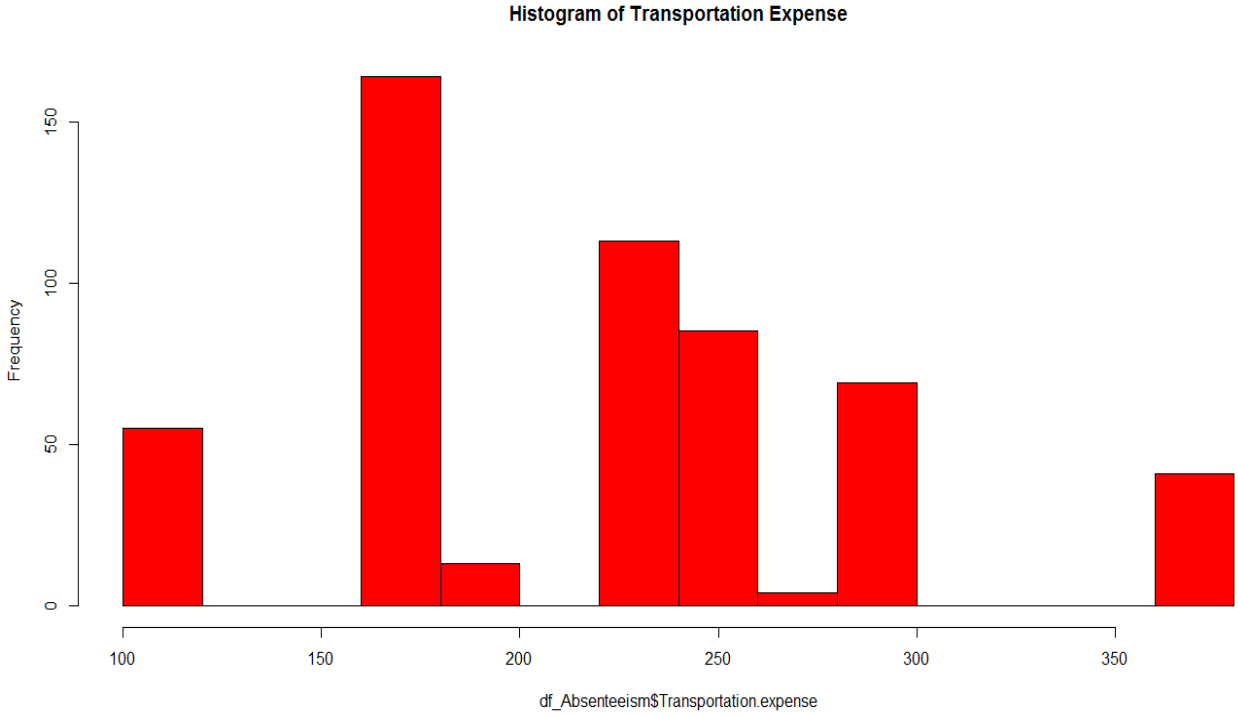

Box Plot of Absenteeism for
Age
Age
Absenteeism time(Hours)
Box Plot of Absenteeism for
Hit.target
Hit.target
Absenteeism time(Hours)
Box Plot of Absenteeism for
Weight
Weight
Absenteeism time(Hours)


Box Plot of Absenteeism for
Height
Height
Absenteeism time(Hours)
Box Plot of Absenteeism for
Body.mass.index
Body.mass.index
Absenteeism time(Hours)
Box Plot of Absenteeism for
Absenteeism.time.in.hours
Absenteeism.time.in.hours
Absenteeism time(Hours)

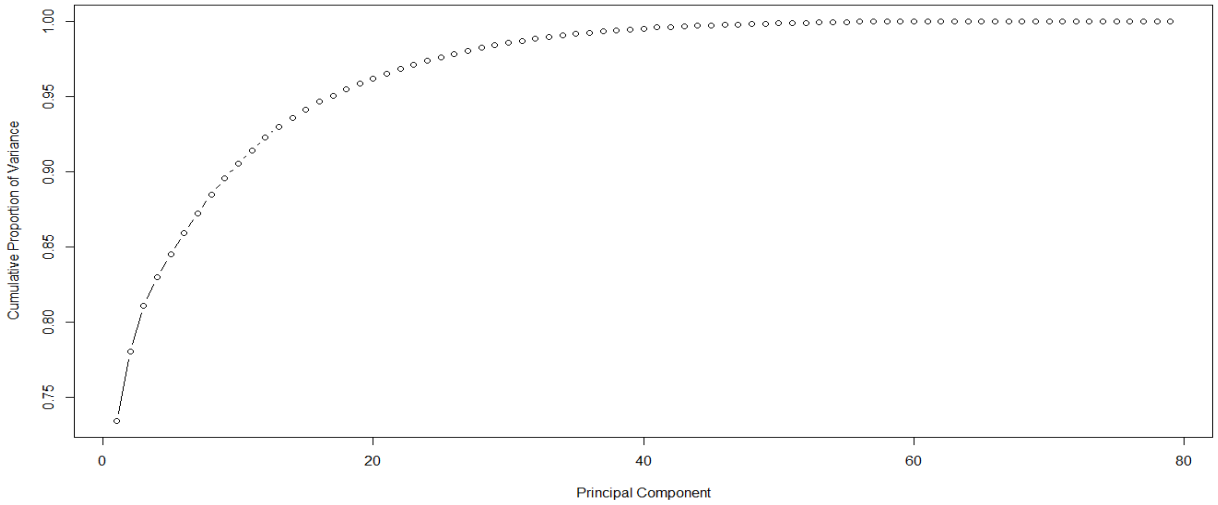
3. Correlation Plot for Future Engineering



4. Normality checking for FUTURE SCALING

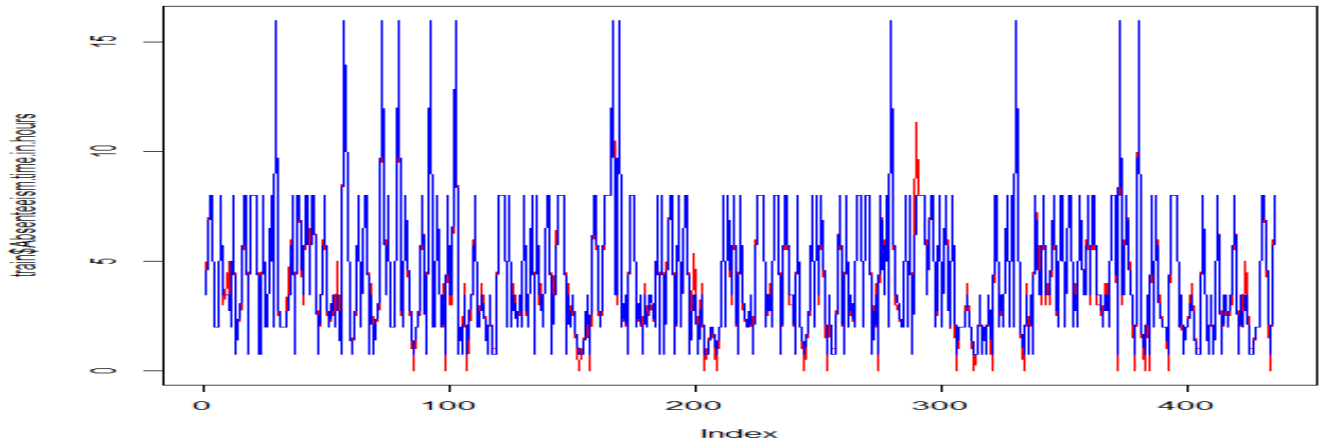


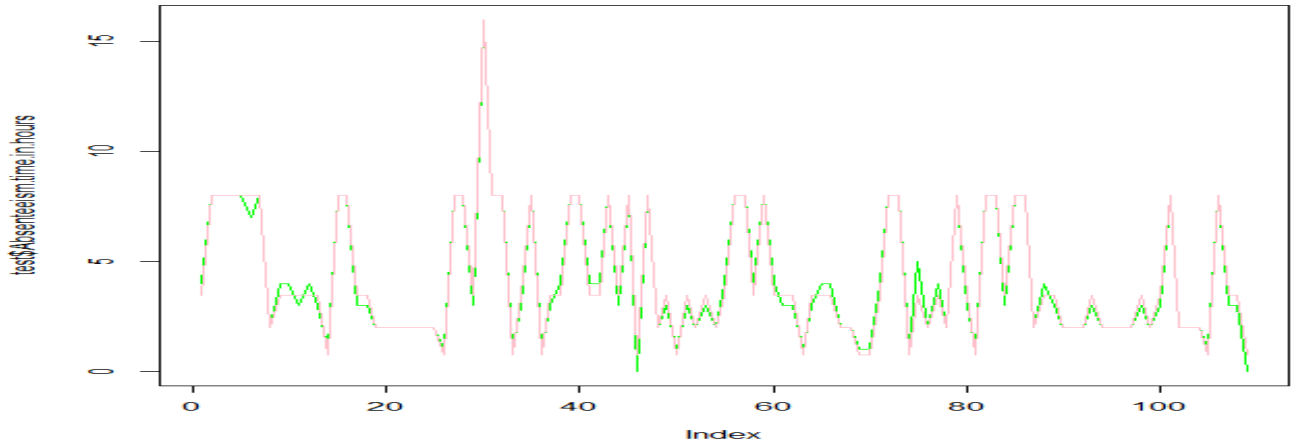
5. Cumulative Plot for PCA



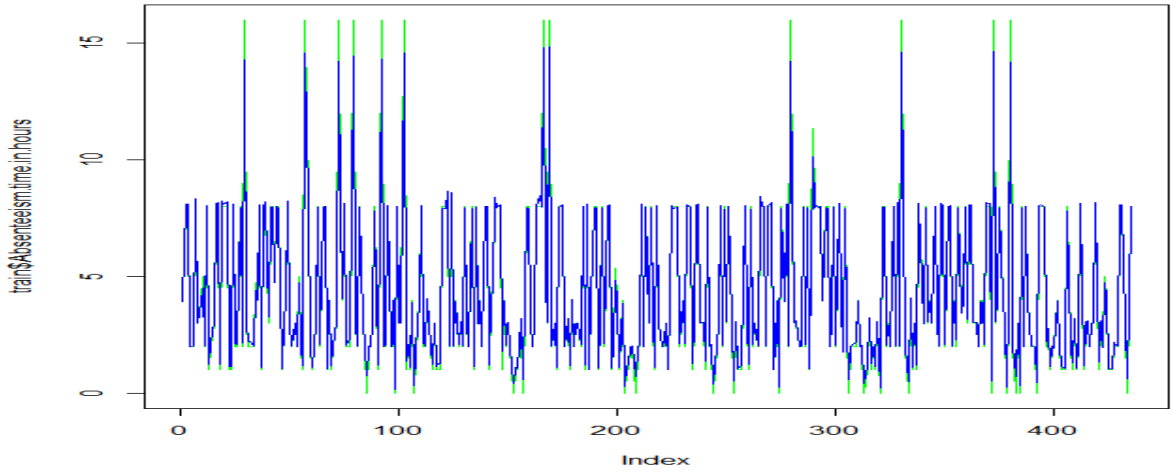
6. Model Performance on Train & Test

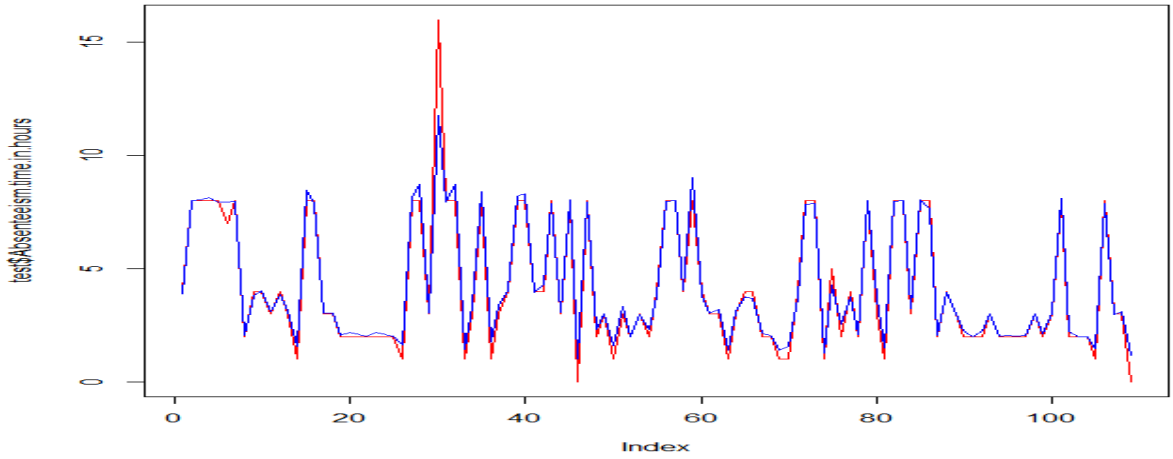
- **DT**



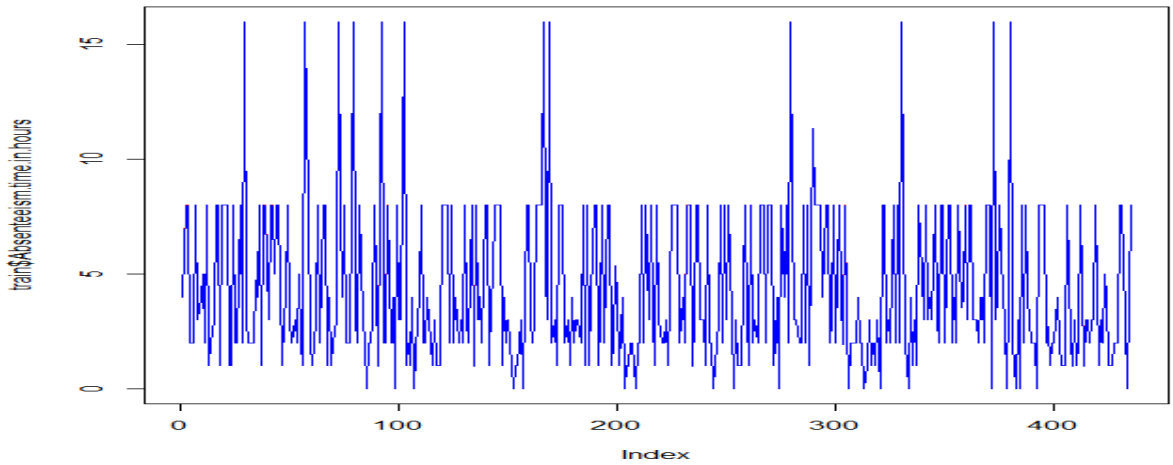


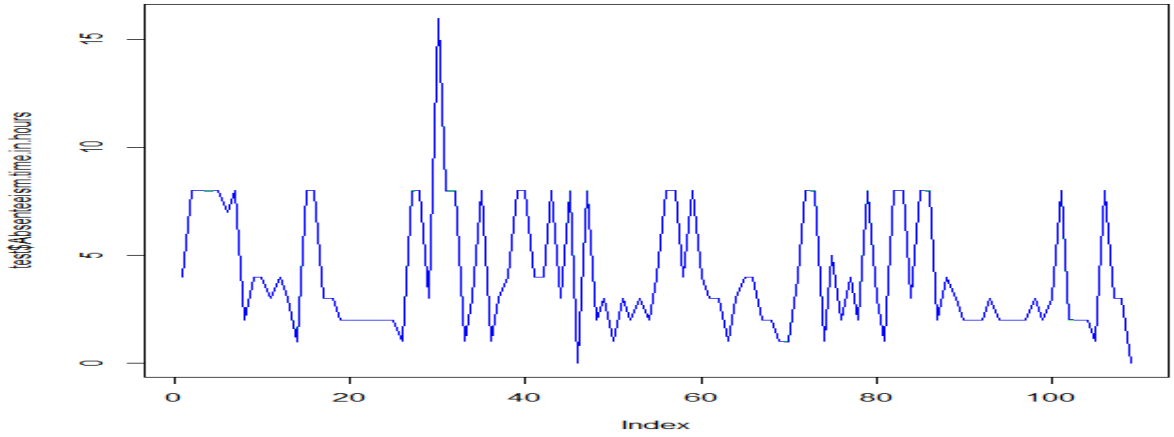
- **RF**


train$Absenteeism.time.in.hours
0
5
10
15
0
100
200
300
400
Index

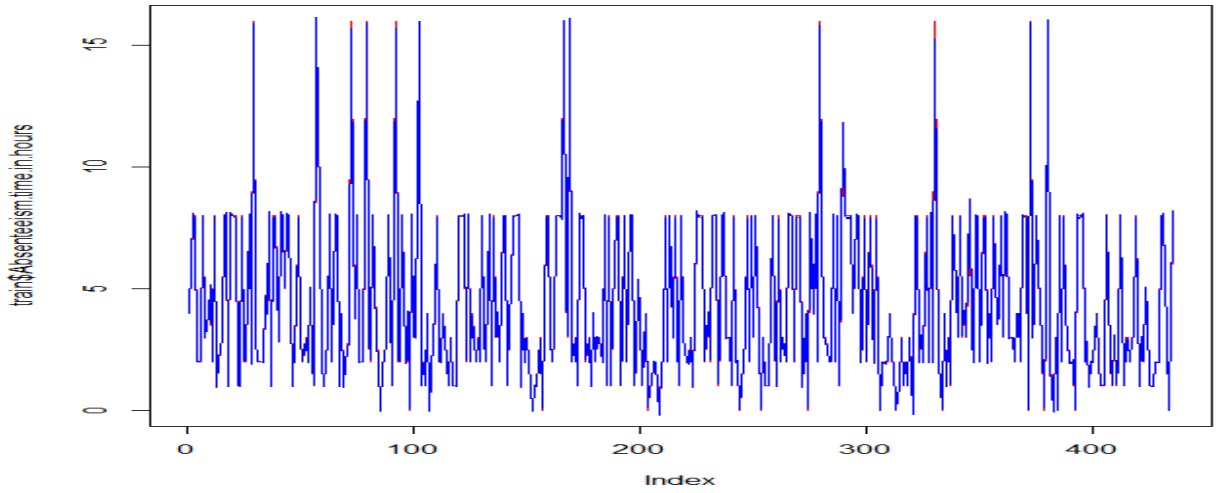

test$Absenteeism.time.in.hours
0
5
10
15
0
20
40
60
80
100
Index

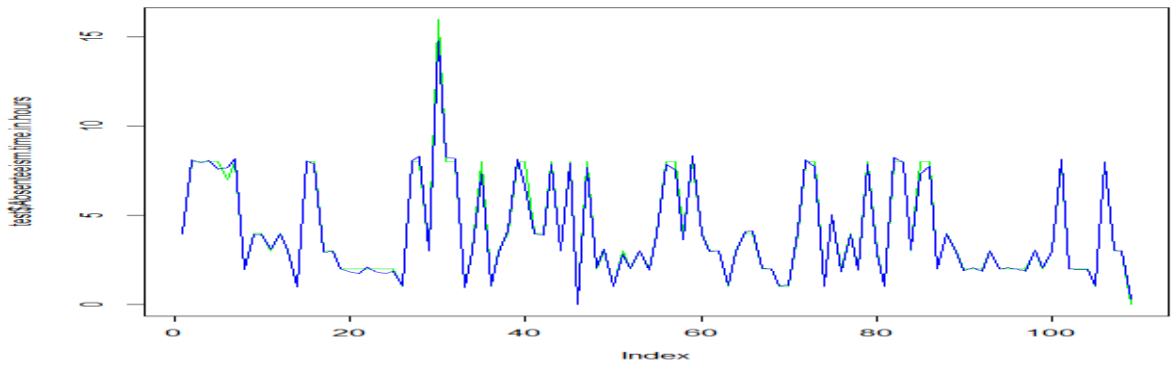
- **LR**


train$Absenteeism.time.in.hours
0
5
10
15
0
100
200
300
400
Index


test$Absenteeism.time.in.hours
0
5
10
15
0
20
40
60
80
100
Index

- **XGBOOST**


train$Absenteeism.time.in.hours
0
5
10
15
0
100
200
300
400
Index


test$Absenteeism.time.in.hours
0
5
10
15
0
20
40
60
80
100
Index

B. Code

R CODE

```
rm(list=ls())

setwd("C:/Users/User/Desktop/Project 2/File")
getwd()

#Load Libraries
x = c("xlsx","ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced",
"dummies", "e1071", "Information",
     "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees')

#install.packages(x)
lapply(x, require, character.only = TRUE)

## Read the data
df_Absenteeism = read.xlsx('Absenteeism.xlsx', sheetIndex = 1)

#----------Exploratory Data Analysis----------#

head(df_Absenteeism)
dim(df_Absenteeism)
names(data)
summary(df_Absenteeism)
colnames(df_Absenteeism)

#----------Missing Values Analysis----------#
#calculating missing value

missing_val = data.frame(apply(df_Absenteeism,2,function(x){sum(is.na(x))}))
missing_val$Columns = row.names(missing_val)
names(missing_val)[1] =  "Missing_percentage"

#Calculating percentage missing value
missing_val$Missing_percentage = (missing_val$Missing_percentage/nrow(df_Absenteeism)) *
100

# Sorting missing_val in Descending order
missing_val = missing_val[order(-missing_val$Missing_percentage),]
row.names(missing_val) = NULL
```

```
# Reordering columns
missing_val = missing_val[,c(2,1)]

# # Missing value plot
ggplot(data = missing_val[1:100,], aes(x=reorder(Columns, -Missing_percentage),y =
Missing_percentage))+
geom_bar(stat = "identity",fill = "red")+xlab("Variables")+
ggtitle("Missing data percentage") + theme_bw()

# Let us consider which method is sutable for our data to remove missing value
df_Absenteeism$Distance.from.Residence.to.Work[85]

# Actual Value = 31
# Mean = 29.66576
# Median = 26
# KNN = 31


#Mean Method
#df_Absenteeism$Distance.from.Residence.to.Work[85]=NA
#df_Absenteeism$Distance.from.Residence.to.Work[is.na(df_Absenteeism$Distance.from.Resid
ence.to.Work)] = mean(df_Absenteeism$Distance.from.Residence.to.Work, na.rm = T)

#Median Method
#df_Absenteeism$Distance.from.Residence.to.Work[85]=NA
#df_Absenteeism$Distance.from.Residence.to.Work[is.na(df_Absenteeism$Distance.from.Resid
ence.to.Work)] = median(df_Absenteeism$Distance.from.Residence.to.Work, na.rm = T)

# kNN Imputation
df_Absenteeism$Distance.from.Residence.to.Work[85]=NA
df_Absenteeism = knnImputation(df_Absenteeism, k = 3)

# Checking for missing value
sum(is.na(df_Absenteeism))

############################### Differentiating data ##############################

cnames = c('Transportation.expense','Distance.from.Residence.to.Work', 'Service.time', 'Age',
        'Work.load.Average', 'Hit.target', 'Weight', 'Height', 'Body.mass.index',
'Absenteeism.time.in.hours')

cat_names= c('ID','Reason.for.absence','Month.of.absence','Day.of.the.week',
                'Seasons','Disciplinary.failure', 'Education', 'Son', 'Social.drinker',
                'Social.smoker',  'Pet')
```

```
#-----------------------------------Outlier Analysis-----------------------------------#

# Boxplot for continuous variables

for(i in 1:length(cnames))

{assign(paste0("PK",i),ggplot(aes_string(y=(cnames[i]),x="Absenteeism.time.in.hours"),d=subs
et(df_Absenteeism))
       +geom_boxplot(outlier.colour = "Red",outlier.shape = 18,outlier.size = 1,
fill="skyblue4")+theme_gray()
       +stat_boxplot(geom = "errorbar", width=0.5) +labs(y=cnames[i],x="Absenteeism
time(Hours)")
       +ggtitle("Box Plot of Absenteeism for",cnames[i]))

}

for (i in 1:length(cnames))

# ## Plotting plots together
gridExtra::grid.arrange(PK1,PK2,PK3,ncol=3)
#gridExtra::grid.arrange(PK4,PK6,PK7,ncol=3)
#gridExtra::grid.arrange(PK8,PK9,PK10,ncol=3)

# #Remove outliers using boxplot method

# #loop to remove from all variables
for(i in cnames)
{
  print(i)
  val = df_Absenteeism[,i][df_Absenteeism[,i] %in% boxplot.stats(df_Absenteeism[,i])$out]
  #print(length(val))
  df_Absenteeism = df_Absenteeism[which(!df_Absenteeism[,i] %in% val),]
}
dim(df_Absenteeism)


#-------------------------------Feature Selection-------------------------------------------#

#Correlation Analysis for continuous variables-

corrgram(df_Absenteeism[,cnames],order=FALSE,upper.panel = panel.pie,
       text.panel = panel.txt,font.labels =1, main="Correlation plot")


# Weight is highly correlated
```

#Anova Test for categorical variable-

```
for(i in cat_names){
  print(i)
  Anova_result= summary(aov(formula = Absenteeism.time.in.hours~df_Absenteeism[,i]
,df_Absenteeism))
  print(Anova_result)
}
```

 # Dimension Reduction

```
df_Absenteeism = subset(df_Absenteeism, select = -c(Weight, Month.of.absence, Seasons ,
Education , Social.smoker , Pet))

dim(df_Absenteeism)
```

############################## FUTURE SCALING ##############################

```
hist(df_Absenteeism$Transportation.expense,col="Red",main="Histogram of Transportation
Expense")
```

 # Updating the continuous and catagorical variable

```
cnames = c('Distance.from.Residence.to.Work', 'Service.time', 'Age','Work.load.Average',
          'Transportation.expense','Hit.target', 'Height', 'Body.mass.index')

cat_names = c('ID','Reason.for.absence','Disciplinary.failure', 'Social.drinker', 'Son',
'Day.of.the.week')
```

 # Normalization

```
for(i in cnames)
{
  print(i)
  df_Absenteeism[,i] = (df_Absenteeism[,i] -
min(df_Absenteeism[,i]))/(max(df_Absenteeism[,i])-min(df_Absenteeism[,i]))
}

df = df_Absenteeism
#df_Absenteeism=df
```

 # save preprocess file

```
write.csv(df_Absenteeism,"Absenteeism_Pre_proc.csv",row.names=FALSE)
```

#################### dummy variables for categorical variables ####################

```
library(mlr)
df_Absenteeism = dummy.data.frame(df_Absenteeism, cat_names)
```

```
#================###### Model Devlopment ###====================#
#Clean the Environment-
rmExcept("df_Absenteeism")

######################### DEVIDE DATA INTO 80:20 ############################

#Divide data into train and test
set.seed(123)
train.index = sample(1:nrow(df_Absenteeism), 0.8 * nrow(df_Absenteeism))
train = df_Absenteeism[ train.index,]
test  = df_Absenteeism[-train.index,]


#----------------------Dimensionality Reduction using PCA-----------------------------#


prin_comp = prcomp(train) #PCA

std_dev = prin_comp$sdev  #compute SD of each principal component

pr_var = std_dev^2 # variance calculation

prop_varex = pr_var/sum(pr_var) ##proportion of variance

#cumulative plot

plot(cumsum(prop_varex), xlab = "Principal Component", ylab = "Cumulative Proportion of
Variance",
     type = "b")

#add a training set with principal components
train.data = data.frame(Absenteeism.time.in.hours = train$Absenteeism.time.in.hours,
prin_comp$x)

# From the above plot selecting 30 components since it explains almost 95+ % data variance
train.data =train.data[,1:30]

#transform test into PCA
test.data = predict(prin_comp, newdata = test)
test.data = as.data.frame(test.data)

#select the first 30 components
test.data=test.data[,1:30]
```

```
###################### DT for Regression ############################

                    ----------#DT Model#-----------

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

#Summary of DT model
#summary(fit_DT)

#Lets predict for training data
DT_train = predict(fit_DT, train.data)

#Lets predict for training data
DT_test = predict(fit_DT,test.data)

############################Error metrics to calculation############################

print(postResample(pred = DT_train, obs = train$Absenteeism.time.in.hours)) # For train

print(postResample(pred = DT_test, obs = test$Absenteeism.time.in.hours)) # For Test

###################### Visulaization to check the model performance #################

                                ########## TRAIN ##############
plot(train$Absenteeism.time.in.hours,type="l",lty=1.8,col="Red")
lines(DT_train,type="l",col="blue")

                                ########## TEST ###############
plot(test$Absenteeism.time.in.hours,type="l",lty=1.8,col="Green")
lines(DT_test,type="l",col="Pink")


############################  RF  ########################################

set.seed(140)
library(randomForest)  #Library for randomforest
library(inTrees)       #Library for intree transformation

#Develop Model on training data
fit_RF = randomForest(Absenteeism.time.in.hours~., data = train.data)

#Lets predict for training data
RF_train = predict(fit_RF, train.data)
```

```
RF_test = predict(fit_RF,test.data)

##########################Error metrics to calculation##########################

print(postResample(pred = RF_train, obs = train$Absenteeism.time.in.hours)) # For Train

print(postResample(pred = RF_test, obs = test$Absenteeism.time.in.hours)) # For Test

################## Visulaization to check the model performance ###################

                                ########### TRAIN ##############
plot(train$Absenteeism.time.in.hours,type="l",lty=1.8,col="Green")
lines(RF_train,type="l",col="Blue")

                                ########### TEST ###############
plot(test$Absenteeism.time.in.hours,type="l",lty=1.8,col="Red")
lines(RF_test,type="l",col="Blue")

          ########################Linear Regression#########################

set.seed(140)

#Develop Model on training data
fit_LR = lm(Absenteeism.time.in.hours ~ ., data = train.data)

#Lets predict for training data
LR_train = predict(fit_LR, train.data)

#Lets predict for testing data
LR_test = predict(fit_LR,test.data)

----------------------------# Error metrics to calculation #--------------------------------------

print(postResample(pred = LR_train, obs = train$Absenteeism.time.in.hours)) # For Train

print(postResample(pred = LR_test, obs = test$Absenteeism.time.in.hours)) # For Test

############ Visualization to check the model performance ################

                                ########### TRAIN ##############
plot(train$Absenteeism.time.in.hours,type="l",lty=1.8,col="red")
lines(LR_train,type="l",col="blue")

                                ########### TEST ###############
plot(test$Absenteeism.time.in.hours,type="l",lty=1.8,col="Green")
lines(LR_test,type="l",col="Blue")
```

########################## XGboost ##############################

```
set.seed(140)

#Develop Model on training data
fit_XGB = gbm(Absenteeism.time.in.hours~., data = train.data, n.trees = 500, interaction.depth =
2)

#Lets predict for training data
XGB_train = predict(fit_XGB, train.data, n.trees = 500)

#Lets predict for testing data
XGB_test = predict(fit_XGB,test.data, n.trees = 500)
```

----------# Error metrics to calculation #----------

```
print(postResample(pred = XGB_train, obs = train$Absenteeism.time.in.hours)) # For train
print(postResample(pred = XGB_test, obs = test$Absenteeism.time.in.hours)) # For Test
```

################ Visualization to check the model performance #################

########### TRAIN ##############

```
plot(train$Absenteeism.time.in.hours,type="l",lty=1.8,col="Red")
lines(XGB_train,type="l",col="Blue")
```

########### TEST ###############

```
plot(test$Absenteeism.time.in.hours,type="l",lty=1.8,col="Green")
lines(XGB_test,type="l",col="Blue")
```

Python CODE

Here I attached it separately

References

1. Data Cleaning, Model Development and Data Visualization we used.
https://edwisor.com/career-data-scientist

3. For Visualization using seaborn.
https://www.geeksforgeeks.org/plotting-graph-using-seaborn-python/

4. For PCA we used analytics vidhya
https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/

Thank You