PROJECT REPORT
EMPLOYEE ABSENTEEISM
Thej Kiran

20-05-2019

Contents

Chapter 1: Introduction

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 its 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 Variables

There are 21 variables in our data in which 20 are independent variables and 1 (Absenteeism time in hours) is dependent variable. Since the type of target variable is continuous, this is a regression

problem.

Variable 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 (KMs)**
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 Sample Data

	ID	Reason.for.absence	Month.of.absence	Day.of.the.week	Seasons	Transportation.expense	Distance.from.Residence.to.Work	Service.time	Age
1	11	26	Jul	Tue	summer	289	36	13	33
2	36	0	Jul	Tue	summer	118	13	18	50
3	3	23	Jul	Wed	summer	179	51	18	38
4	7	7	Jul	Thu	summer	279	5	14	39
5	11	23	Jul	Thu	summer	289	36	13	33

Work.load.Average.day.	Hit.target	Disciplinary.failure	Education	Son	Social.drinker	Social.smoker	Pet	Weight	Height	Body.mass.index
239554	97	no	high school	two	yes	no	one	90	172	30
239554	97	yes	high school	one	yes	no	zero	98	178	31
239554	97	no	high school	NA	yes	no	zero	89	170	31
239554	97	no	high school	two	yes	yes	zero	68	168	24
239554	97	no	high school	two	yes	no	one	90	172	30

Absenteeism.time.in.hours
4
0
2
4
2

Fig 1.3 – First five rows of data

1.4 Unique count

Below figure shows the unique count of all the variables present in the data.

Variables	Unique count
ID	36
Reason.for.absence	29
Month.of.absence	13
Day.of.the.week	5
Seasons	4
Transportation.expense	25
Distance.from.Residence.to.Work	26
Service.time	19
Age	23
Work.load.Average.day.	39
Hit.target	14
Disciplinary.failure	3
Education	5
Son	5
Social.drinker	3
Social.smoker	3
Pet	7
Weight	27
Height	15
Body.mass.index	18
Absenteeism.time.in.hours	20

Fig 1.4 – Unique Count of data

Chapter 2: Methodology

2.1 Pre – Processing

A predictive model need to be looked at the data before we start to create a model. However, in data mining, looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is known as Exploratory Data Analysis. In this project we look at the distribution of categorical variables and continuous variables. We also look at the missing values in the data and the outliers present in the data.

2.2 Missing Value Analysis

In statistics, missing data or missing values may occur when no data value is stored for the variable in an observation. Missing values are a common occurrence in data analysis. These values may have a significant impact on the results or conclusions that would be drawn from these data. If a variable has more than 30% of its values missing, then those values can be ignored, or the column itself is ignored. In our case, none of the columns have a high percentage of missing values. The maximum missing percentage is 4.18% i.e., Body Mass Index column. The missing values have been computed using KNN computation method.

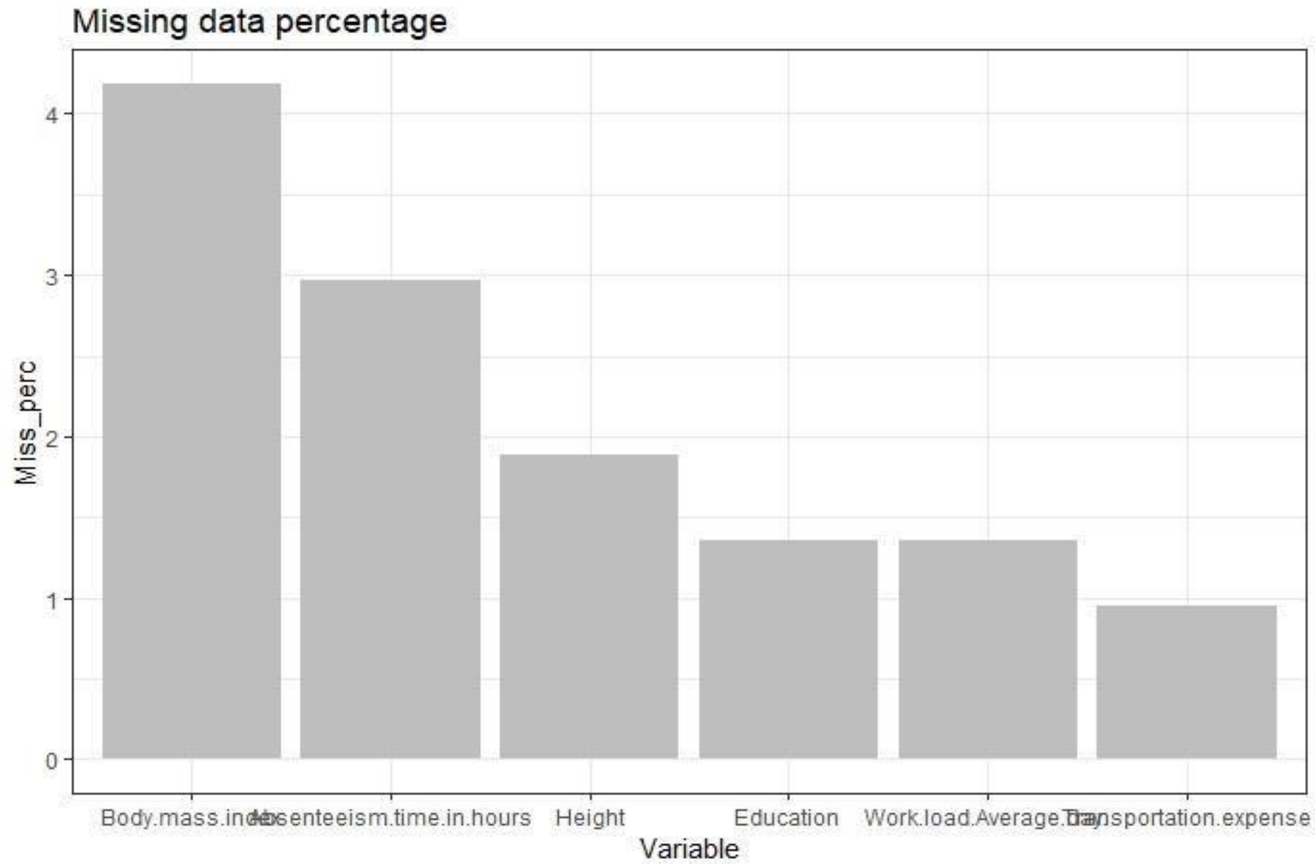


Fig 2.2 – Missing value Percentage

2.3 Outlier Analysis

It can be observed from the distribution of variables that almost none of the variables are normally distributed. The skew in these distributions can be explained by presence of outliers and extreme values in the data. One of the steps in pre-processing involves the detection and removal of such outliers. In this project, we use boxplot to visualize and remove outliers. Any value lying outside of the lower and upper whisker of the boxplot are outliers.

Variables excluding Distance from residence to work, Weight and Body mass index, contain outliers.

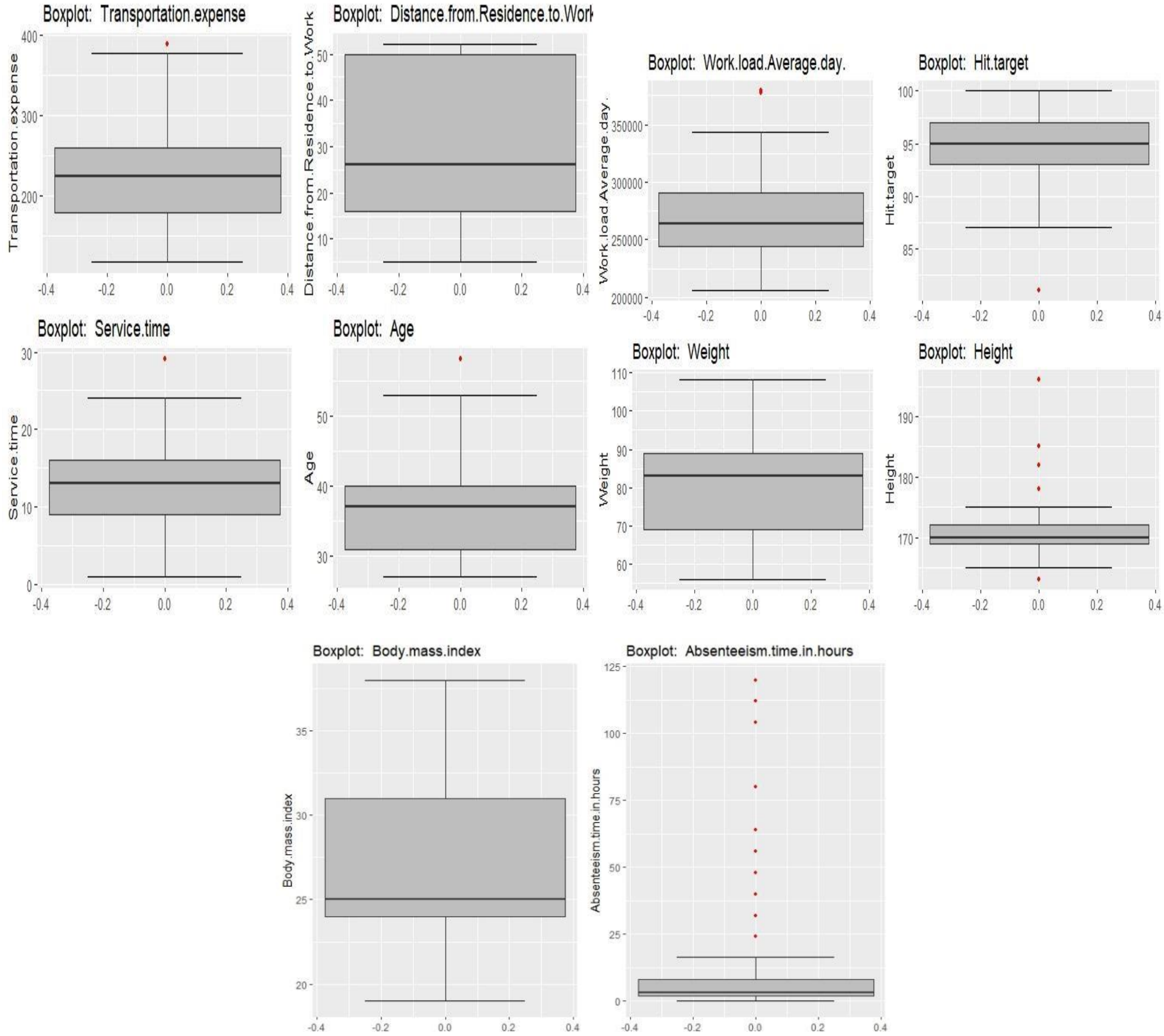


Fig 2.3.1 – Boxplots of continuous variables with outliers

Imputing outlier values:

Missing values obtained from boxplots are first converted to have NA values. Then these missing values can be imputed using KNN imputation method.

Below figure shows the boxplots of variables after removing outliers.

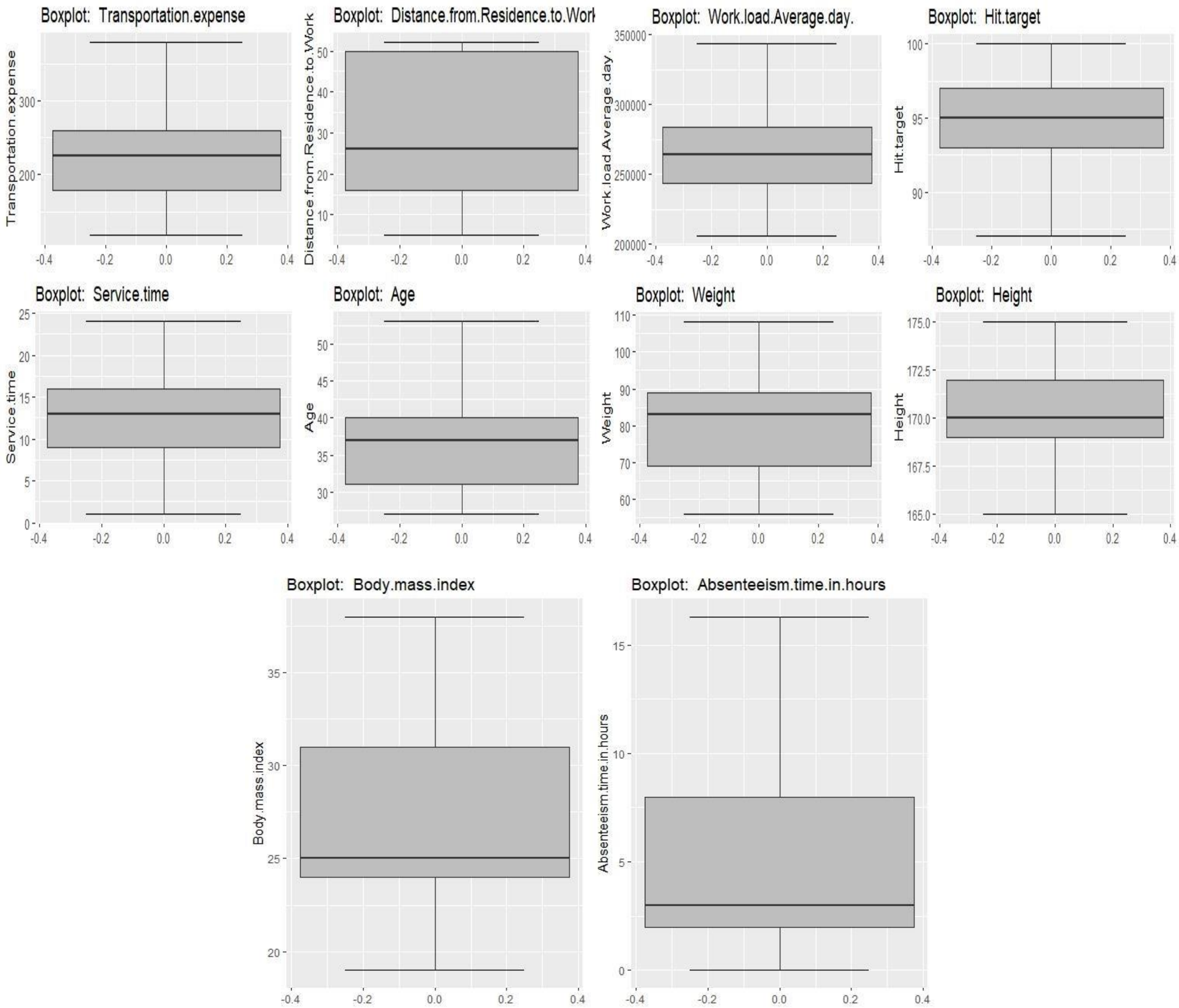


Fig 2.3.2 – Boxplots of continuous variables without outliers

2.4 Distribution of Continuous variables

By looking at the distribution of continuous variables, it can be observed that the variables are not normally distributed. Histograms are used to observe the distribution of continuous variables.

Figures are shown below

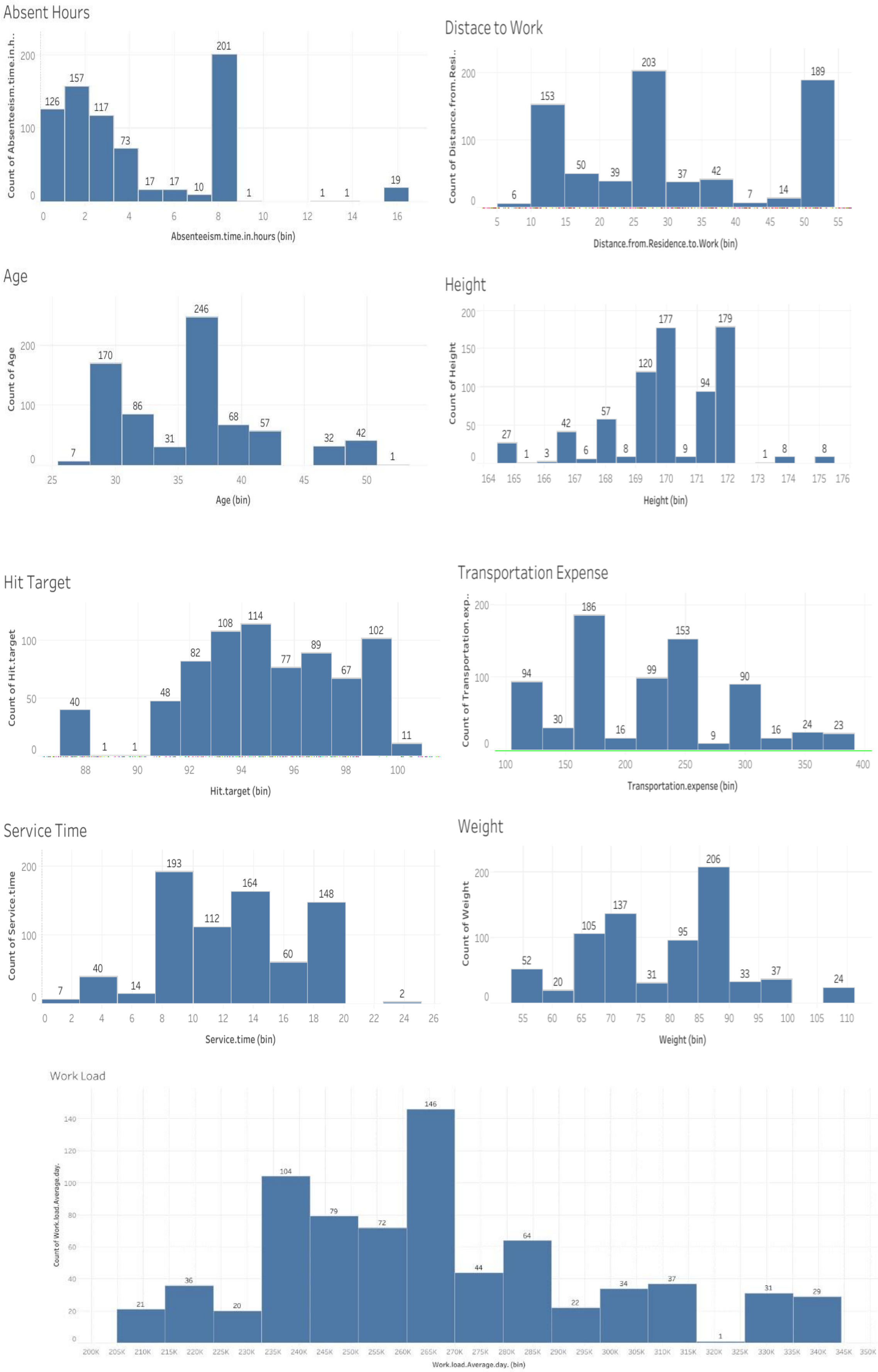


Fig 2.4 – Distribution of Continuous variables using Histogram

2.5 Distribution of Categorical Variables

Bar graphs are used to visualize the distribution of categorical variables.

Employees who are social drinkers have more absent hours than those who do not drink.

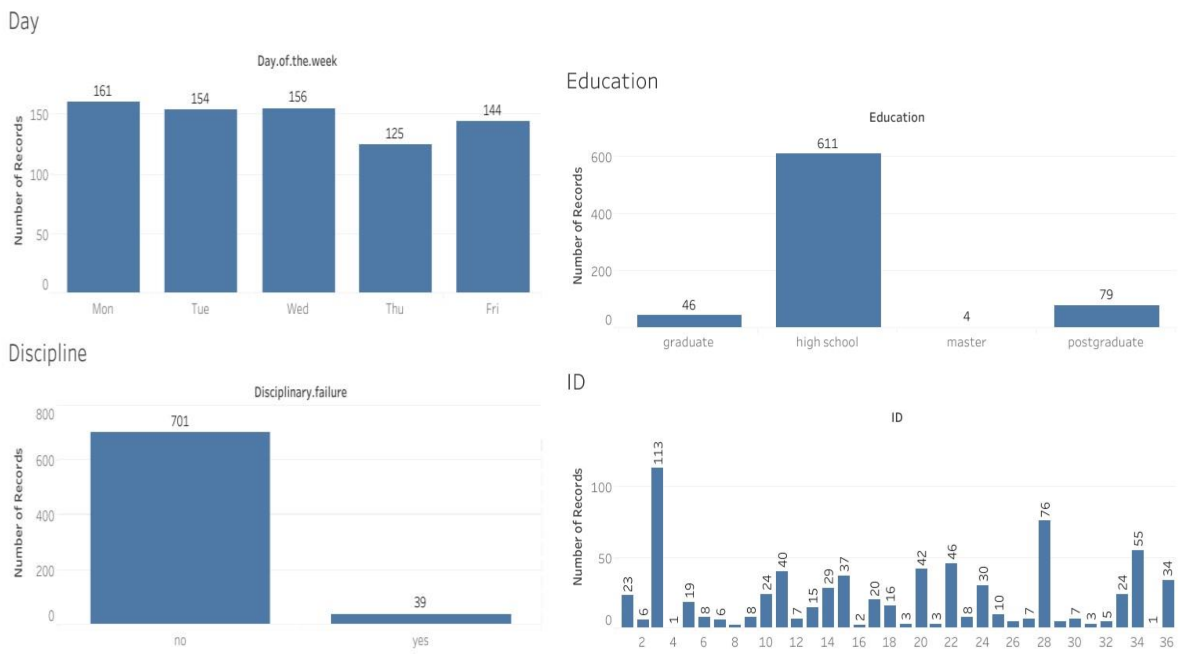
Employees having zero, one or two children have more absent hours.

Employees with ID number 3 and 28 are absent the most.

Employees are absent the most on Mondays and the least on Thursdays.

Reason 23 and 28 are the reasons employee give the most for being absent.

Employees who have completed only high school education are absent more than others. Employees are absent the most in the month of March.



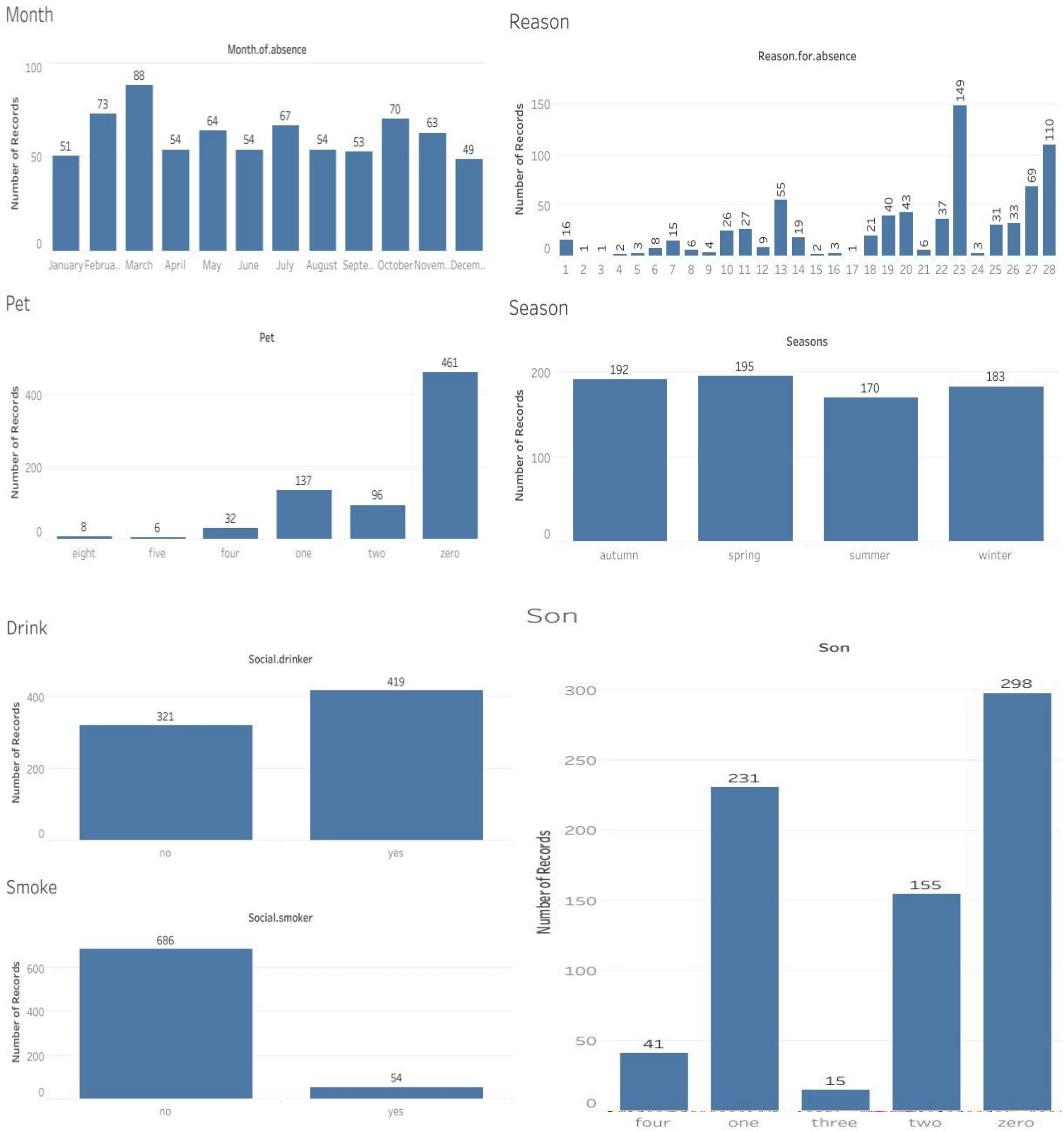


Fig 2.5 – Distribution of Categorical variables using Bar graph

2.6 Feature Selection

Feature Selection reduces the complexity of a model and makes it easier to interpret. It also reduces overfitting. Features are selected based on their scores in various statistical tests for their correlation with the outcome variable. Correlation plot is used to find out if there is any multicollinearity between variables. The highly collinear variables are dropped and then the model is executed.

From correlation analysis we have found that Weight and Body Mass Index has high correlation (>0.7), so we have excluded the Body Mass Index column.

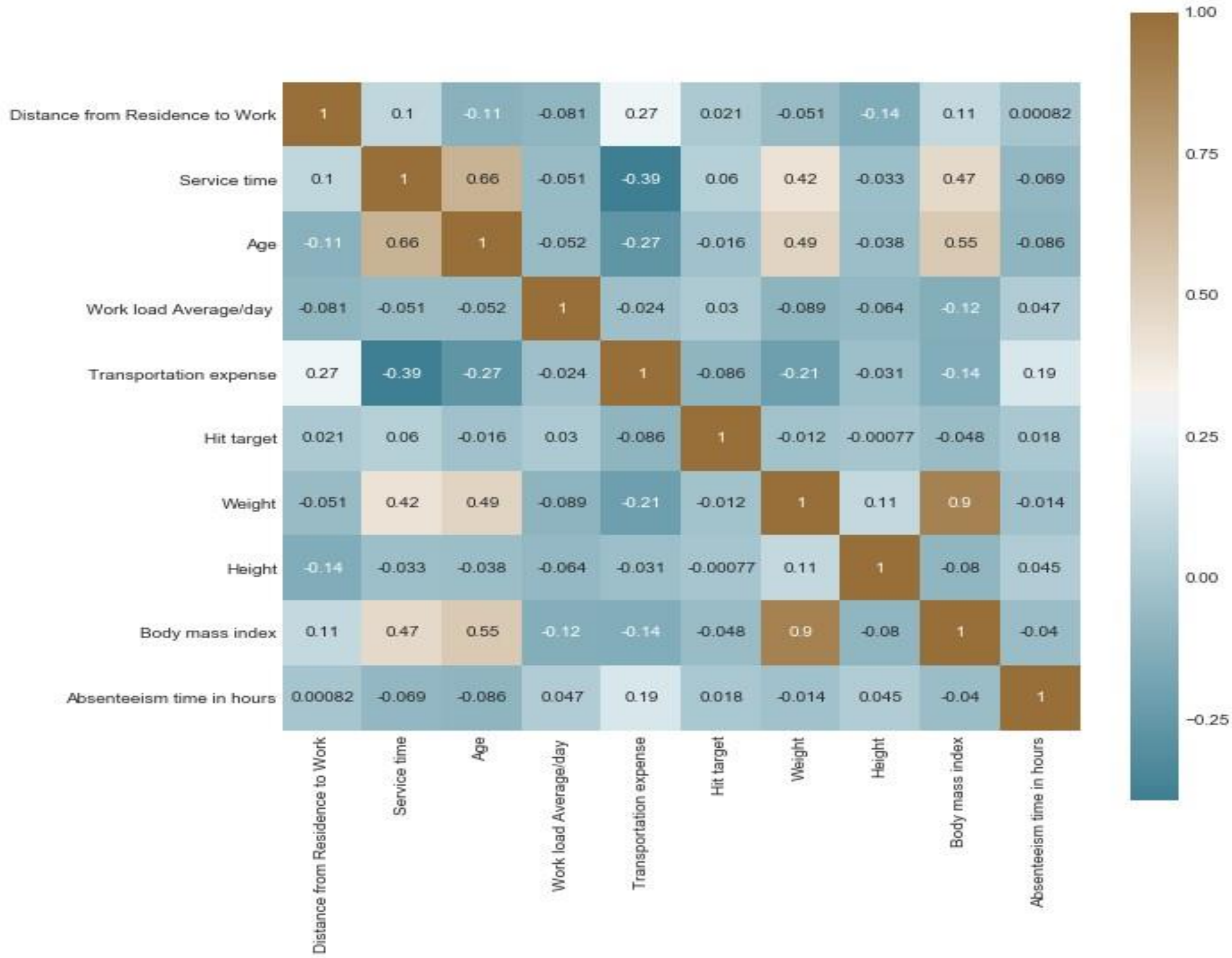


Fig 2.6 – Correlation plot of Continuous variables

2.7 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 pre-processing step.

Most classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this feature. Therefore, the range of all features should be normalized so that each feature contributes proportionately to the final

distance. Since our data is not uniformly distributed, we will use Normalization as Feature Scaling Method.

2.8 Principal Component Analysis (PCA)

Principal component analysis is a method of extracting important variables (in form of components) from a large set of variables available in a data set. It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information as possible.

After creating dummy variable of categorical variables, the data would have 116 columns and 740 observations. This high number of columns leads to bad accuracy.

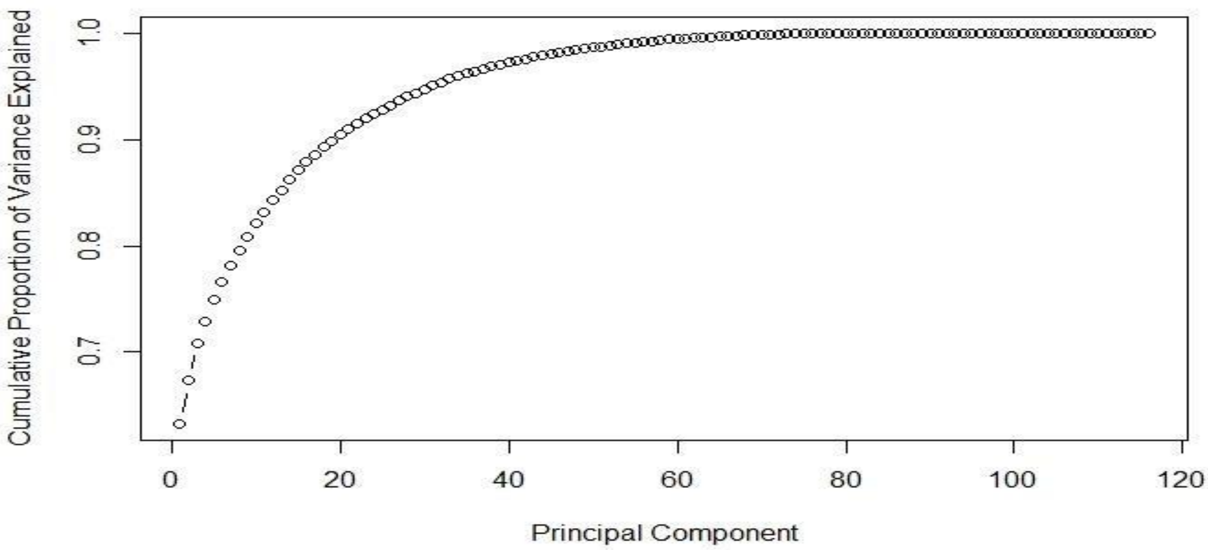


Fig 2.8 – Cumulative Scree Plot of Principal Components

After applying PCA algorithm and observing the above Cumulative Scree Plot, it can be observed that almost 95% of the data can be explained by 45 variables out of 116. Hence, we choose only 45 variables as input to the models.

Chapter 3: Modelling

3.1 Model Selection

After a thorough pre- processing we will be using some regression models on our processed data to

predict the target variable. The target variable in our model is a continuous variable i.e.,

Absenteeism time in hours. Hence the models that we choose are Linear Regression, Decision Tree and Random Forest. The error metric chosen for the given problem statement is Root Mean Square Error (RMSE).

3.2 Decision Tree

Decision Tree algorithm belongs to the family of supervised learning algorithms. Decision trees are used for both classification and regression problems.

A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome (categorical or continues value). The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data (training data).

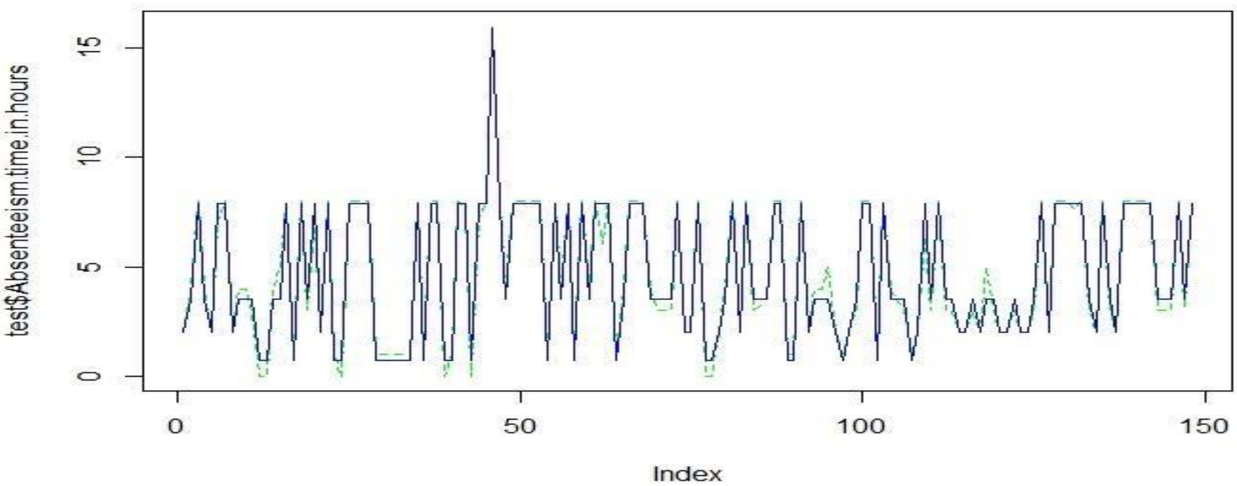


Fig 3.2 – Plot of actual values vs predicted values for Decision Tree

The RMSE values and R^2 values for the given project in R and Python are:

DECISION TREE	RMSE	R^2
R	0.442	0.978
PYTHON	0.0353	0.9998

3.3 Random Forest

Random Forest is a supervised learning algorithm. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. It can be used for both classification and regression problems. The method of combining trees is known as an ensemble method. Ensembling is nothing but a combination of weak learners (individual trees) to produce a strong learner.

The number of decision trees used for prediction in the forest is 500.

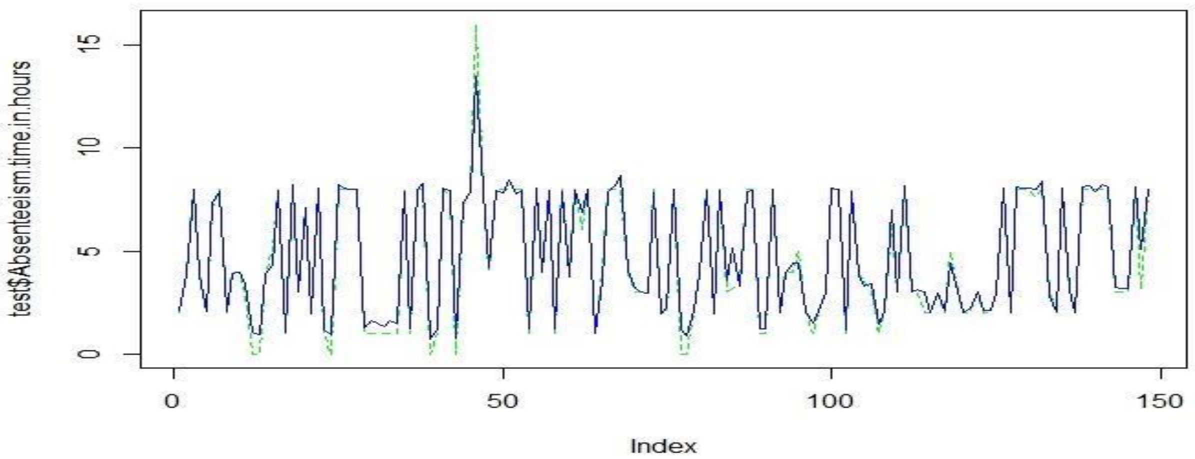


Fig 3.3 – Plot of actual values vs predicted values for Random Forest

RANDOM FOREST	RMSE	R^2
R	0.480	0.978
PYTHON	0.0445	0.9998

3.4 Linear Regression

Multiple linear regression is the most common form of linear regression analysis. Multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical.

LINEAR REGRESSION	RMSE	R^2
R	0.003	0.9999
PYTHON	0.0013	0.9999

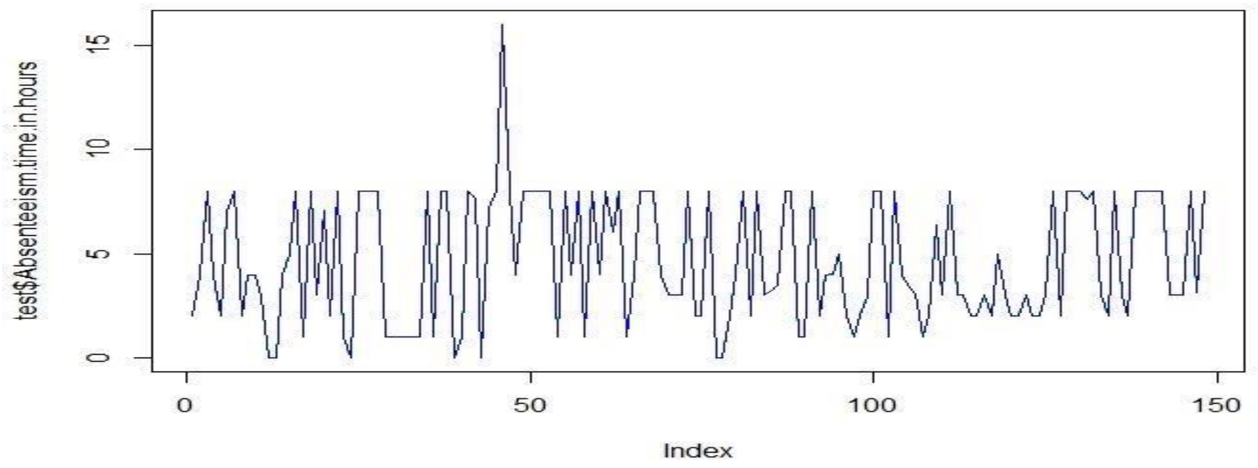


Fig 3.5 – Plot of actual values vs predicted values for Linear Regression

Chapter 4: Conclusion

4.1 Model Evaluation

In the previous chapter we have seen the Root Mean Square Error (RMSE) and R-Squared Value of different models. Root Mean Square Error (RMSE) 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. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance and has the useful property of being in the same units as the response variable. Lower values of RMSE and higher value of R-Squared Value indicate better fit.

4.2 Model Selection

From the observation of all RMSE Value and R-Squared Value we have concluded that <u>Linear</u> <u>Regression Model</u> has minimum value of RMSE and its R-Squared Value is also maximum.

4.3 Solutions of Problem Statement

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

Solution:

a. It can be observed that employees having education only till high school tend to be absent more than others. So, the company can either hire employees who have at least graduated from college or inform those employees who have completed only their high school education to reduce the number of hours they are absent.

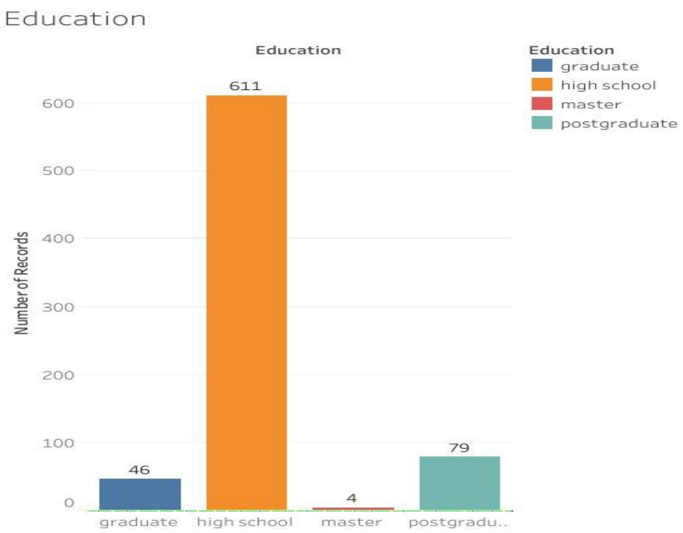


Fig 4.3.1 – Plot of Education vs Absent Hours

b. Employees with ID 3, 28 and 34 are some of the employees who are absent the most. The company may act warn such employees to reduce being absent a lot or if repeated further, it can against them if necessary.

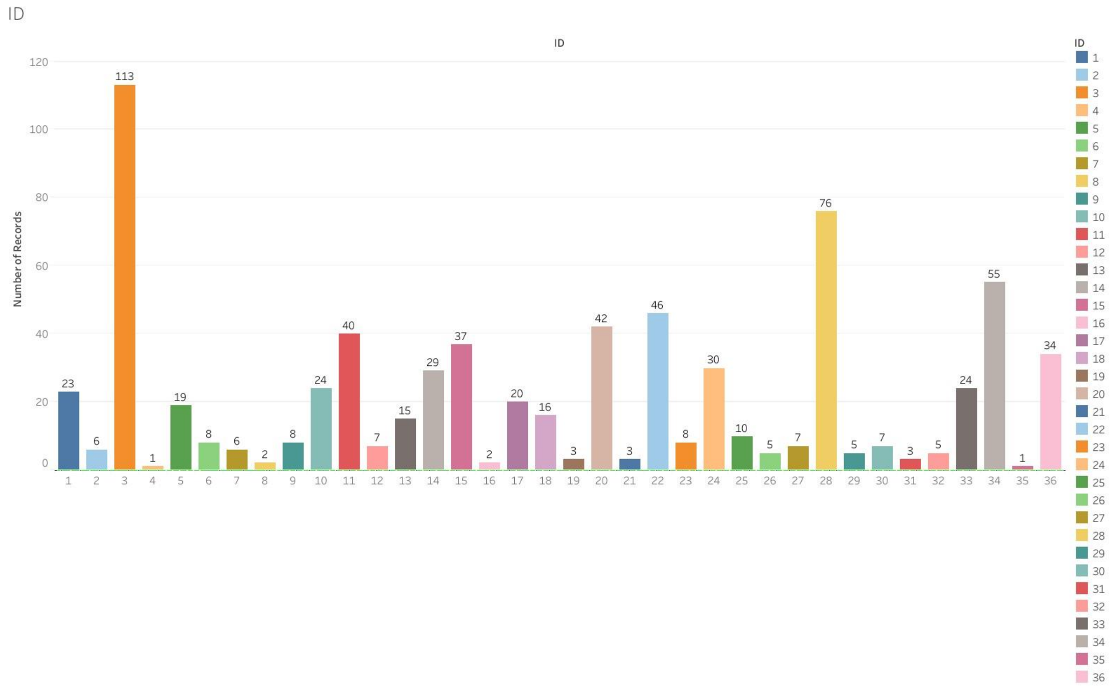


Fig 4.3.2 – Plot of ID vs Absent Hours

c. The reasons most used by employees to be absent are reason 13, 20, 23 and 28. These reasons include Medical consultation, Dental appointments, morbidity, mortality and diseases of musculoskeletal system and connective tissue. The company XYZ can help in informing employees on how to keep themselves healthier by having monthly campus consultations.

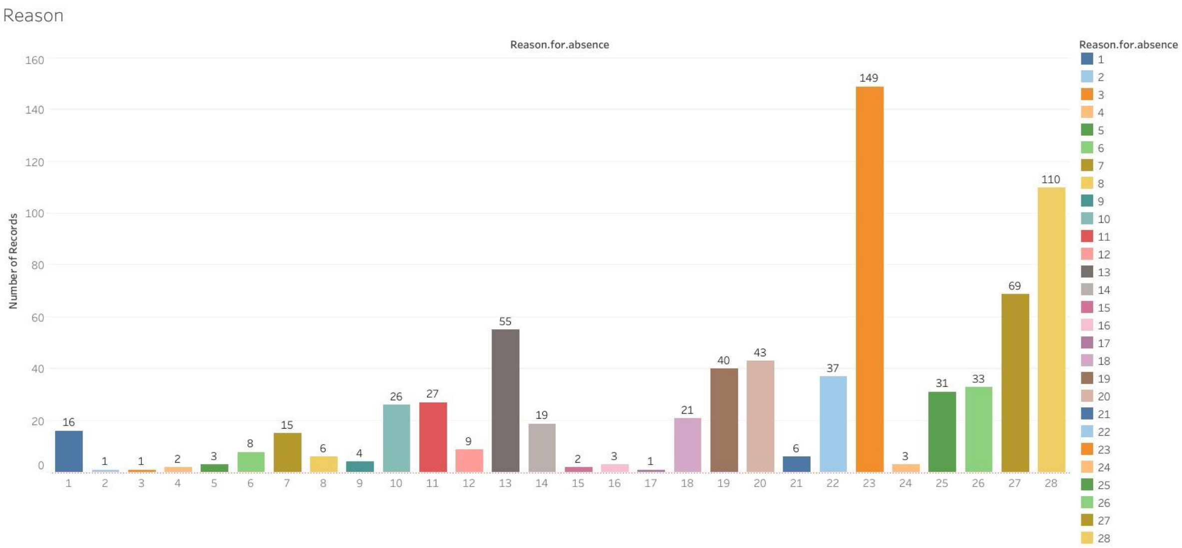


Fig 4.3.3 – Plot of Reason of Absence vs Absent Hours

d. People who tend to be social drinkers tend to be more absent than those who don't drink. XYZ can keep a track of those people and inform those employees to reduce the intake of alcohol during working days.

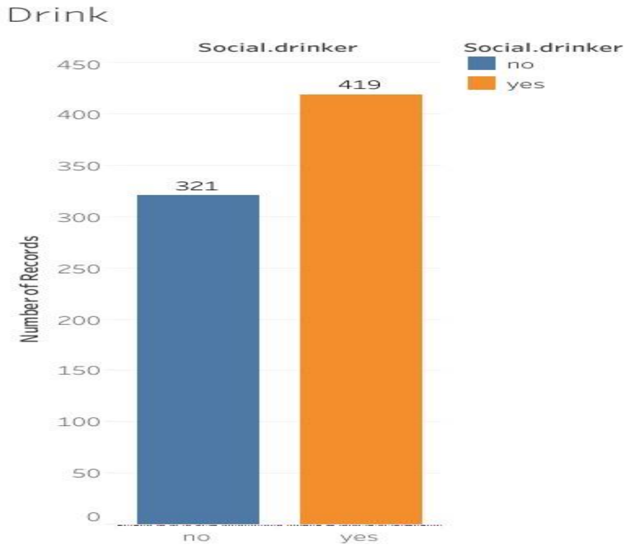


Fig 4.3.4 – Plot of Social Drinker vs Absent Hours

e. Employees are absent the most on Mondays with absent hours equal to 1426 and Tuesdays with absent hours equal to 1322.4. XYZ can inform employees to not take as many absent hours on such days.

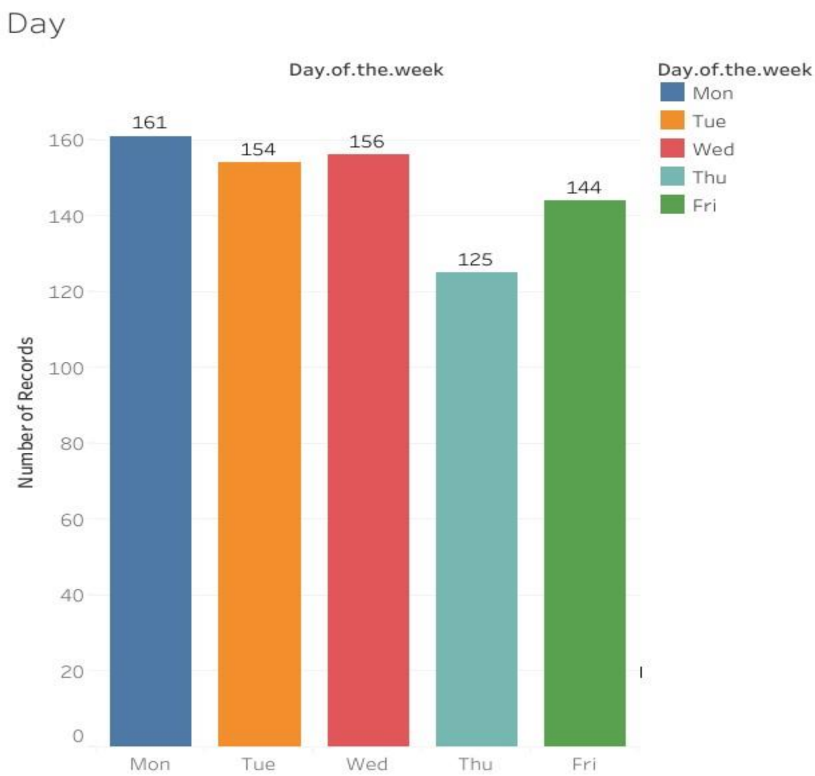


Fig 4.3.5 – Plot of Day of the Week vs Absent Hours

f. Employees are mostly absent during Spring Season.

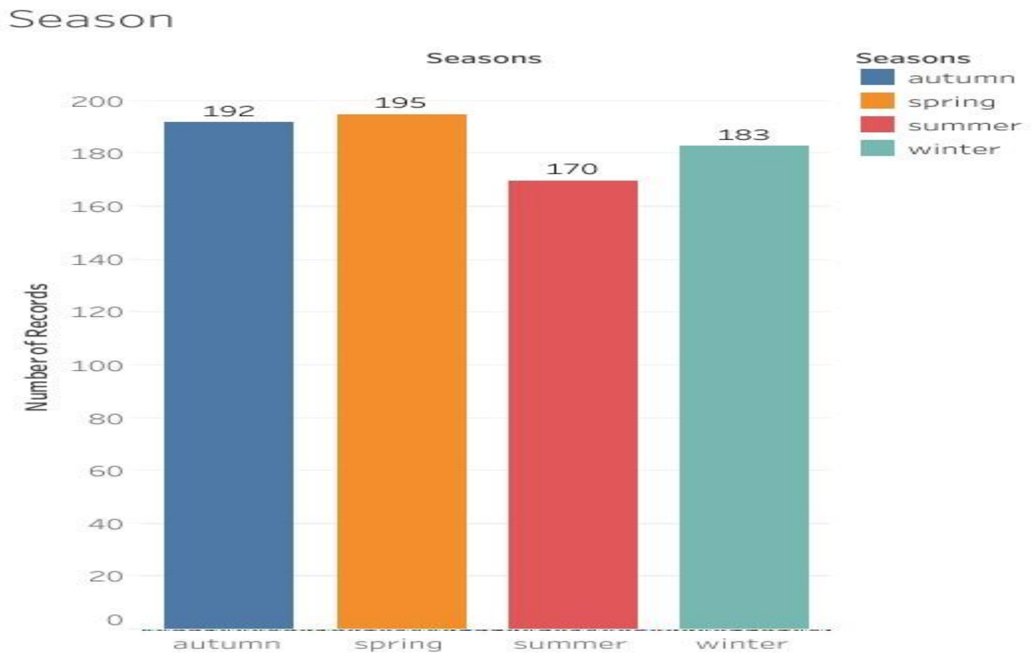


Fig 4.3.6 – Plot of Season vs Absent Hours

g. Employees having a maximum of two children or no child at all are absent the most.

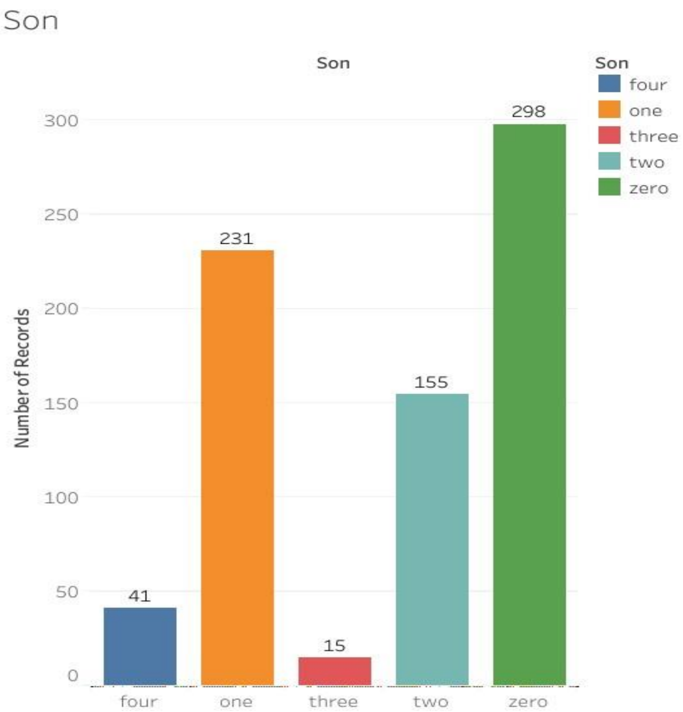


Fig: 4.3.7 – Plot of Sons vs Absent Hours

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

Solution:

Considering the losses to be the absenteeism time in hours, if the same trend of absenteeism continues, then the total losses per month is as shown in the graph below.

Employees are absent the most in the month of March, with total Absenteeism hours equal to 458.2 hours. Employees are absent the least in the month of January, with total Absenteeism hours equal to 173.6.

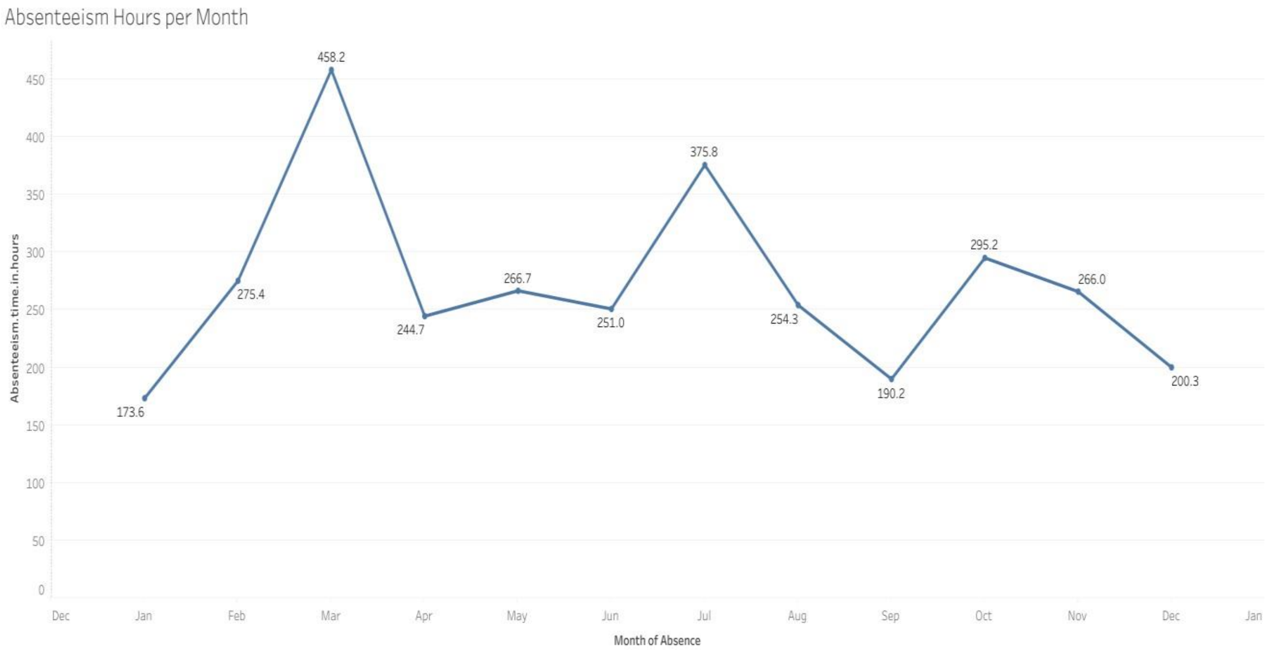


Fig 4.3.2 – Absenteeism Hours per Month

Below table shows the monthly losses of absenteeism hours:

Month	Absent Hours
January	173.6
February	275.4
March	458.2
April	244.7
May	266.7
June	251
July	375.8
August	254.3
September	190.2
October	295.2

November	**266**
December	**200.3**

Chapter 5: Appendix

5.1 Figures

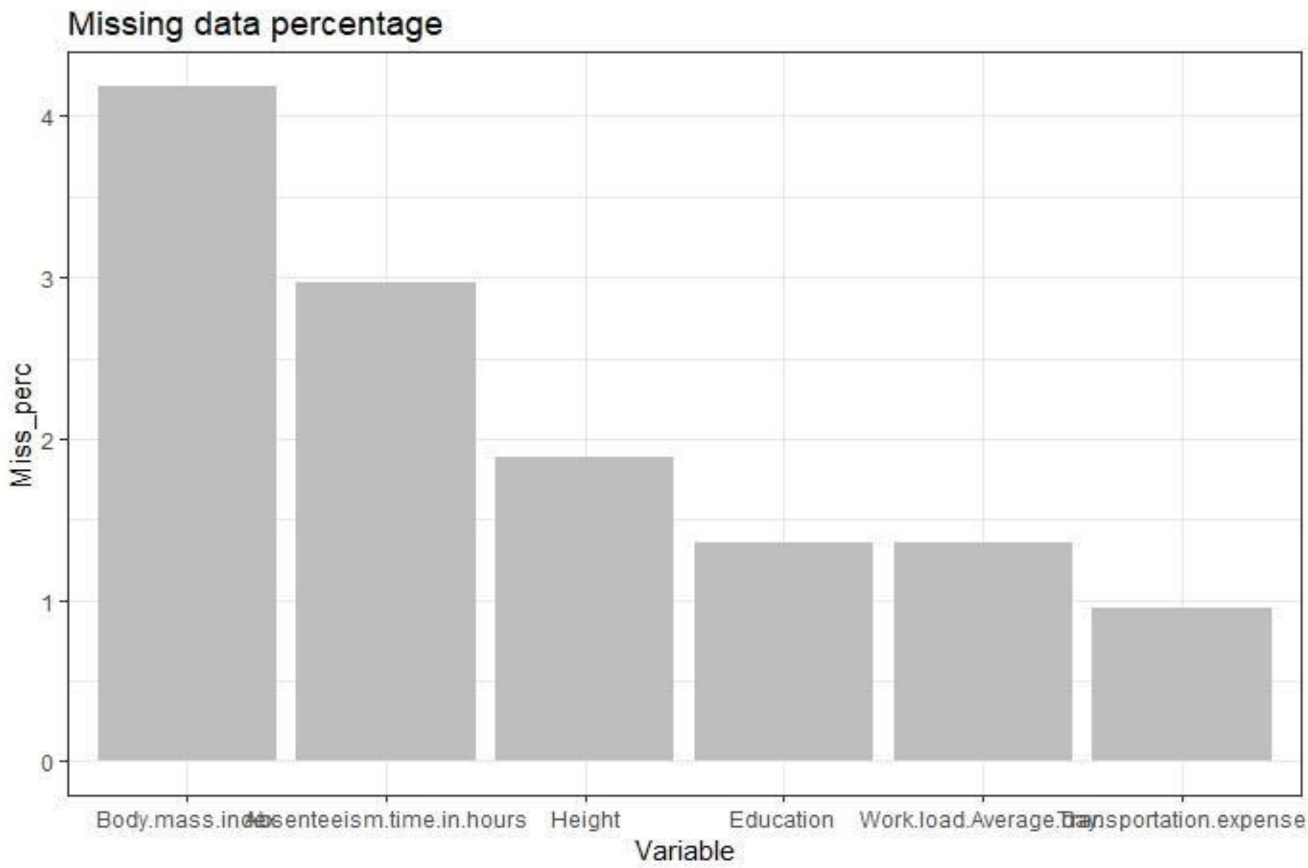


Fig 2.2 – Missing value Percentage

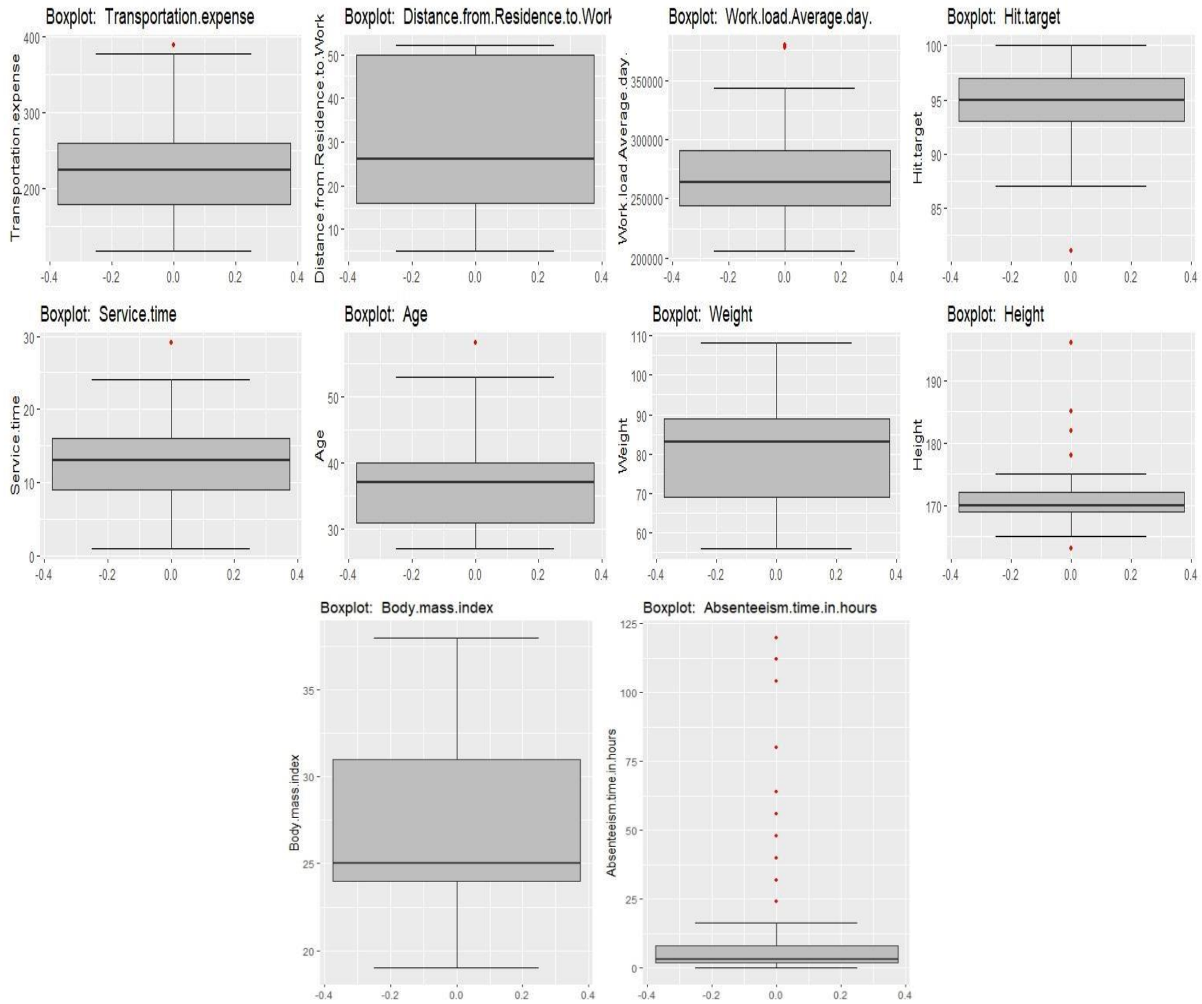
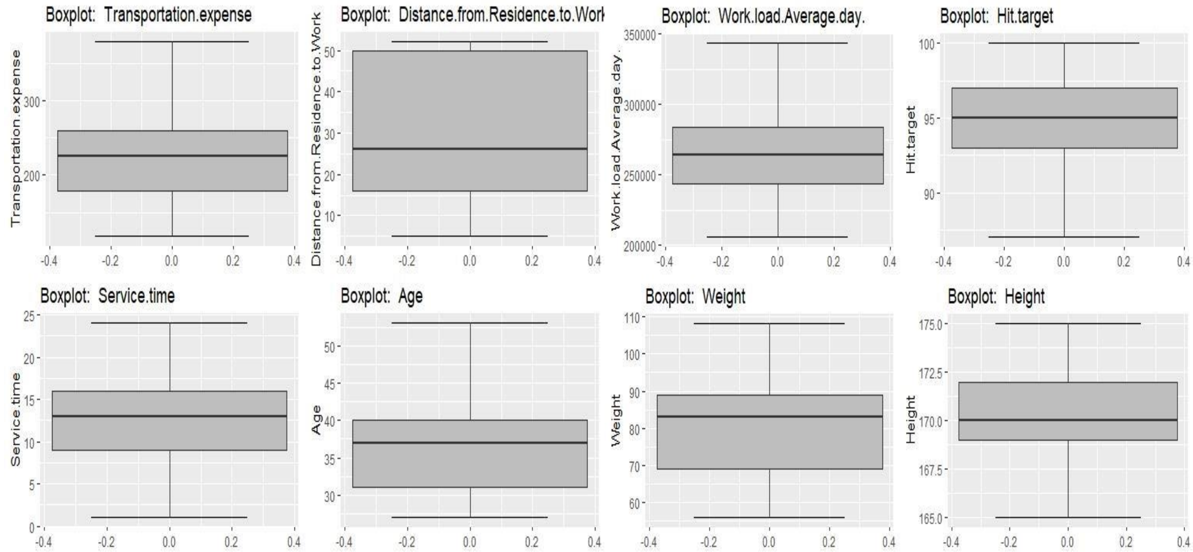


Fig 2.3.1 – Boxplots of continuous variables with outliers



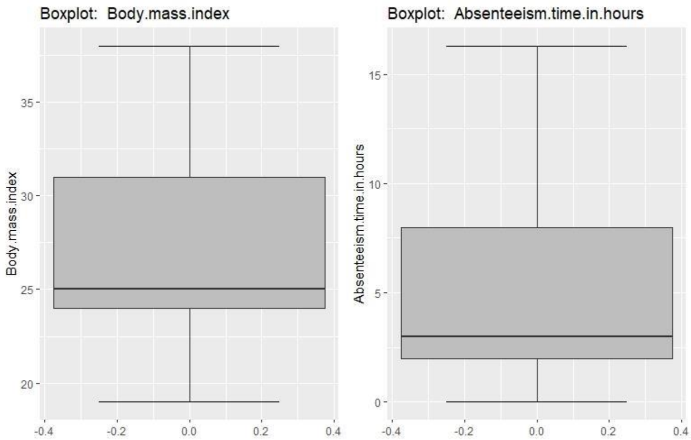


Fig 2.3.2 – Boxplots of continuous variables without outliers

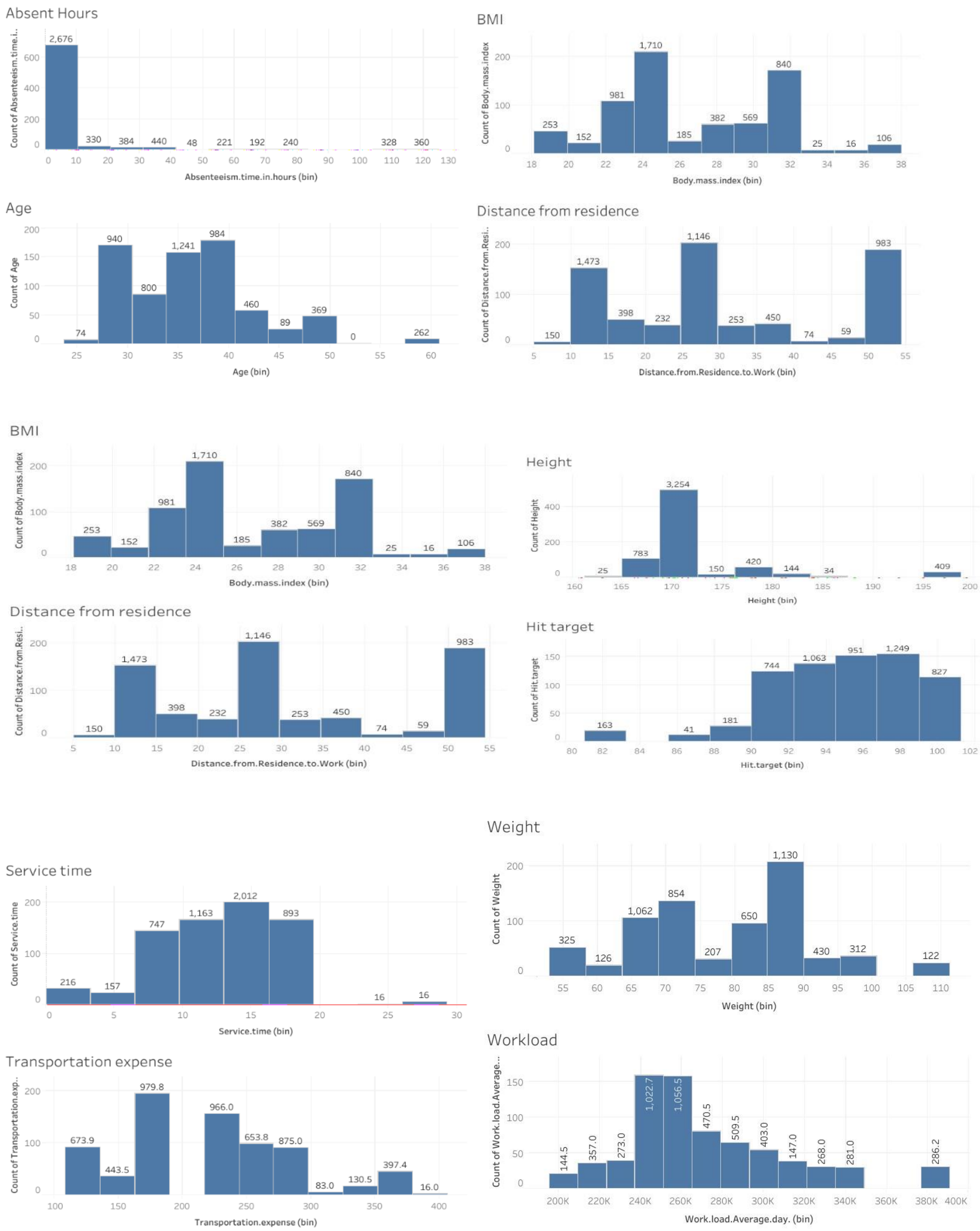
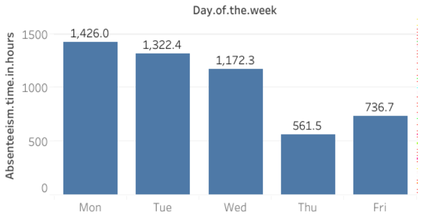


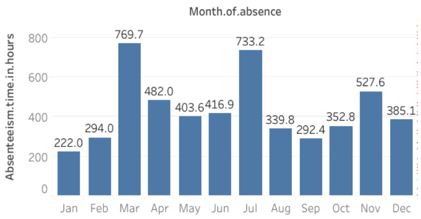
Fig 2.4 – Distribution of Continuous variables using Histogram

Day of the Week

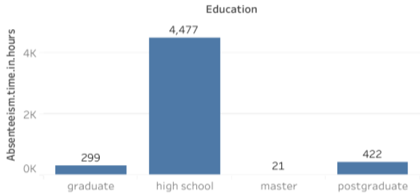


Discipline

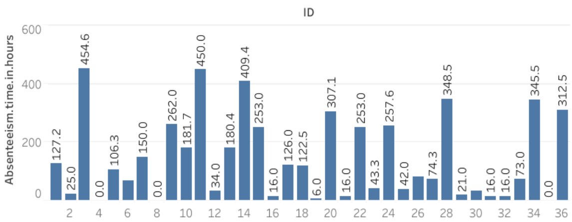
Month of Absence



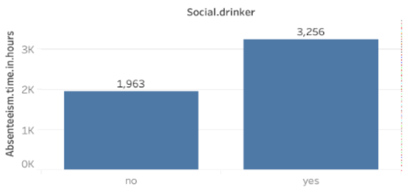
Education



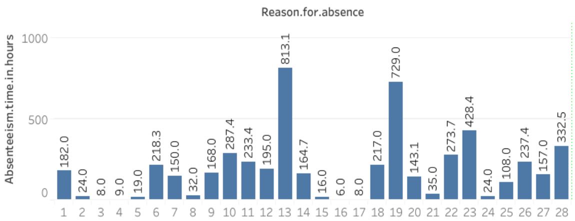
ID



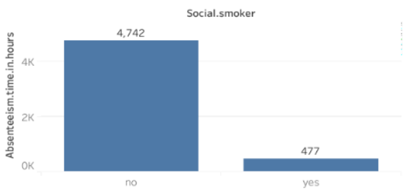
Social Drinker



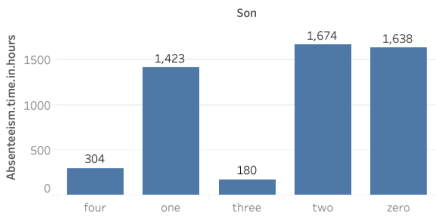
Reason for Absence



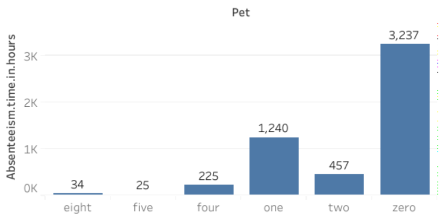
Social Smoker



Sons



Pets



Seasons

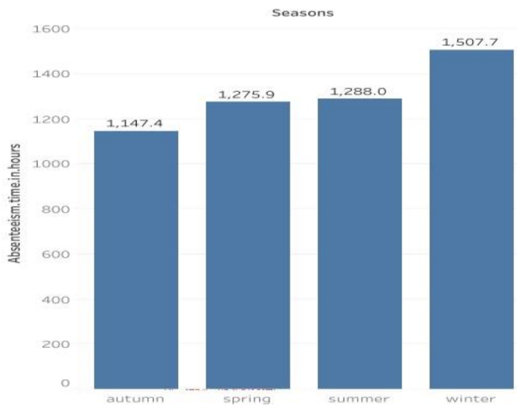


Fig 2.5 – Distribution of Categorical variables using Bar graph

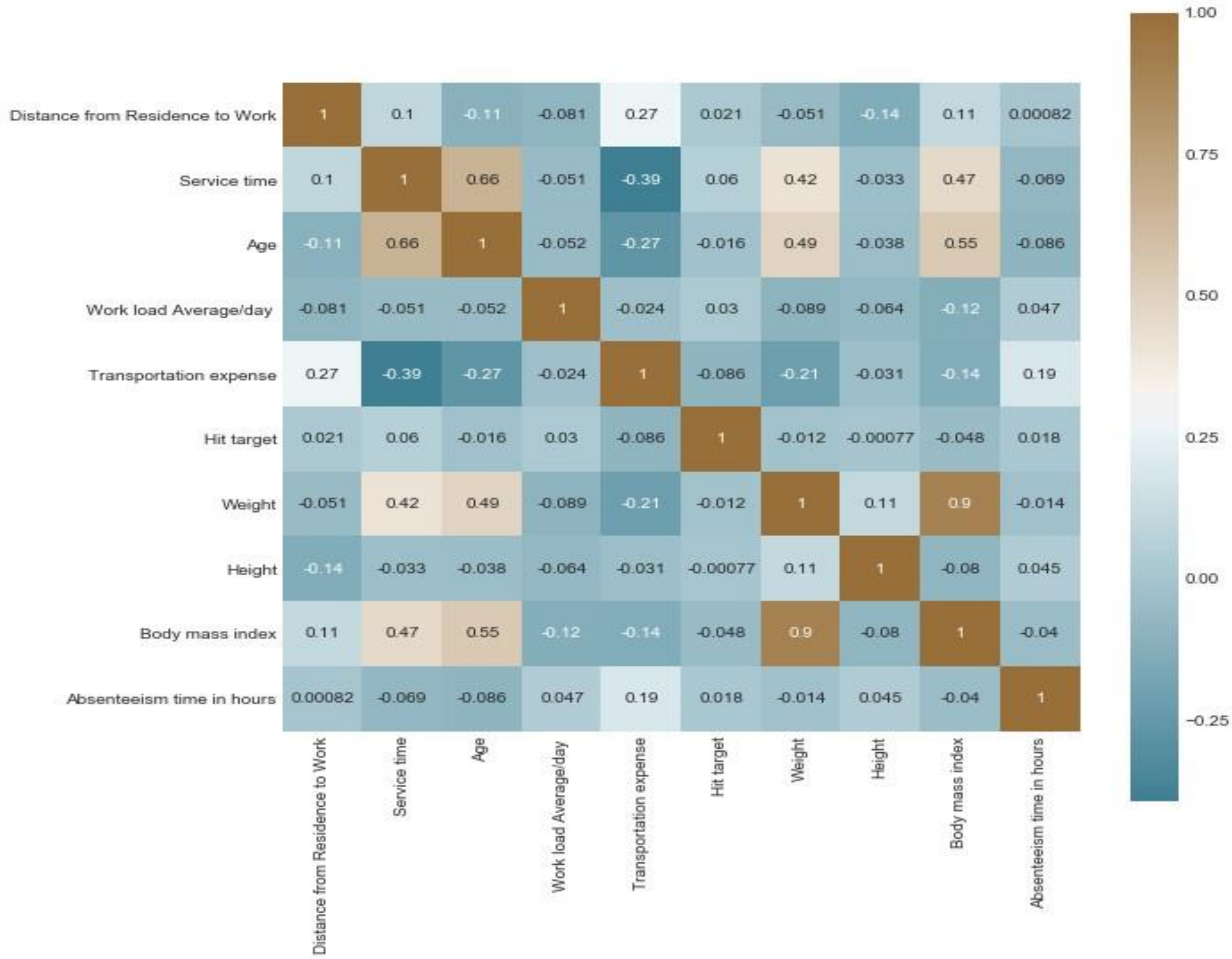


Fig 2.6 – Correlation plot of Continuous variables

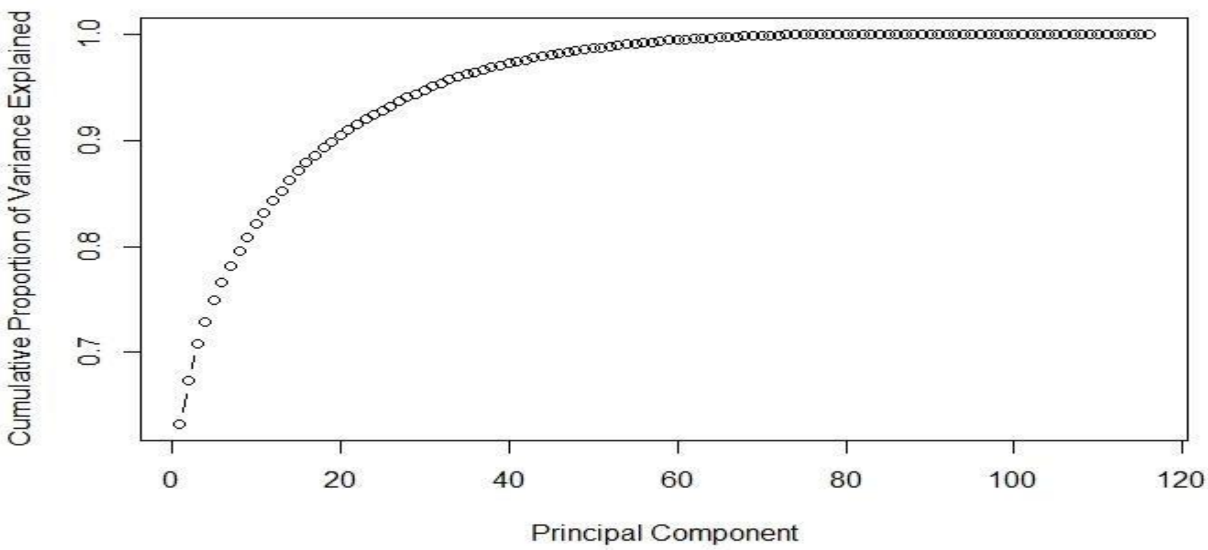


Fig 2.8 – Cumulative Scree Plot of Principal Components

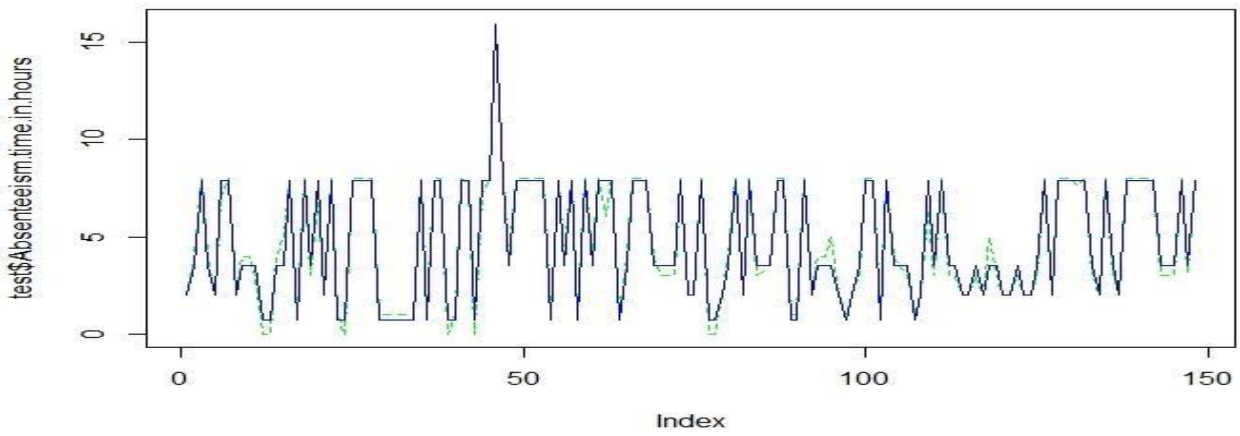


Fig 3.2 – Plot of actual values vs predicted values for Decision Tree

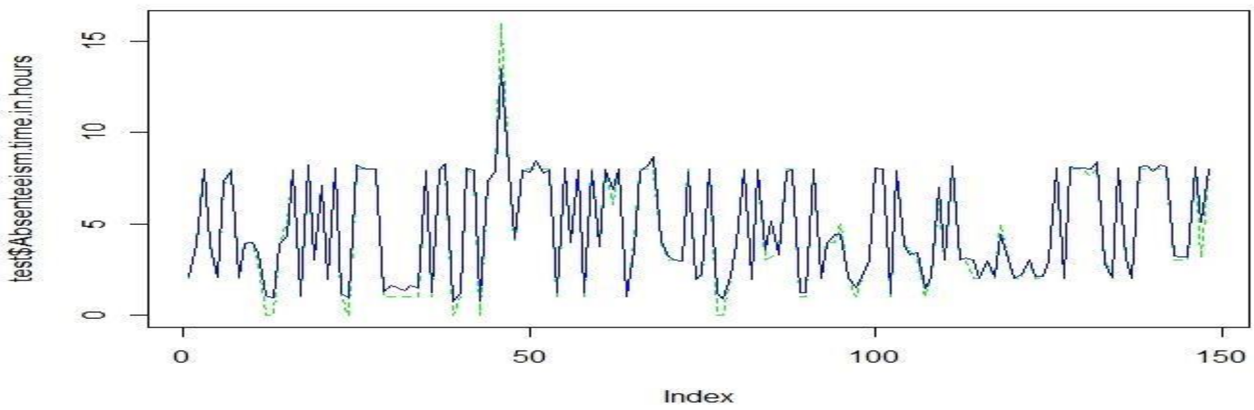


Fig 3.3 – Plot of actual values vs predicted values for Random Forest

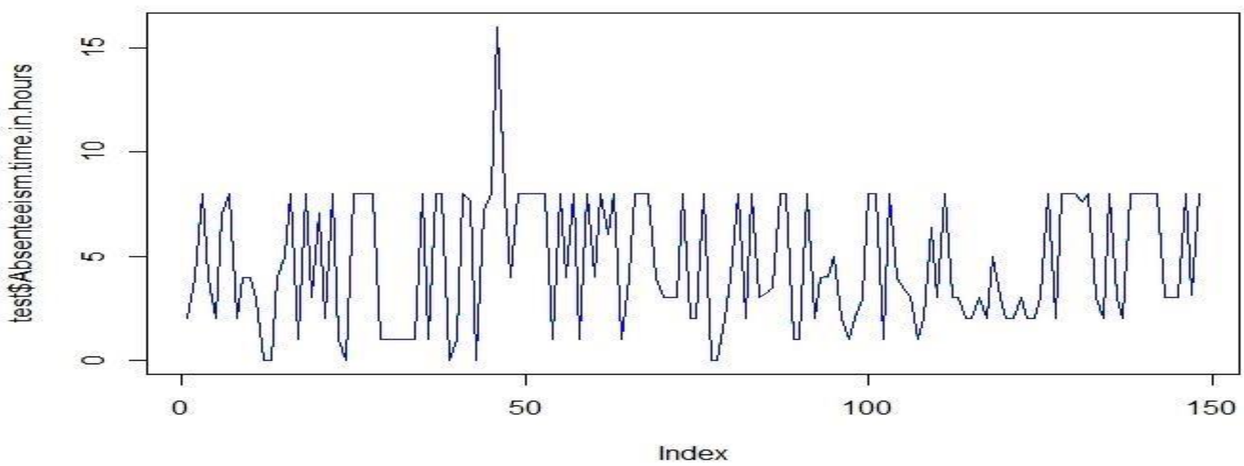


Fig 3.5 – Plot of actual values vs predicted values for Linear Regression

Chapter 6: R code

```
#Read the csv file
employ = read.xlsx(file = "Absenteeism_at_work_Project.xls", header = T,
sheetIndex = 1)

#Exploration of data
#To check number of rows and columns
dim(employ)

#To observe top 5 rows
head(employ)

#Structure of variables
str(employ)

#Transform data types
employ$ID = as.factor(as.character(employ$ID))

employ$Reason.for.absence[employ$Reason.for.absence %in% 0] = 20

employ$Reason.for.absence =
as.factor(as.character(employ$Reason.for.absence))

employ$Month.of.absence[employ$Month.of.absence %in% 0] = NA

employ$Month.of.absence = as.factor(as.character(employ$Month.of.absence))

employ$Day.of.the.week = as.factor(as.character(employ$Day.of.the.week))

employ$Seasons = as.factor(as.character(employ$Seasons))

employ$Disciplinary.failure =
as.factor(as.character(employ$Disciplinary.failure))

employ$Education = as.factor(as.character(employ$Education))

employ$Son = as.factor(as.character(employ$Son))

employ$Social.drinker = as.factor(as.character(employ$Social.drinker))

employ$Social.smoker = as.factor(as.character(employ$Social.smoker))
```

```
employ$Pet = as.factor(as.character(employ$Pet))

#Structure of variables
str(employ)

#Make a copy of data
df = employ

###########################Missing_value_analysis###############################
#Get number of missing values
sapply(df,function(x){sum(is.na(x))})
missing_values = data.frame(sapply(df,function(x){sum(is.na(x))}))

#Get the rownames as new column
missing_values$Variables = row.names(missing_values)

#Reset the row names
row.names(missing_values) = NULL

#Rename the column
names(missing_values)[1] = "Miss_perc"

#Calculate missing percentage
missing_values$Miss_perc = ((missing_values$Miss_perc/nrow(employ)) *100)

#Reorder the columns
missing_values = missing_values[,c(2,1)]

#Sort the rows according to decreasing missing percentage
missing_values = missing_values[order(-missing_values$Miss_perc),]

#Create a bar plot to visualie top 5 missing values
library(ggplot2)
ggplot(data = missing_values[1:5,], aes(x=reorder(Variables, -Miss_perc),y = 
Miss_perc))+
  geom_bar(stat = "identity",fill = "grey")+xlab("Parameter")+
  ggtitle("Missing_data_percentage") + theme_bw()

#Create missing value and impute using mean, median and knn
#Value = 31
#Mean = 26.67
#Median = 25
#KNN = 31
```

```
library(DMwR)
df = knnImputation(data = df, k = 5)

#Check if any missing values
sum(is.na(df))

# Saving output result into excel file
write.xlsx(missing_values, "Miss_percentage_R.xlsx", row.names = F)

#Explore distribution using graphs
#Get numerical data
numeric_index = sapply(df, is.numeric)
numeric_data = df[,numeric_index]

#Distribution of factor data using bar plot
bar1 = ggplot(data = df, aes(x = ID)) + geom_bar() + ggtitle("Count_of_ID") +
theme_bw()
bar2 = ggplot(data = df, aes(x = Reason.for.absence)) + geom_bar() +
  ggtitle("Count_of_Reason_for_absence") + theme_bw()
bar3 = ggplot(data = df, aes(x = Month.of.absence)) + geom_bar() +
ggtitle("Count_of_Month") + theme_bw()
bar4 = ggplot(data = df, aes(x = Disciplinary.failure)) + geom_bar() +
  ggtitle("Count_of_Disciplinary_failure") + theme_bw()
bar5 = ggplot(data = df, aes(x = Education)) + geom_bar() +
ggtitle("Count_of_Education") + theme_bw()
bar6 = ggplot(data = df, aes(x = Son)) + geom_bar() + ggtitle("Count_of_Son") +
theme_bw()
bar7 = ggplot(data = df, aes(x = Social.smoker)) + geom_bar() +
  ggtitle("Count_of_Social_smoker") + theme_bw()

gridExtra::grid.arrange(bar1,bar2,bar3,bar4,ncol=2)
gridExtra::grid.arrange(bar5,bar6,bar7,ncol=2)

#Check the distribution of numerical data using histogram
hist1 = ggplot(data = numeric_data, aes(x =Transportation.expense)) +
  ggtitle("Transportation.expense") + geom_histogram(bins = 25)
hist2 = ggplot(data = numeric_data, aes(x =Height)) +
  ggtitle("Distribution_of_Height") + geom_histogram(bins = 25)
hist3 = ggplot(data = numeric_data, aes(x =Body.mass.index)) +
  ggtitle("Distribution_of_Body.mass.index") + geom_histogram(bins = 25)
hist4 = ggplot(data = numeric_data, aes(x =Absenteeism.time.in.hours)) +
  ggtitle("Distribution_of_Absenteeism.time.in.hours") + geom_histogram(bins =
25)
```

```
gridExtra::grid.arrange(hist1,hist2,hist3,hist4,ncol=2)

##############################Outlier_Analysis####################################
#Get the data with only numeric columns
numeric_index = sapply(df, is.numeric)
numeric_data = df[,numeric_index]

#Get the data with only factor columns
factor_data = df[,!numeric_index]

#Check for outliers using boxplots
for(i in 1:ncol(numeric_data)) {
  assign(paste0("box",i), ggplot(data = df, aes_string(y = numeric_data[,i])) +
         stat_boxplot(geom = "errorbar", width = 0.5) +
         geom_boxplot(outlier.colour = "red", fill = "grey", outlier.size = 1) +
         labs(y = colnames(numeric_data[i])) +
         ggtitle(paste("Boxplot: ",colnames(numeric_data[i]))))
}

#Arrange the plots in grids
gridExtra::grid.arrange(box1,box2,box3,box4,ncol=2)
gridExtra::grid.arrange(box5,box6,box7,box8,ncol=2)
gridExtra::grid.arrange(box9,box10,ncol=2)

#Get the names of numeric columns
numeric_columns = colnames(numeric_data)

#Replace all outlier data with NA
for(i in numeric_columns){
  val = df[,i][df[,i] %in% boxplot.stats(df[,i])$out]
  print(paste(i,length(val)))
  df[,i][df[,i] %in% val] = NA
}

#Check number of missing values
sapply(df,function(x){sum(is.na(x))})

#Get number of missing values after replacing outliers as NA
missing_values_out = data.frame(sapply(df,function(x){sum(is.na(x))}))
missing_values_out$Columns = row.names(missing_values_out)
row.names(missing_values_out) = NULL
names(missing_values_out)[1] = "Miss_perc"
missing_values_out$Miss_perc = ((missing_values_out$Miss_perc/nrow(employ))
*100)
```

```
missing_values_out = missing_values_out[,c(2,1)]
missing_values_out = missing_values_out[order(-
missing_values_out$Miss_perc),]
missing_values_out

#Compute the NA values using KNN imputation
df = knnImputation(df, k = 5)

#Check if any missing values
sum(is.na(df))

############################Feature_Selection####################################
#Check for multicollinearity using VIF
library(VIF)
library(usdm)
vifcor(numeric_data)

#Check for multicollinearity using correlation graph
library(corrgram)
corrgram(numeric_data, order = F, upper.panel=panel.pie,
        text.panel=panel.txt, main = "Correlation Plot")

#Variable Reduction
df = subset.data.frame(df, select = -c(Body.mass.index))

#Make a copy of Clean Data
library(xlsx)
clean_data = df
write.xlsx(clean_data, "clean_data.xlsx", row.names = F)

################################Feature_Scaling##################################
#Normality check
hist(df$Absenteeism.time.in.hours)

#Remove dependent variable
numeric_index = sapply(df,is.numeric)
numeric_data = df[,numeric_index]
numeric_columns = names(numeric_data)
numeric_columns = numeric_columns[-9]

#Normalization of continuous variables
for(i in numeric_columns){
  print(i)
  df[,i] = (df[,i] - min(df[,i]))/
```

```
  (max(df[,i]) - min(df[,i]))
}

#Get the names of factor variables
factor_columns = names(factor_data)

#Create dummy variables of factor variables
library(dummies)
df = dummy.data.frame(df, factor_columns)

#To remove the object names excluding the required objects
rm(list=ls()[! ls() %in% c("df","employ")])

#################################DECISION_TREE###################################
#RMSE: 2.276
#MAE: 1.694
#R squared: 0.44

#Splitting data into train and test data
set.seed(1)
train_index = sample(1:nrow(df), 0.8*nrow(df))
train = df[train_index,]
test = df[-train_index,]

#Build decsion tree using rpart
dt_model = rpart(Absenteeism.time.in.hours ~ ., data = train, method = "anova")

#Plot the tree
rpart.plot(dt_model)

#Predict for test cases
dt_predictions = predict(dt_model, test[,-115])

#Create data frame for actual and predicted values
df_pred = data.frame("actual"=test[,115], "dt_pred"=dt_predictions)
head(df_pred)

#Calculate MAE, RMSE, R-sqaured for testing data
print(postResample(pred = dt_predictions, obs = test[,115]))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism.time.in.hours,type="l",lty=2,col="green")
lines(dt_predictions,col="blue")
```

```
##############################RANDOM_FOREST######################################
#RMSE: 2.194
#MAE: 1.61
#R squared: 0.479

##Train the model using training data
library(randomForest)
rf_model = randomForest(Absenteeism.time.in.hours~., data = train, ntree = 500)

#Predict the test cases
rf_predictions = predict(rf_model, test[,-115])

#Create dataframe for actual and predicted values
df_pred = cbind(df_pred,rf_predictions)
head(df_pred)

#Calculate MAE, RMSE, R-sqaured for testing data
library(caret)
print(postResample(pred = rf_predictions, obs = test[,115]))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism.time.in.hours,type="l",lty=2,col="green")
lines(rf_predictions,col="blue")

#RMSE: 2.194
#MAE: 1.61
#R squared: 0.479

##Train the model using training data
rf_model = randomForest(Absenteeism.time.in.hours~., data = train, ntree = 500)

#Predict the test cases
rf_predictions = predict(rf_model, test[,-115])

#Create dataframe for actual and predicted values
df_pred = cbind(df_pred,rf_predictions)
head(df_pred)

#Calculate MAE, RMSE, R-sqaured for testing data
print(postResample(pred = rf_predictions, obs = test[,115]))

#Plot a graph for actual vs predicted values
```

```
plot(test$Absenteeism.time.in.hours,type="l",lty=2,col="green")
lines(rf_predictions,col="blue")

#############################LINEAR_REGRESSION##################################
#RMSE: 2.559
#MAE: 1.86
#R squared: 0.358

##Train the model using training data
lr_model = lm(formula = Absenteeism.time.in.hours~., data = train)

#Get the summary of the model
summary(lr_model)

#Predict the test cases
lr_predictions = predict(lr_model, test[,-115])

#Create dataframe for actual and predicted values
df_pred = cbind(df_pred,lr_predictions)
head(df_pred)

#Calculate MAE, RMSE, R-sqaured for testing data
print(postResample(pred = lr_predictions, obs = test[,115]))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism.time.in.hours,type="l",lty=2,col="green")
lines(lr_predictions,col="blue")

########################DIMENSION_REDUCTION_USING_PCA#####################
####
#Principal component analysis
prin_comp = prcomp(train)

#Compute standard deviation of each principal component
pr_stdev = prin_comp$sdev

#Compute variance
pr_var = pr_stdev^2

#Proportion of variance explained
prop_var = pr_var/sum(pr_var)

#Cumulative screen plot
plot(cumsum(prop_var), xlab = "Principal Component",
```

```
    ylab = "Cumulative Proportion of Variance Explained",
    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 45 components since it explains almost 95+ %
data variance
train.data =train.data[,1:45]

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

#Select the first 45 components
test.data=test.data[,1:45]

##########################DECISION_TREE########################################
#RMSE: 0.442
#MAE: 0.301
#R squared: 0.978

#Build decision tree using rpart
dt_model = rpart(Absenteeism.time.in.hours ~., data = train.data, method =
"anova")

#Predict the test cases
dt_predictions = predict(dt_model,test.data)

#Create data frame for actual and predicted values
df_pred = data.frame("actual"=test[,115], "dt_pred"=dt_predictions)
head(df_pred)

#Calculate MAE, RMSE, R-sqaured for testing data
print(postResample(pred = dt_predictions, obs =
test$Absenteeism.time.in.hours))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism.time.in.hours,type="l",lty=2,col="green")
lines(dt_predictions,col="blue")

###############################RANDOM_FOREST###################################
#RMSE: 0.480
```

```
#MAE: 0.264
#R squared: 0.978

#Train the model using training data
rf_model = randomForest(Absenteeism.time.in.hours~., data = train.data, ntrees
= 500)

#Predict the test cases
rf_predictions = predict(rf_model,test.data)

#Create dataframe for actual and predicted values
df_pred = cbind(df_pred,rf_predictions)
head(df_pred)

#Calculate MAE, RMSE, R-squared for testing data
print(postResample(pred = rf_predictions, obs = test$Absenteeism.time.in.hours))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism.time.in.hours,type="l",lty=2,col="green")
lines(rf_predictions,col="blue")

############################LINEAR_REGRESSION###################################
#RMSE: 0.003
#MAE: 0.002
#R squared: 0.999

#Train the model using training data
lr_model = lm(Absenteeism.time.in.hours ~ ., data = train.data)

#Get the summary of the model
summary(lr_model)

#Predict the test cases
lr_predictions = predict(lr_model,test.data)

#Create dataframe for actual and predicted values
df_pred = cbind(df_pred,lr_predictions)
head(df_pred)

#Calculate MAE, RMSE, R-squared for testing data
print(postResample(pred = lr_predictions, obs =test$Absenteeism.time.in.hours))

#Plot a graph for actual vs predicted values
plot(test$Absenteeism.time.in.hours,type="l",lty=2,col="green")
```

lines(lr_predictions,col="blue")