Employee Absenteeism Project

Name: Ganesh Ramachandran S

Date: 13th January 2020

Contents

Codes

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 it dataset and requested to have an answer on the following areas:

1. What changes the company should bring to reduce the number of absenteeism?
2. How much losses every month can we project in 2011, if a same trend of absenteeism continuous?

1.2 Data

The sample data used for analysis has 21 variables in which 20 are dependent and 1 is dependent variable (Absentees time in hours). Since target variable is continuous, we can use regression-based machine learning analysis.

Attribute information:

1. Individual identification (ID)
2. Reason for absence (ICD)

   Absences attested by the 'International Code of Disease' (ICD) stratified into 21 categories (From I to XXI) as follows:

   I. Certain infectious and Parasitic diseases
   II. Neoplasms
   III. Diseases of 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. Disease 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 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.

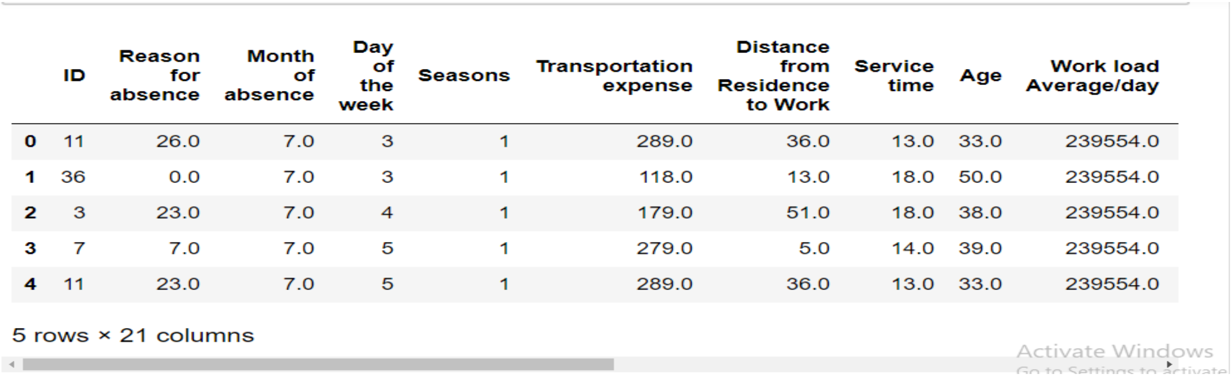
And 7 categories without (CID) patient follow-up (22), medical consultation (23), blood donation (24), laboratory examination (25), unjustified absence (26), physiotherapy (27), dental consultation (28).

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 expenses
7. Distance form 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 Sample data

1. The data used for analysis has 740 observation 21 variables, some of the observation has missing values, all the variables are in numerical in class, some of the variables has set of numeric codes with character valued levels such as Seasons (summer (1), autumn (2), winter (3), spring (4))

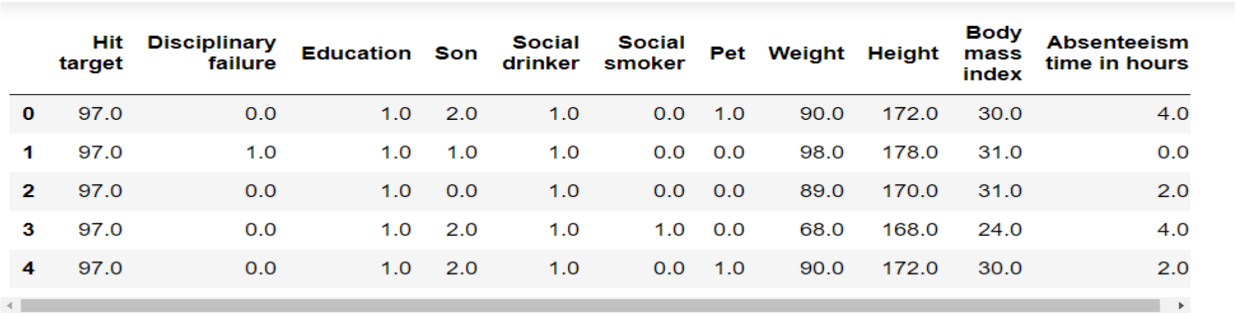
Figure - 1.3.1 (First 5 rows of columns no: 1 to 10)



	ID	Reason for absence	Month of absence	Day of the week	Seasons	Transportation expense	Distance from Residence to Work	Service time	Age	Work load Average/day
0	11	26.0	7.0	3	1	289.0	36.0	13.0	33.0	239554.0
1	36	0.0	7.0	3	1	118.0	13.0	18.0	50.0	239554.0
2	3	23.0	7.0	4	1	179.0	51.0	18.0	38.0	239554.0
3	7	7.0	7.0	5	1	279.0	5.0	14.0	39.0	239554.0
4	11	23.0	7.0	5	1	289.0	36.0	13.0	33.0	239554.0

5 rows × 21 columns

Figure – 1.3.2 (First 5 rows of columns no: 11 to 21)



	Hit target	Disciplinary failure	Education	Son	Social drinker	Social smoker	Pet	Weight	Height	Body mass index	Absenteeism time in hours
0	97.0	0.0	1.0	2.0	1.0	0.0	1.0	90.0	172.0	30.0	4.0
1	97.0	1.0	1.0	1.0	1.0	0.0	0.0	98.0	178.0	31.0	0.0
2	97.0	0.0	1.0	0.0	1.0	0.0	0.0	89.0	170.0	31.0	2.0
3	97.0	0.0	1.0	2.0	1.0	1.0	0.0	68.0	168.0	24.0	4.0
4	97.0	0.0	1.0	2.0	1.0	0.0	1.0	90.0	172.0	30.0	2.0

2. These variables need to be transformed into numerical, factor and categorical variables based on the model requirements.

Chapter 2

Pre-processing

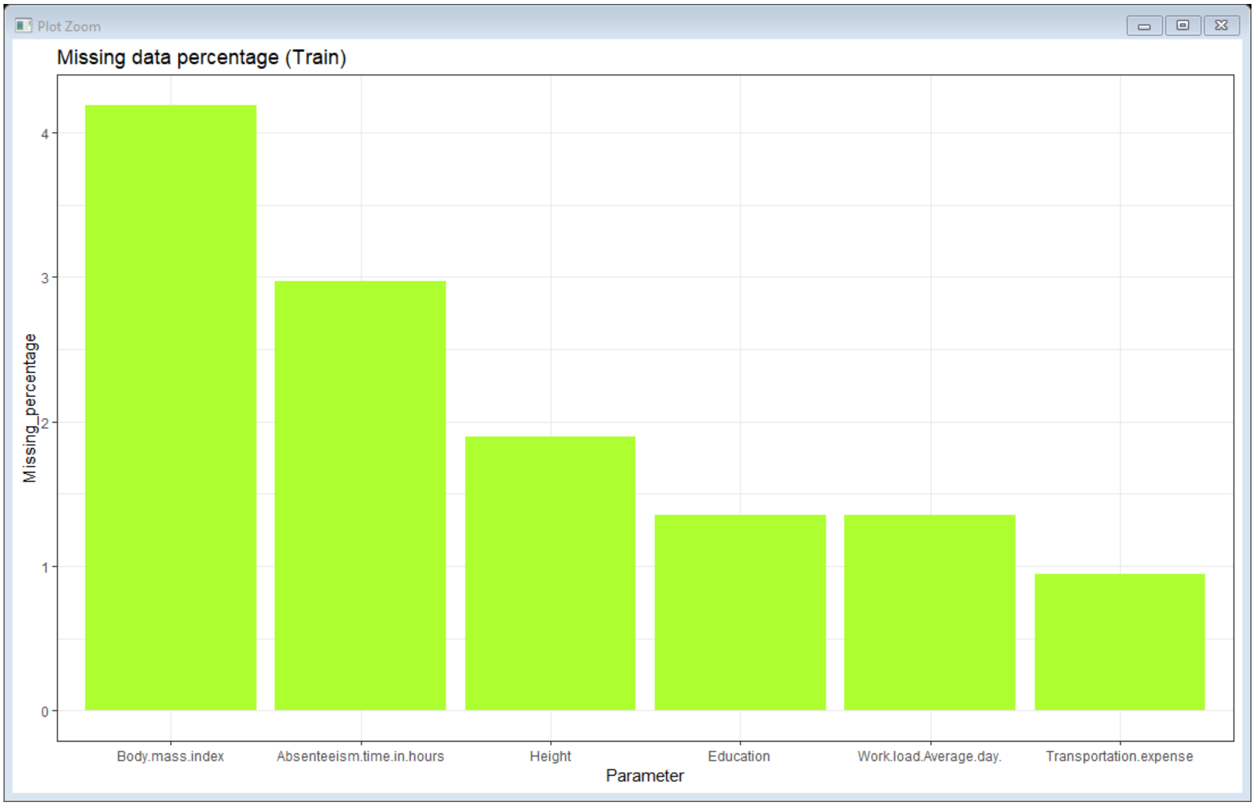
2.1 Exploratory data analysis

This data has missing values as well as two variables has ‘0’ (zero) for numerical class with character value levels, as per given data attribute information, there are no ‘0’ for 2nd variable “Reason for absence” and 3rd variable “Month of absence”. Data transformation has done based the summary statistics information. Categorical variables which has numerical values are also transformed.

2.2 Missing value analysis

This data has 21 variable and 740 observations, some the data are missing, it may be due to human error, refuse to answer while surveying and optional box in the questionnaire.

The missing values are understood by using graphs, if each variable missing percentage is more than 30%, we drop variable or observation depends on the data. In this data missing value percentage is less than 30%, we will try to impute by using mean, median and KNN method. Which imputation method gives nearest possible result in actual value would be chosen for the model. In this case KNN gives impute values with better accuracy compared to mean and median.



This data has missing value percentage from 0% to 4.18%. it is less than 30% missing value; hence we will go for imputation methods to replace NA in observation. In this missing value imputation mean, median and KNN imputation methods are used. From all three methods KNN imputation gives better imputed value. The K = 3 is used for missing value imputation method

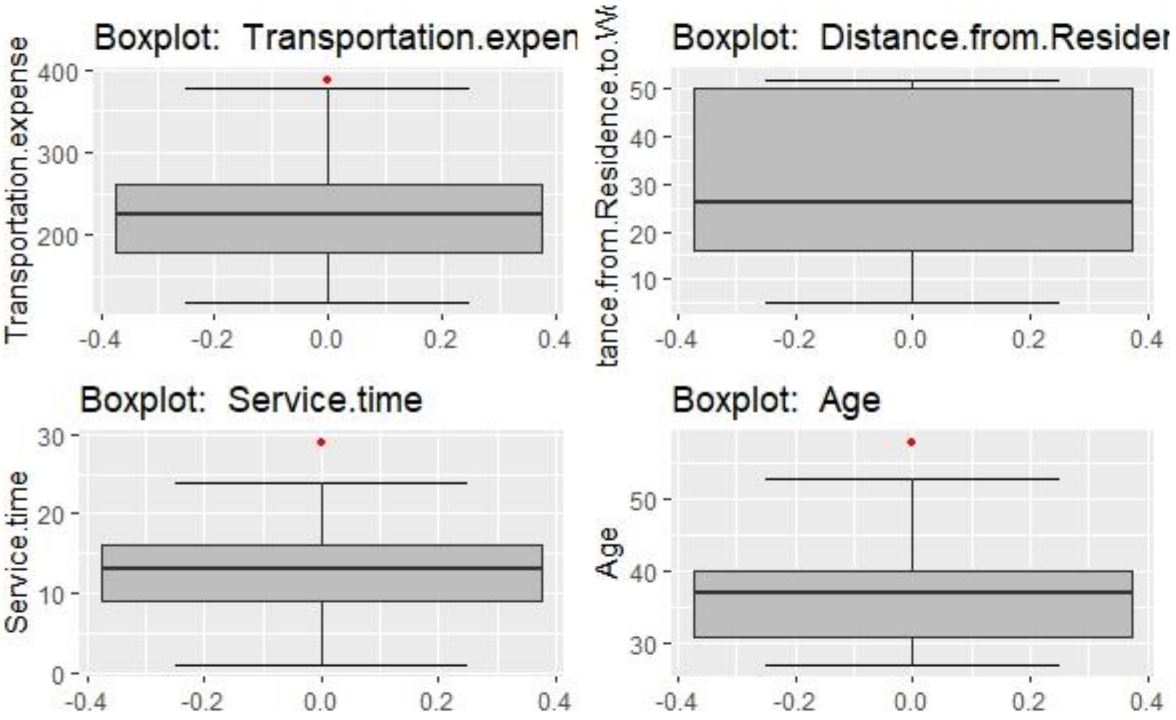
2.3 Outlier analysis

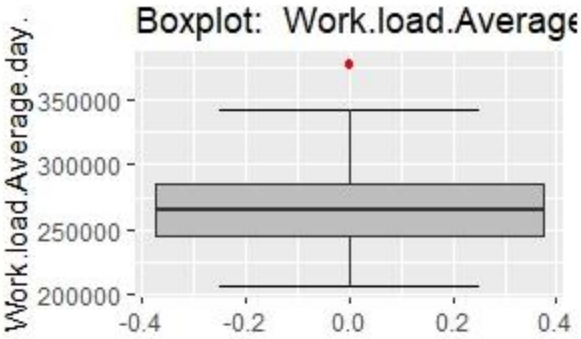
Outlier are observations inconsistent with the rest of the dataset, it is due to poor data quality, low quality measurement, malfunctioning equipment and manual error. Sometime data may be correct, but exceptional data. In statistical central tendency measures, the mean is more sensitive outliers compared to median measure.

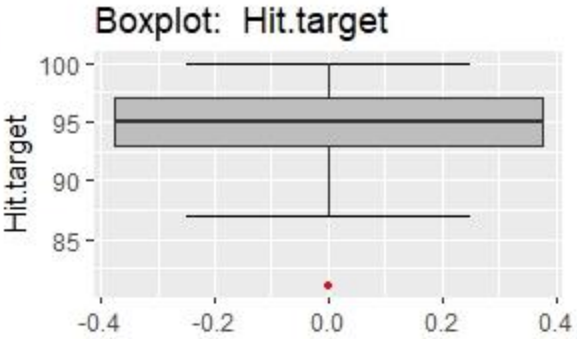
Outliers are detected by box plot (Graphical tool), The Grubbs' test for Outliers, Outliers are detected with the help of R and Python tools, based on the data, Outliers would be removed or replaced as a missing value. These missing values are imputed with better method.

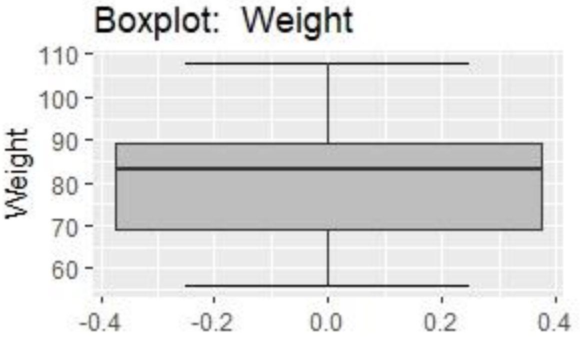
In this data Box plot has been used to identify the Outliers in continuous variables, Continuous variables used for analysis are listed below:

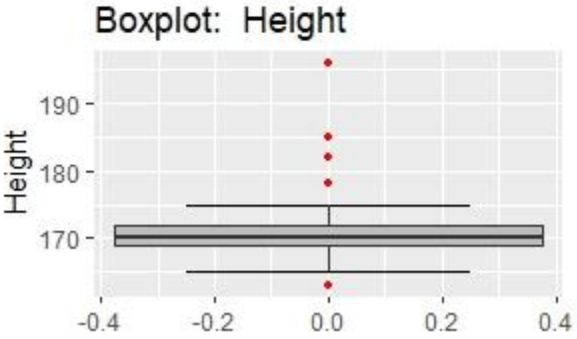
1.Transportation expense, 2. Distance from residence to work, 3. Service time, 4. Age,
5. Work load average /day, 6. Hit target, 7. Weight, 8. Height, 9. Body mass index,
10. Absenteeism time in hours

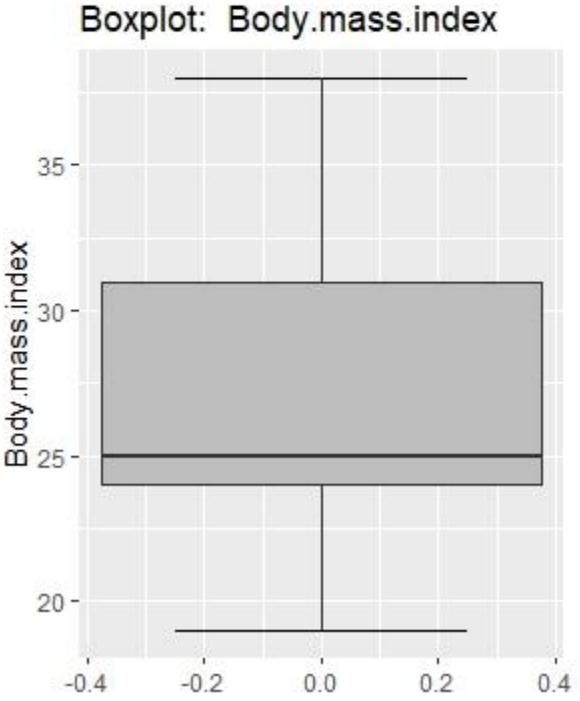


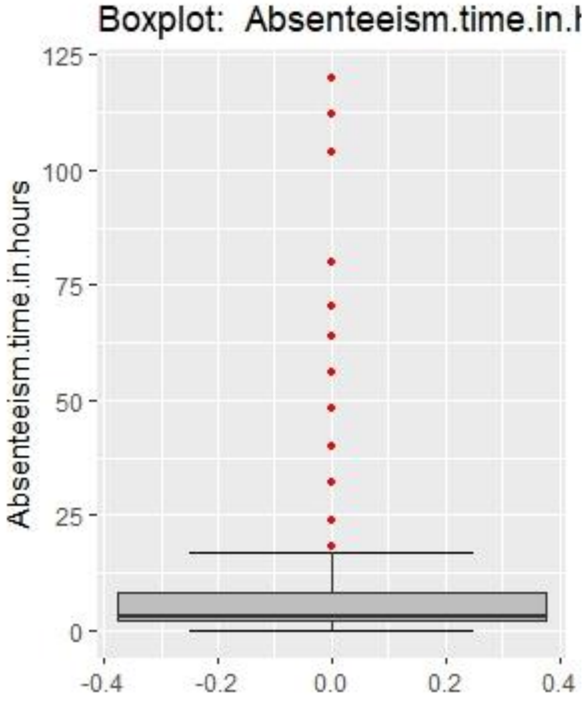






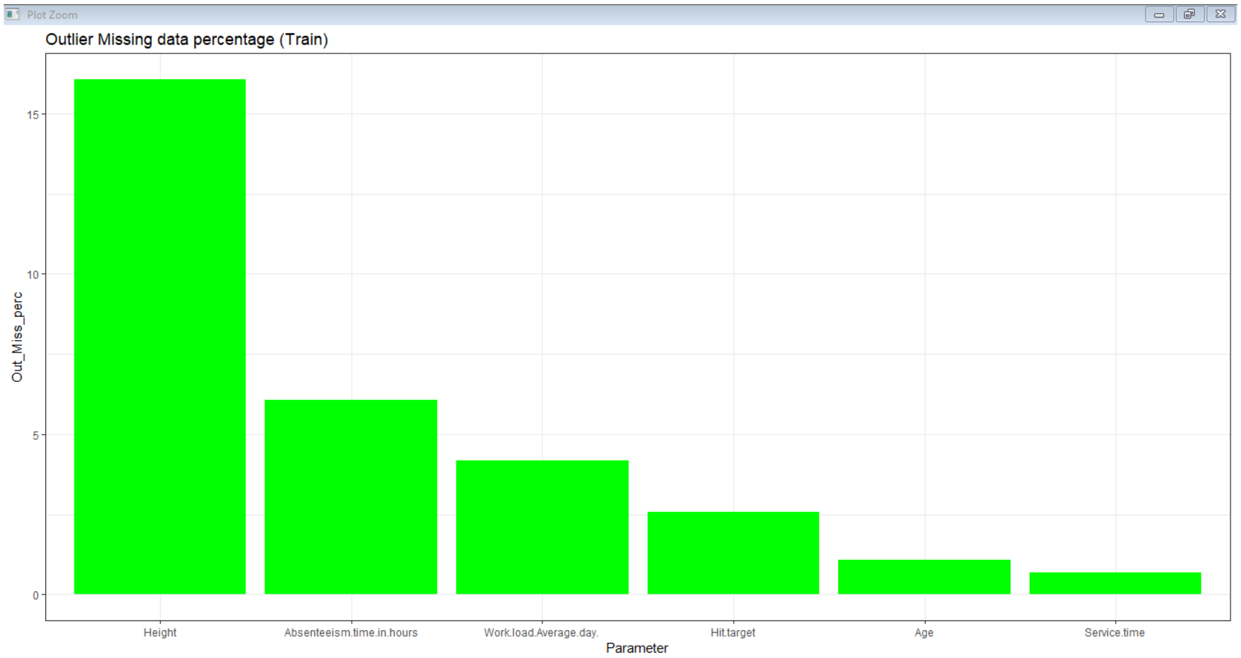






Based on the box plot variables such as Distance from residence to work, Weight and Body mass index has no outliers.

Remaining all the seven continuous variables has outlier, these Outliers are replaced with NA. The ‘Height variable has missing value percentage 16.08%, which is highest missing value percentage and other 6 continuous variables are missing value percentage from 6% to 0.4% in descending order.



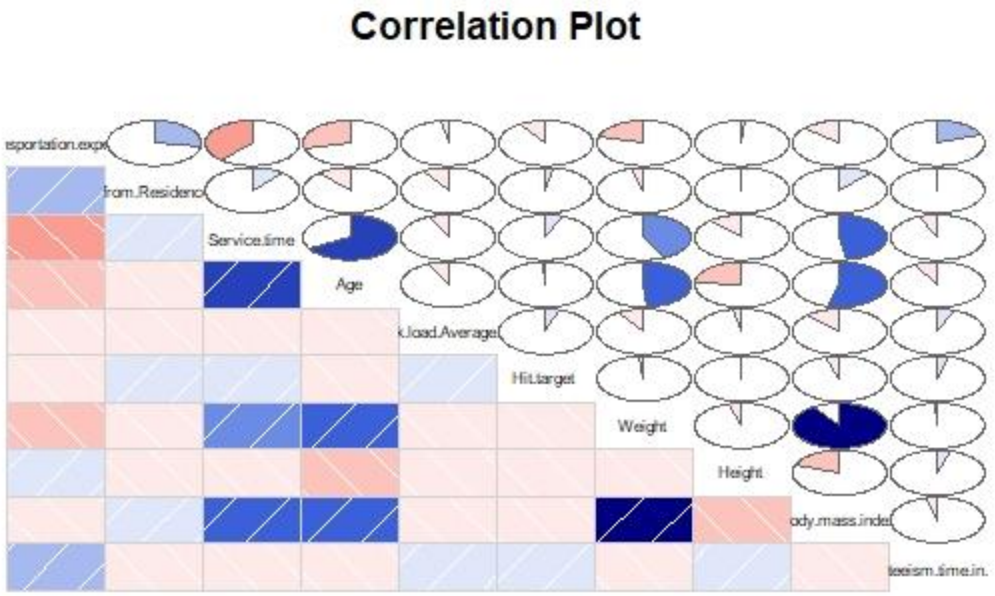
Height > Absenteeism time in hours > Work load average /day > Hit target > Age > Service time > Transportation expense.

The KNN imputation method is used to impute outlier in this analysis. The K = 3 is used for missing value imputation method

2.4 Feature selection

- Feature selection/Variable importance is selecting a subset of relevant feature (variable/predictor) for use in model construction.
- Data subset of a learning algorithm’s input variable upon which it should focus attention, while ignoring the rest.
- It is useful for dimensionality reduction, few variables are highly correlated to each other, it will reduce the model accuracy, for dimensionality reduction correlation analysis used for continuous variables and PCA is used.

In this analysis Correlation plot is used for the continuous variables to check highly correlated variables, we have found that ‘Height” variable correlated with another independent variable. Hence, the ‘Height’ is dropped from the model. It has a correlation of 0.90 and impact the model performance



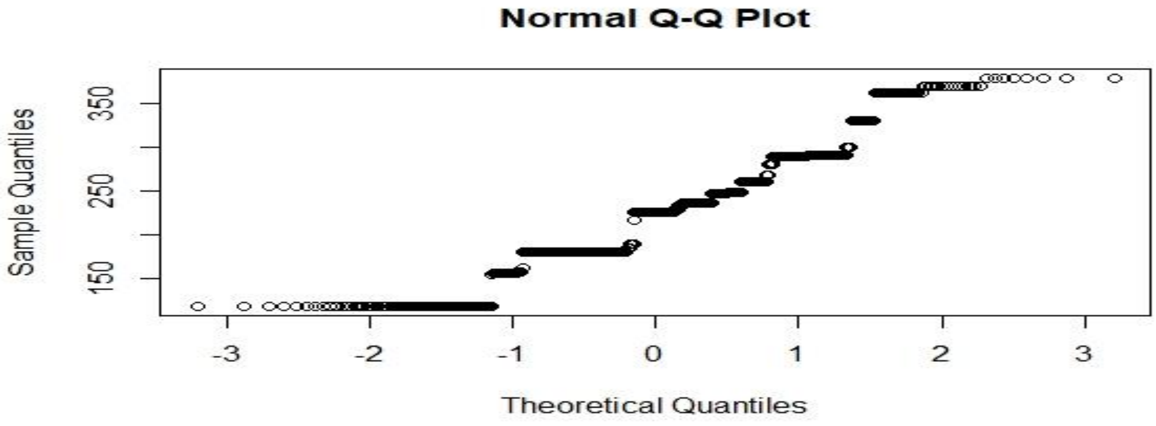
2.5 Feature scaling

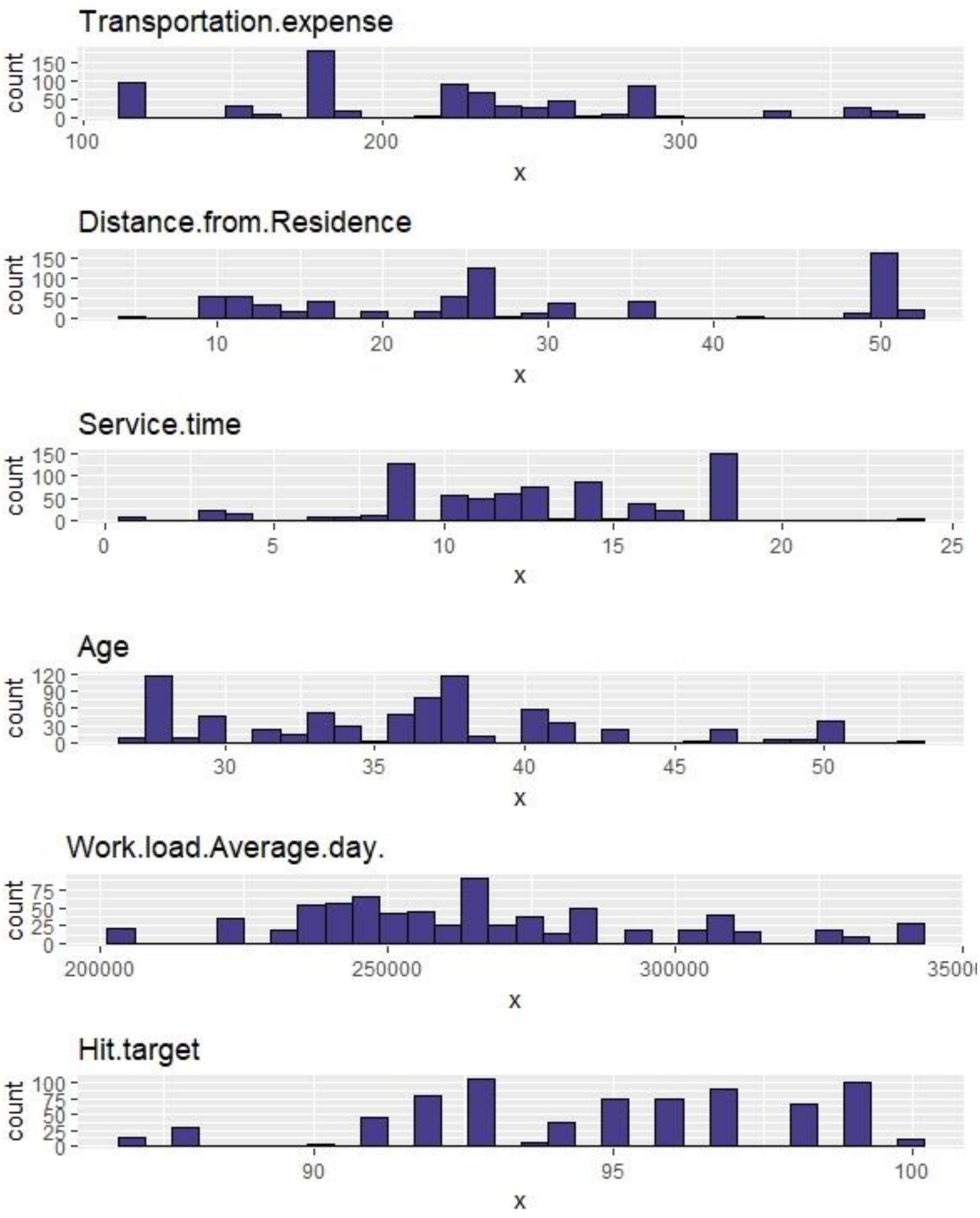
Feature scaling involves normalization and standardization processing of the data.

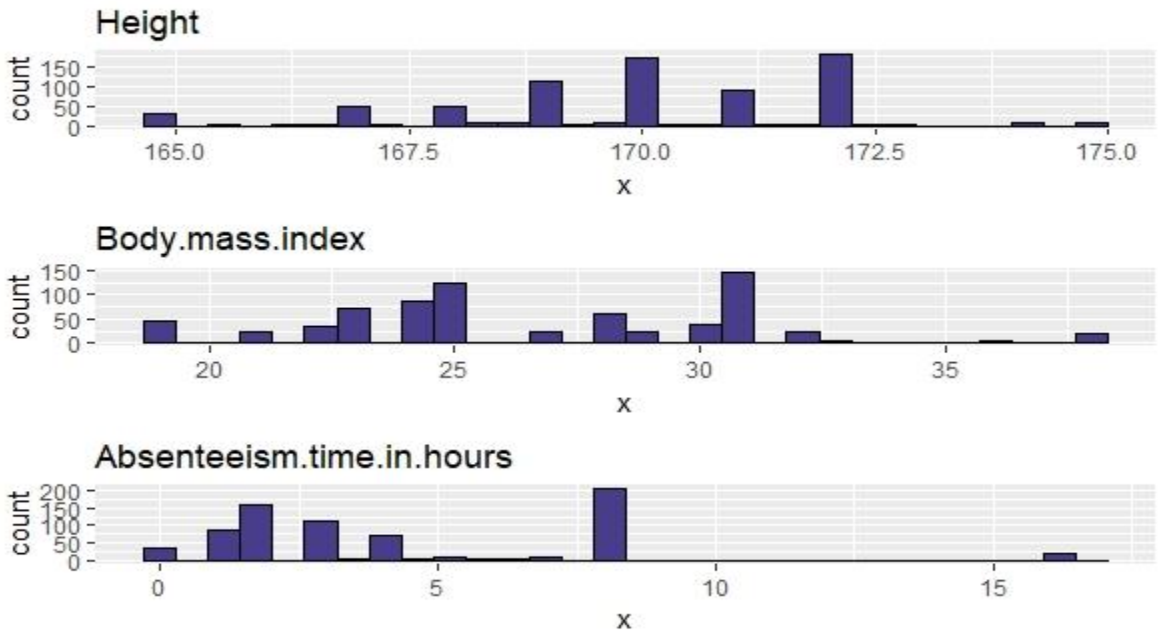
Normalization – it is a database design technique which reduces redundancy and dependency of data. It is used to bring data variables to a common scale into proportion with one another in a range between 0 to 1. If data are not uniformly distributed, will for normalization technique.

Standardization – it works well if the data is uniformly distributed, data points are expressed in positive and negative standard deviation. It shows how many units away from the mean.

The Normal Q-Q plot and histogram on continuous variables shows that the data is non-uniformly distributed. For this analysis normalization technique is used to bring all the data in a common scale.




Transportation.expense
count
150
100
50
0
100
200
300
x
Distance.from.Residence
count
150
100
50
0
10
20
30
40
50
x
Service.time
count
150
100
50
0
0
5
10
15
20
25
x
Age
count
120
90
60
30
0
30
35
40
45
50
x
Work.load.Average.day.
count
75
50
25
0
200000
250000
300000
3500
x
Hit.target
count
100
75
50
25
0
90
95
100
x



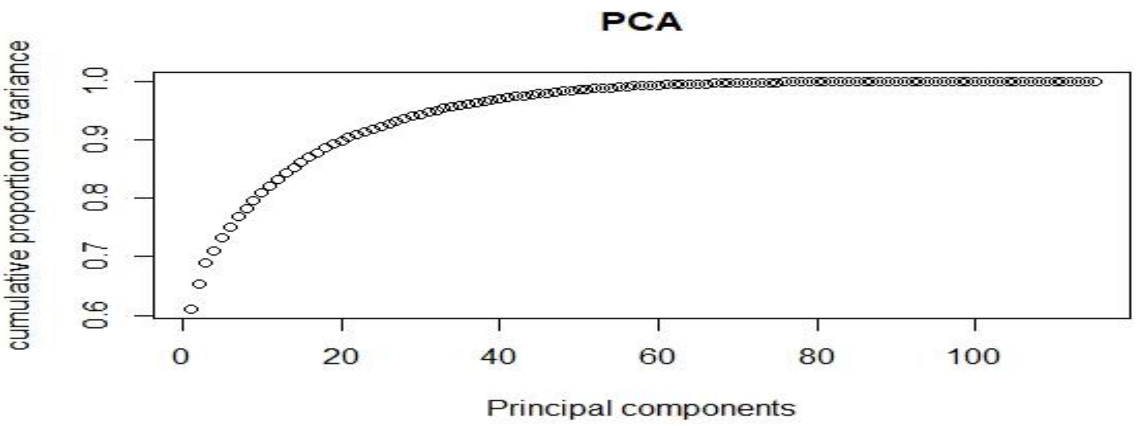
2.6 Dummy variables

This data set has 11 categorical variables are in numeric form, our target variable ‘Absenteeism time in hours ‘are continuous, we will develop model related to regression, Hence, all the categorical variables transformed into dummy variables based on number of categories present in each variable.

2.7 Principal component analysis (PCA)

PCA is a procedure for reducing the dimensionality of the variable space by representing it with a few orthogonal (uncorrelated) variables that capture most of its variability.

Here we have data with 115 variables after dummy variable creation, after the Principal component analysis, we found that 40 principal components contribute 97+ variance out of 115. Hence, we choose only 40 variables as input to the models



Chapter 3

Model development

3.1 Model selection

The data underwent exploratory analysis, pre-processing analysis for model development, our target variable is the continuous variable. Hence, we have to go for regression related models, to check the performance of the model, we used error metrics, based error metrics we will decide the best predictive model for Absenteeism time in hours prediction.

3.2 Decision tree-regression model

The Decision tree model is predictive model, it belongs to supervised learning model, this model used for both classification and regression problems. Here we are using this model for regression purpose; regression trees are used to predict the target variable. It breaks down the dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

The 97+ of data variance is explained by 40 principle components, this is data sampled into training and test, Decision tree model builds with train data and tested with new test cases.

Decision tree model (tools)	Root Mean Squared Error (RMSE)	coefficient of determination (R2)
R software	0.4350969	0.9861675
Python software	0.5456006483456934	0.9738544976012422

3.3 Random forest model

This model is one of the supervised learning models. The random forest is an ensemble that consists of many decision trees, this method combines Breiman's "bagging idea" and the random selection of features. This method can be used for both classification and regression.

In this analysis we have used the random forest model for regression purpose, the number of trees used for analysis is 500 in both R and Python. The 40 principle components-based train and test samples are used for model development.

Random forest (tools)	Root Mean Squared Error (RMSE)	coefficient of determination (R2)
R software	0.7315962	0.9756930
Python software	0.045233485755368785	0.9998202920257213

3.4 Linear regression

The linear regression is one of the prediction models, it classified into simple linear regression and multiple linear regression model, it describes the relationship among variables such as one dependent variable may relate to one independent variable or multiple independent variables.

The multiple linear regression model used in this analysis, the 40 principle components-based train and test samples are used for model development.

Linear regression (tools)	Root Mean Squared Error (RMSE)	coefficient of determination (R2)
R software	0.004611191	0.999998431
Python software	0.006291368885717461	0.9999965235376019

Chapter 4

Conclusion

4.1 Model evaluation

Three models are used for Absenteeism time in hours prediction such as Decision tree, Random forest model, Linear regression model. These models are evaluated by using error metrics such as Root mean squared error (RMSE) and Coefficient of determination (R^2).

RMSE – It is based the assumption that data error follows normal distribution. This is the measure of the average deviation of model predictions from the actual values in the dataset. It ranges from 0 to 1, model should have lower value for better performance.

R2 - It is the proportion of the variance of a dependent variable that's explained by an independent variable or variables in the regression model. Whereas co-relation explains the strength of the relationship between an independent variables and dependent variable. It ranges from 0 to 1, model should have a higher value for goodness of fit.

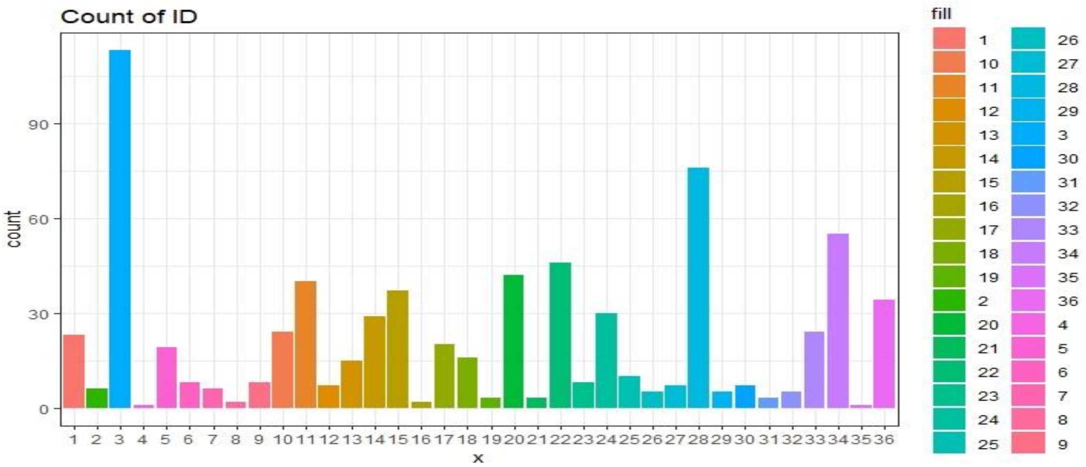
4.2 Model selection

Based on the error scores ***linear regression model*** is better than decision tree and random forest model. It has lower RMSE and higher R2 value.

4.3 Problem statement solutions:

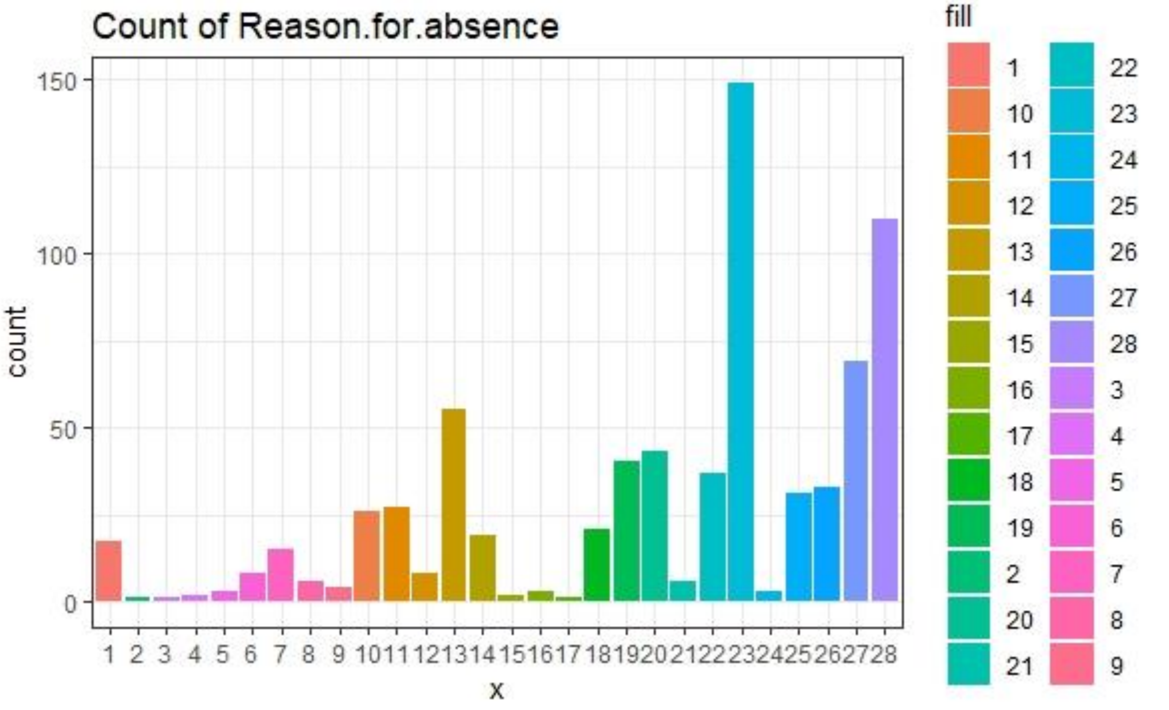
1. What changes the company should bring to reduce the number of absenteeism?

Solution-1



a) Based on the ID categorical variable ID numbers 3,28 and 34 has more counts, employer must ask these employees to plan their leaves in advance based on business priority.
b) Employer also ask for employees restrict their leaves to take monthly allowed limits
c) Employer must inform to employees if they continue this trend, the company will take necessary steps for best of the business needs.

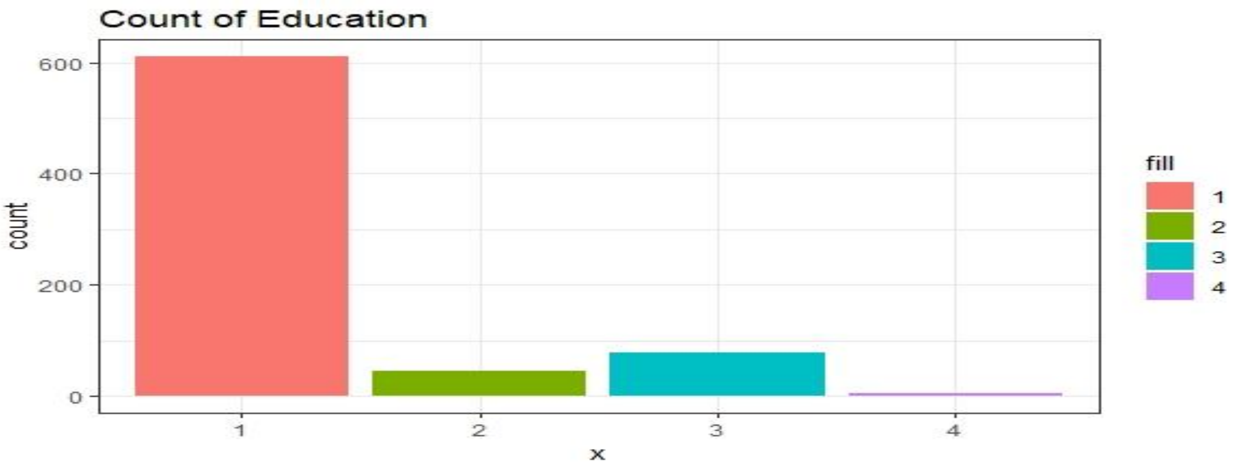
Solution -2



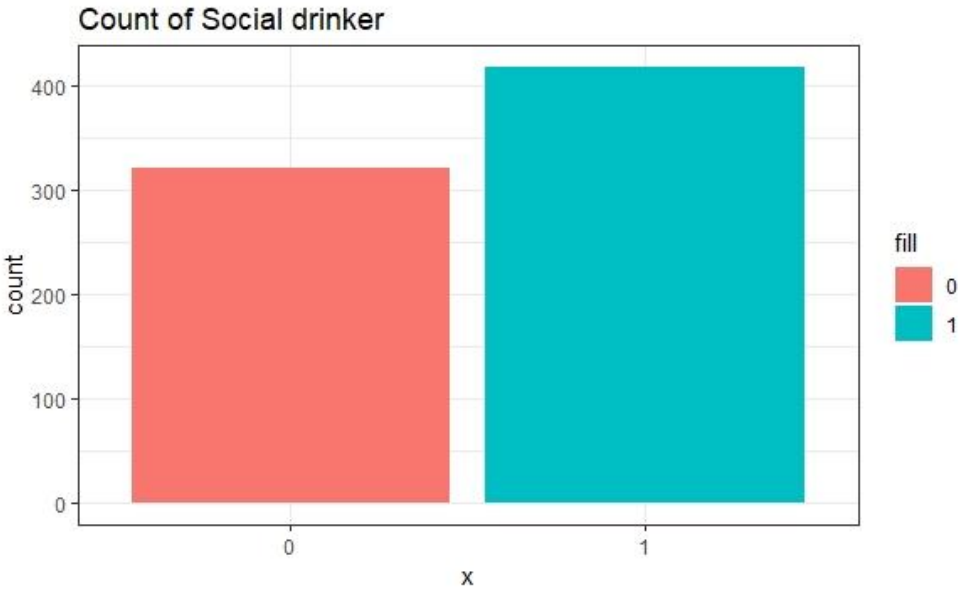
a) Based on Reason for absence category 23,28,27 and 13 accounts for most of absenteeism hours.
b) Employer HR team must take steps to educate disease awareness and prevention session
c) If they have insurance policies for employees, asks them to utilize for full recovery

Solution – 3

a) Education category shows employees' education level 1 contributes to more absenteeism in hours.
b) Employer must communicate to these employees their leaves affects the business needs and ask them to stick to company leave policy
c) Company must hire employees with minimum education quality at the graduate level



Solution – 4



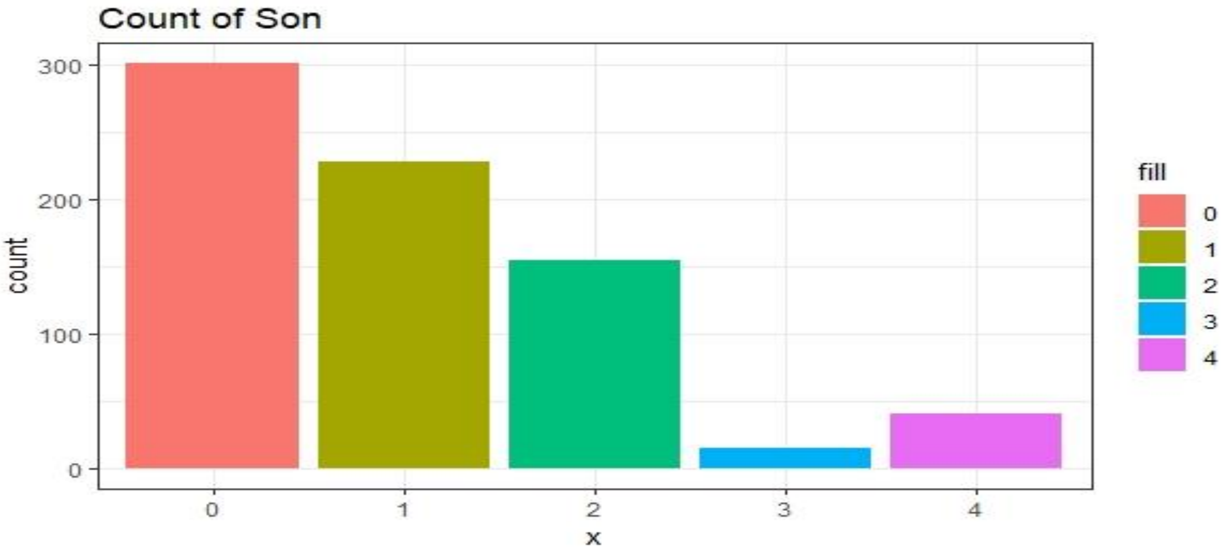
a) Social drinker percentage is higher among absentees, the company must take a session to employees related alcohol use disorder and its impact on personal health and family.

b) The company must avoid the use of alcoholic beverage at its management meetings and employee engagement meetings.

Solution – 5

a) Employees with a son of 0 and 1 are with more absenteeism hours, they may plan to witch a job.

b) The company need to introduce employee engagement programs like learning, carrier growth sessions for them increase their moral and commitment for the job.



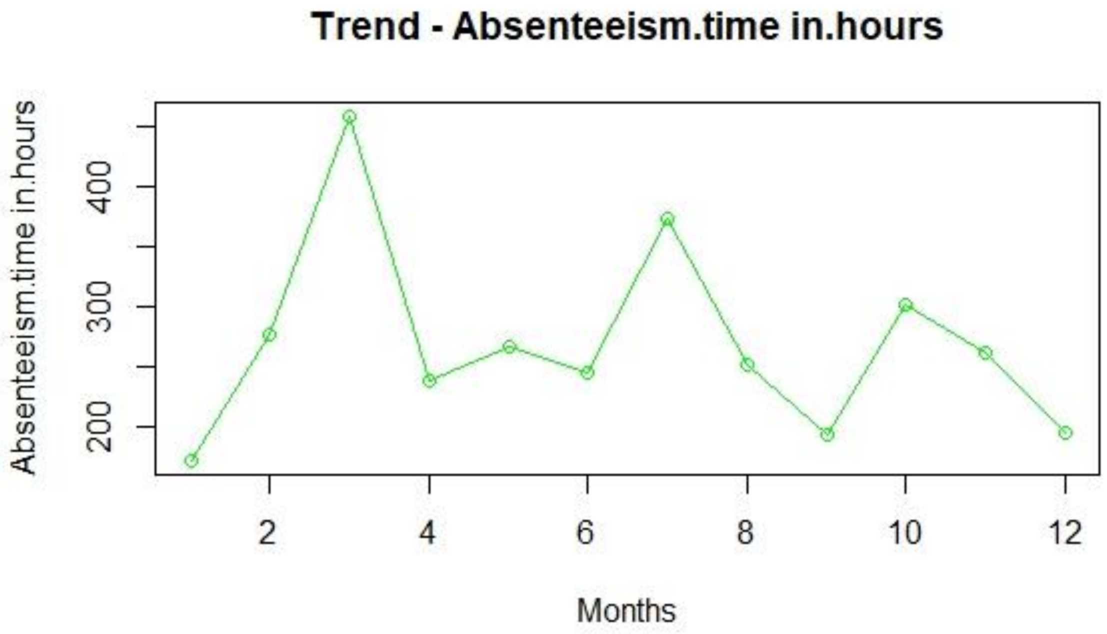
2. How much losses every month can we project in 2011, if a same trend of absenteeism continuous?

   After the model implementation, trend of absenteeism based on months is calculated. They are listed below: Python based model implemented data is used for this solution

Months	Absenteeism time in hours
Month of absence_1	171.8062485
Month of absence_2	277.1417883
Month of absence_3	458.9204435
Month of absence_4	238.0416918
Month of absence_5	265.9221843
Month of absence_6	245.3290264
Month of absence_7	374.479797
Month of absence_8	251.1669413
Month of absence_9	192.8479438
Month of absence_10	302.0889285
Month of absence_11	261.0868442
Month of absence_12	195.1109823

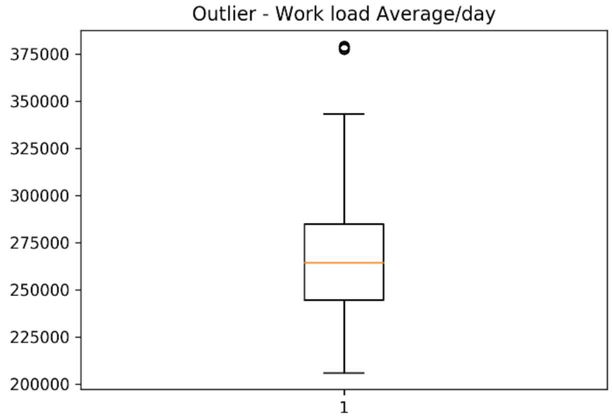
Based on the data March month has a greater number of absenteeism time in hours, if this trend continuous total Absenteeism time in hours would be 3233.94 hours.

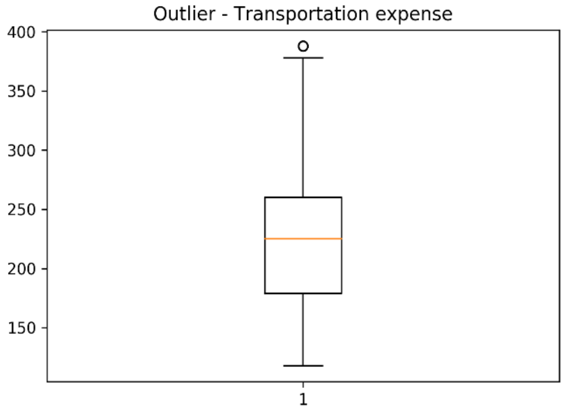
If I consider one employee works for 8 hours per day for 5 days in a week, it will come for 40 hours per week as working hours. For 52 weeks in a year in comes as 2080 hours. (3233/2080) = 1.55, which means a total time loss in a year equal to 1.5 employees working hours loss in a year.

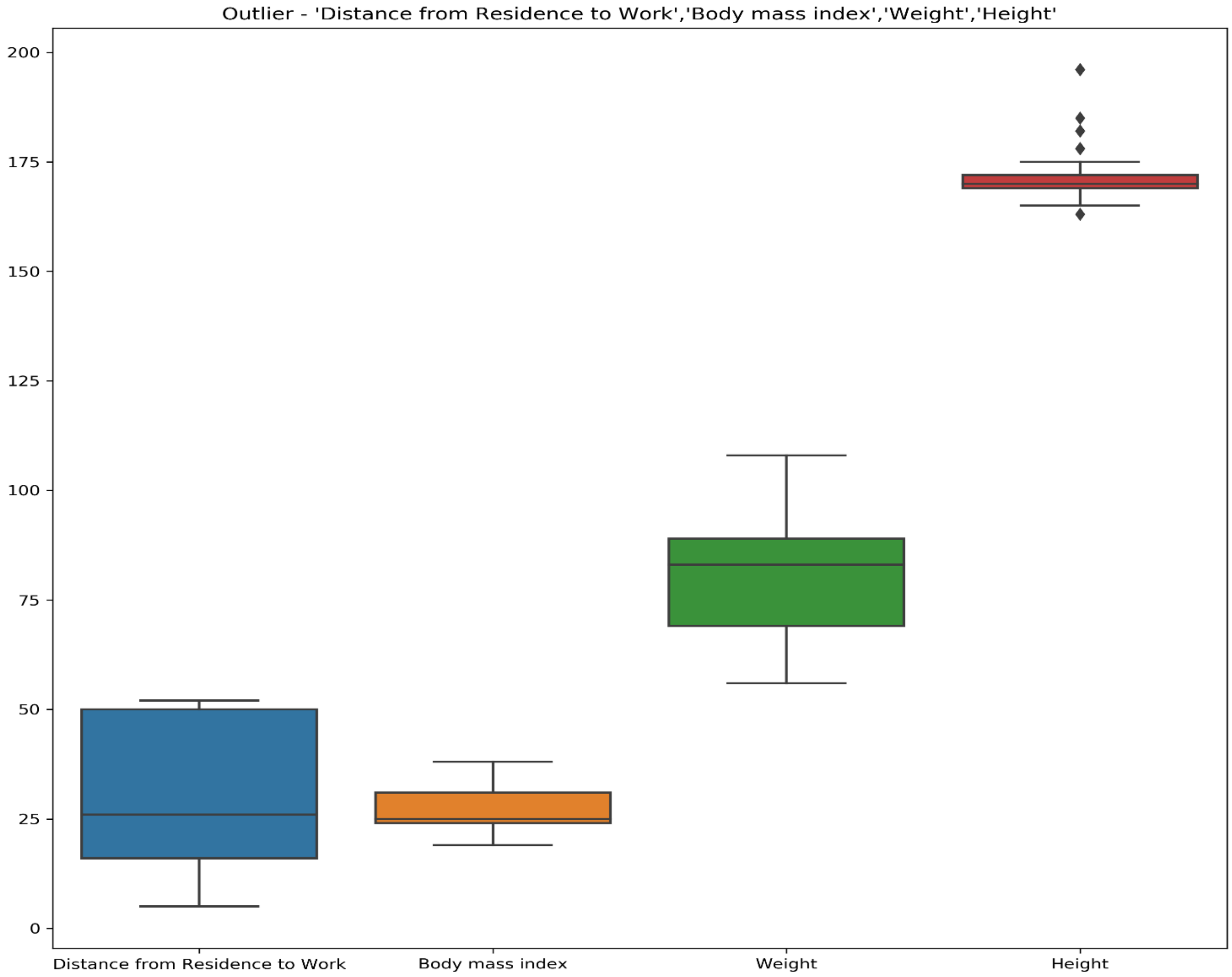

Trend - Absenteeism.time in.hours
Absenteeism.time in.hours
200
300
400
2
4
6
8
10
12
Months

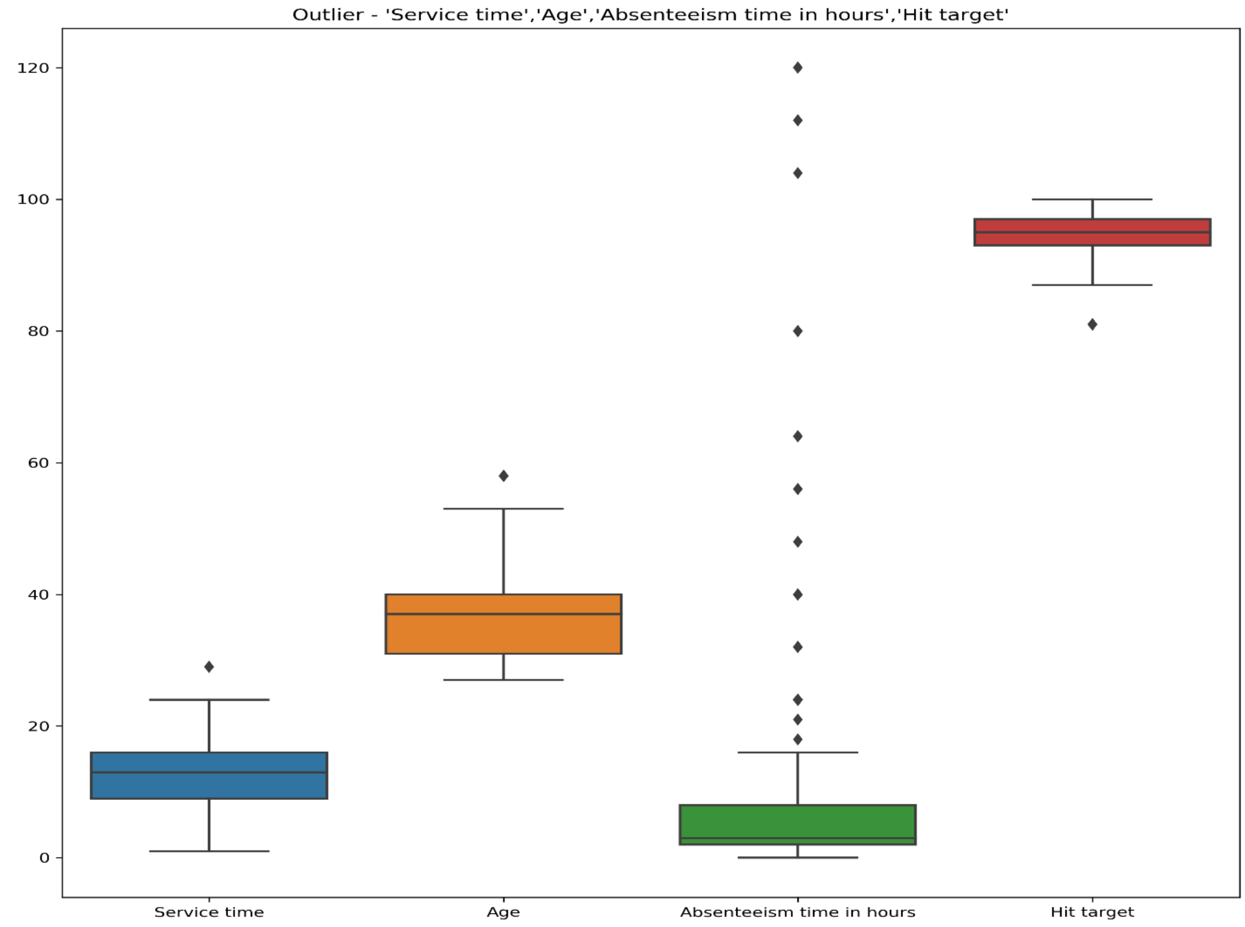
Appendix A - Python plots

Outlier plot

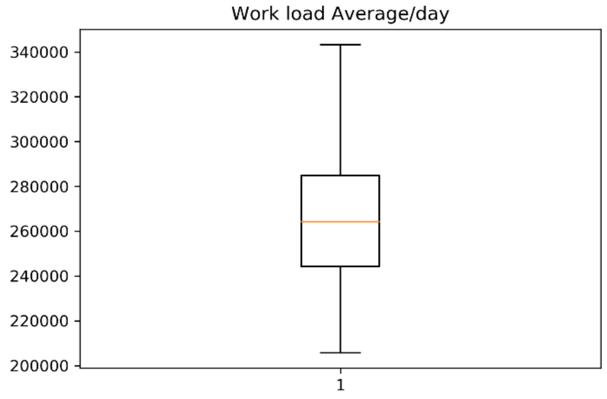

Outlier - Work load Average/day

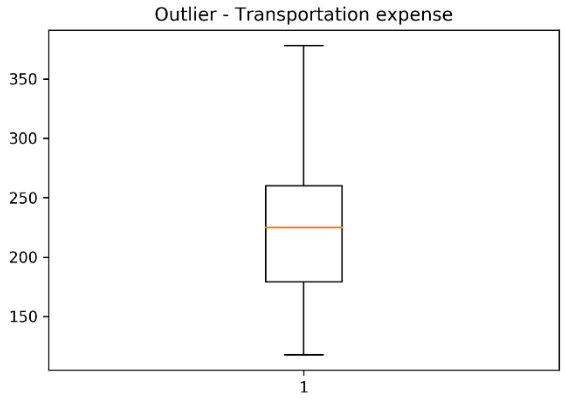

Outlier - Transportation expense

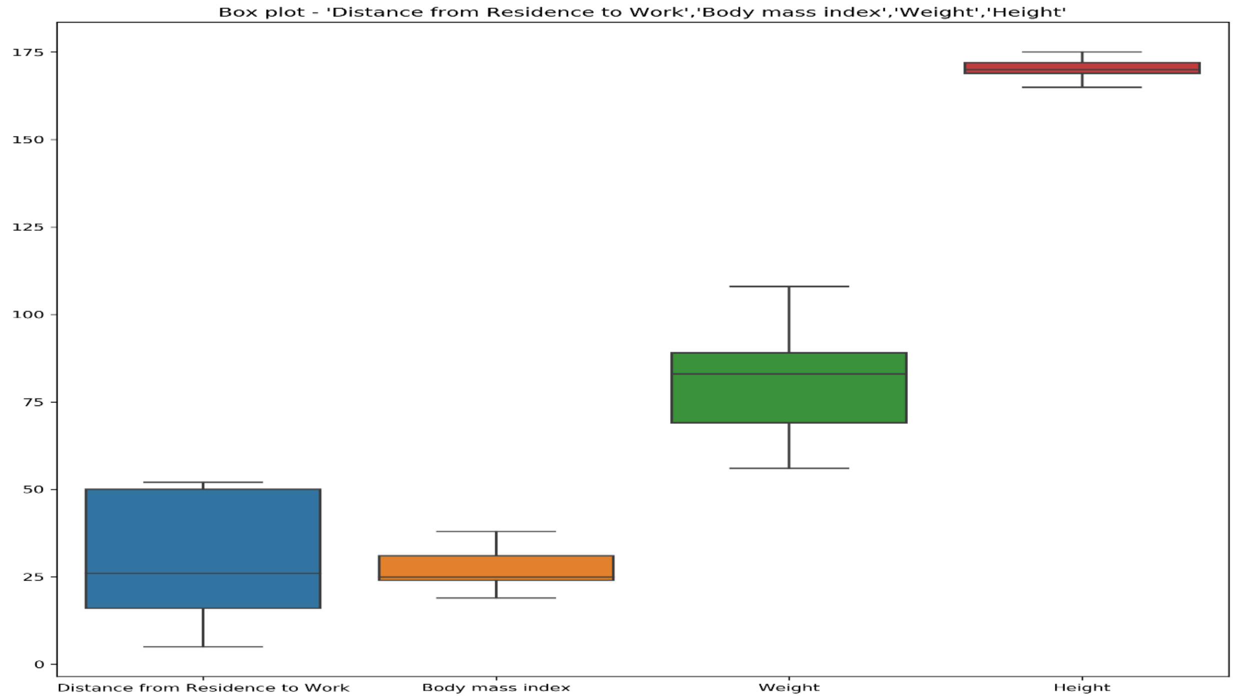

Outlier - 'Distance from Residence to Work','Body mass index','Weight','Height'

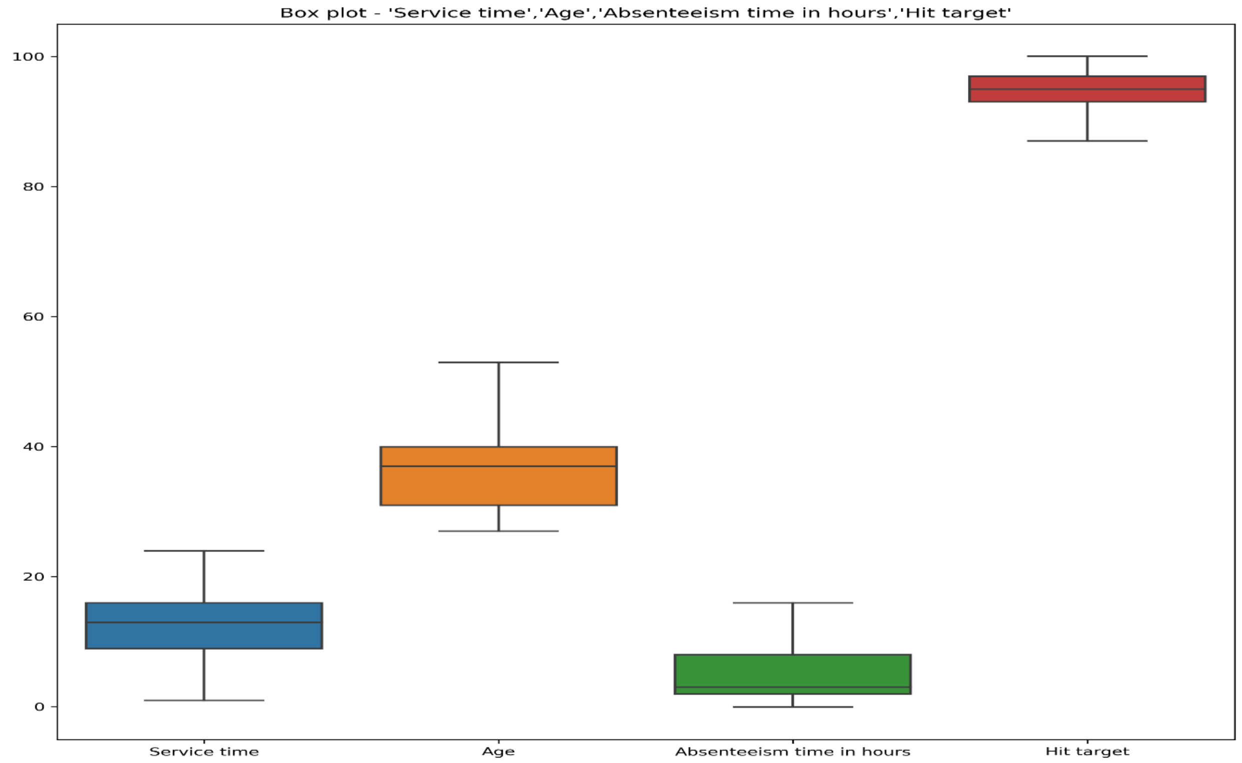


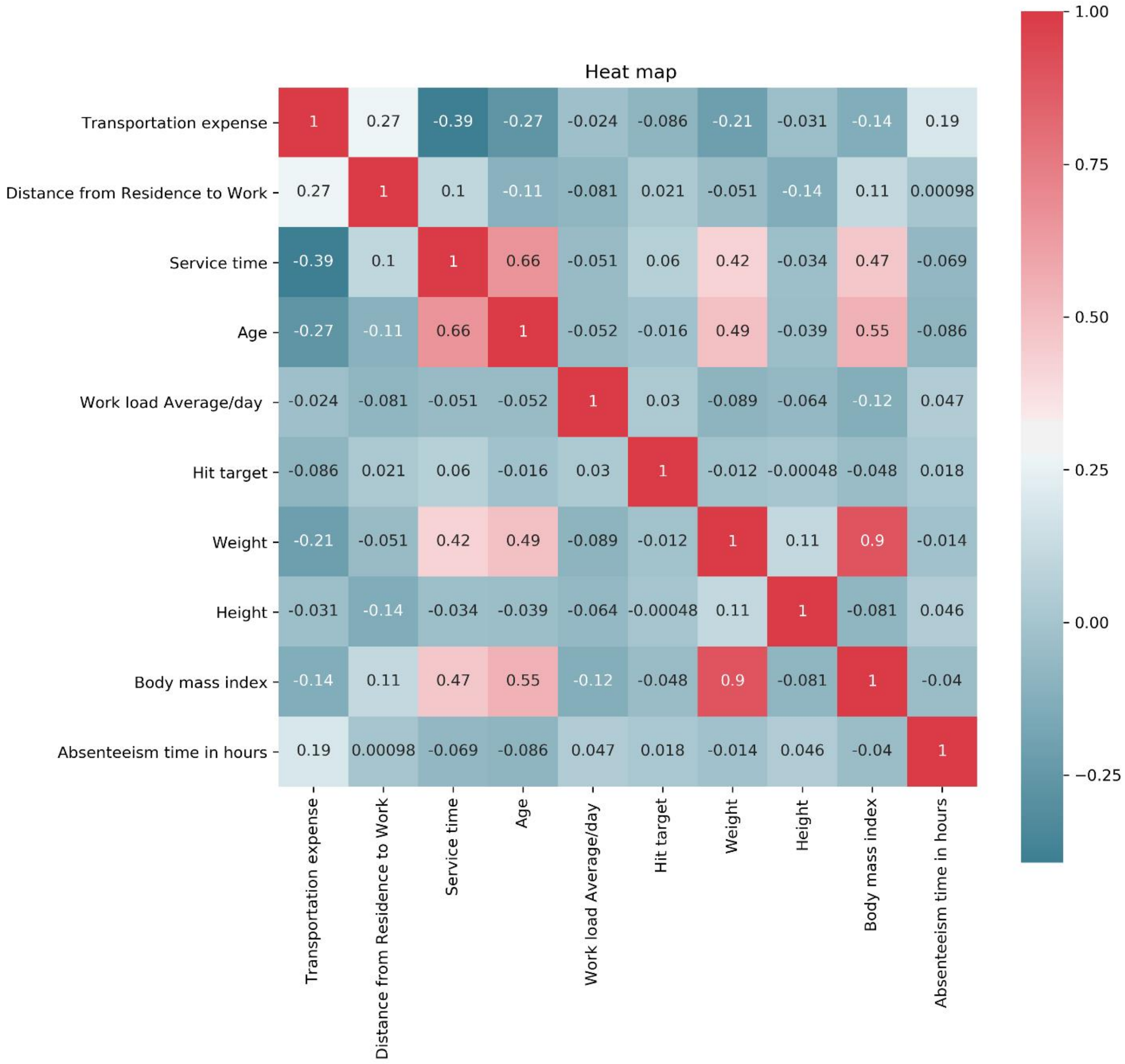
Box plot – without outliers





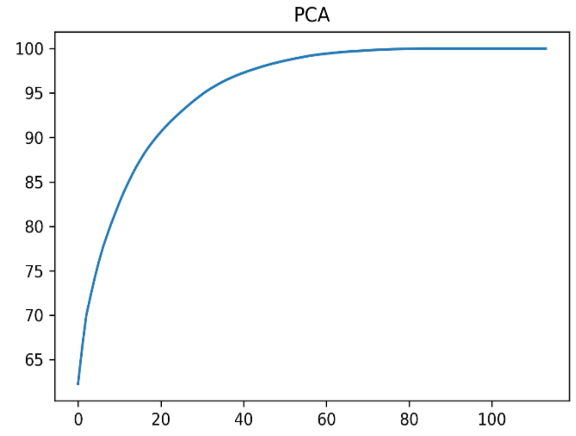

Box plot - 'Distance from Residence to Work','Body mass index','Weight','Height'
175
150
125
100
75
50
25
0
Distance from Residence to Work
Body mass index
Weight
Height


Box plot - 'Service time','Age','Absenteeism time in hours','Hit target'
100
80
60
40
20
0
Service time
Age
Absenteeism time in hours
Hit target

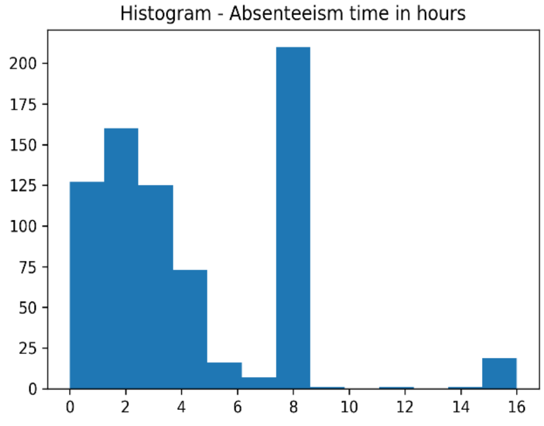


Heat map

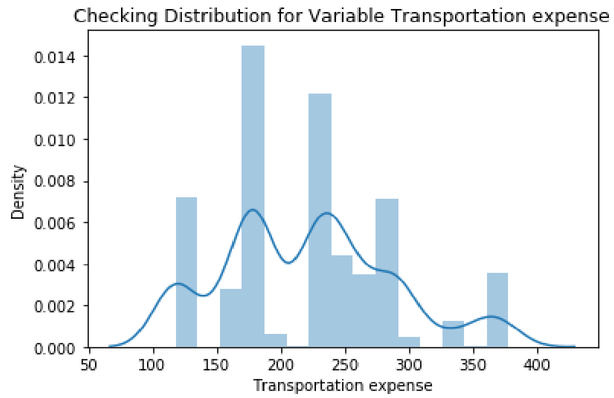
	Transportation expense	Distance from Residence to Work	Service time	Age	Work load Average/day	Hit target	Weight	Height	Body mass index	Absenteeism time in hours
Transportation expense	1	0.27	-0.39	-0.27	-0.024	-0.086	-0.21	-0.031	-0.14	0.19
Distance from Residence to Work	0.27	1	0.1	-0.11	-0.081	0.021	-0.051	-0.14	0.11	0.00098
Service time	-0.39	0.1	1	0.66	-0.051	0.06	0.42	-0.034	0.47	-0.069
Age	-0.27	-0.11	0.66	1	-0.052	-0.016	0.49	-0.039	0.55	-0.086
Work load Average/day	-0.024	-0.081	-0.051	-0.052	1	0.03	-0.089	-0.064	-0.12	0.047
Hit target	-0.086	0.021	0.06	-0.016	0.03	1	-0.012	-0.00048	-0.048	0.018
Weight	-0.21	-0.051	0.42	0.49	-0.089	-0.012	1	0.11	0.9	-0.014
Height	-0.031	-0.14	-0.034	-0.039	-0.064	-0.00048	0.11	1	-0.081	0.046
Body mass index	-0.14	0.11	0.47	0.55	-0.12	-0.048	0.9	-0.081	1	-0.04
Absenteeism time in hours	0.19	0.00098	-0.069	-0.086	0.047	0.018	-0.014	0.046	-0.04	1



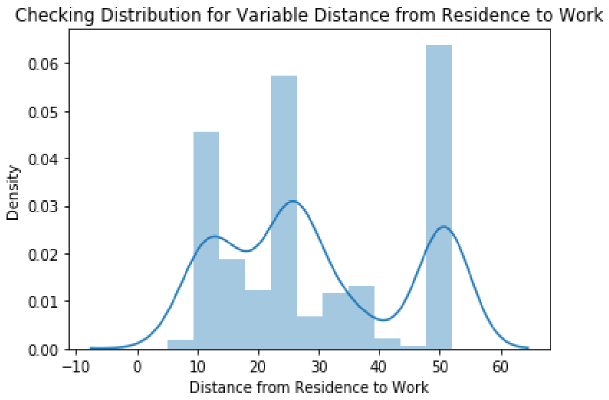
PCA



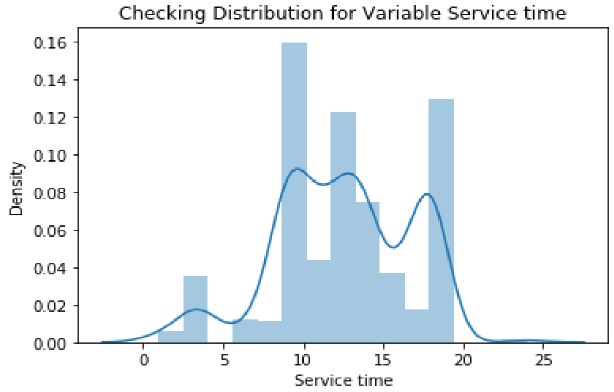
Histogram - Absenteeism time in hours



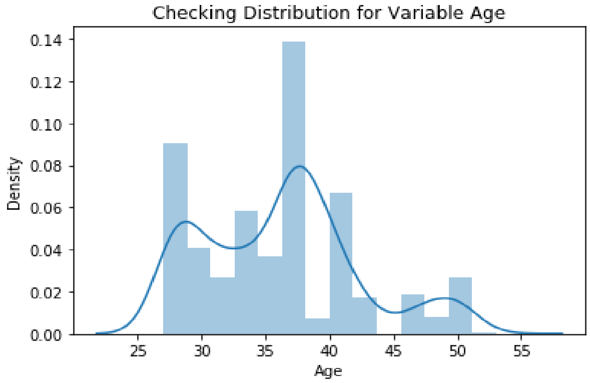
Checking Distribution for Variable Transportation expense



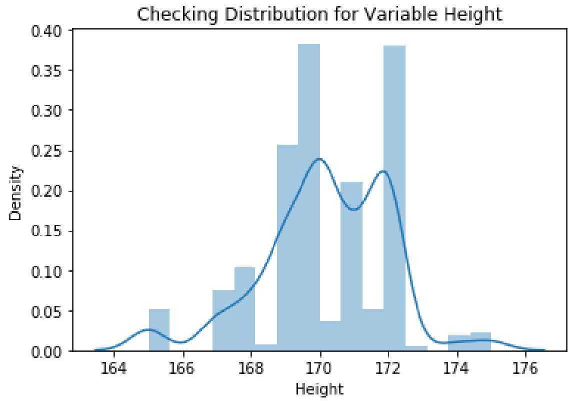
Checking Distribution for Variable Distance from Residence to Work



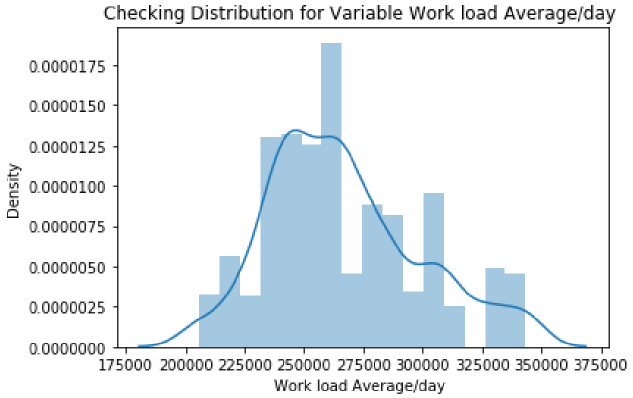
Checking Distribution for Variable Service time



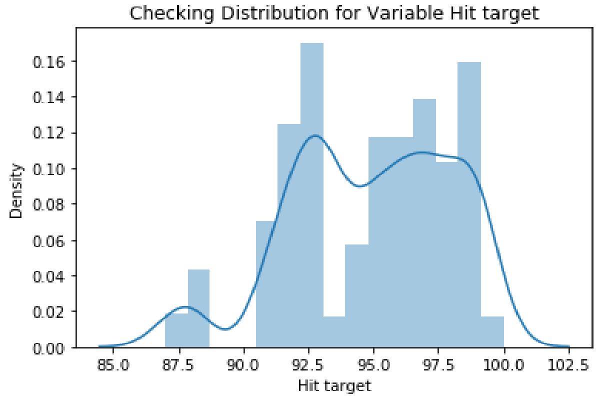
Checking Distribution for Variable Age



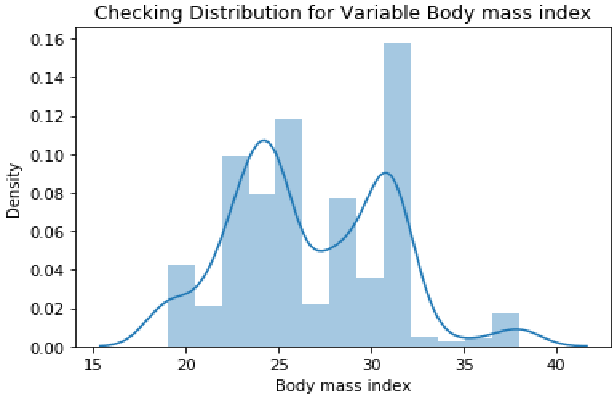
Checking Distribution for Variable Height



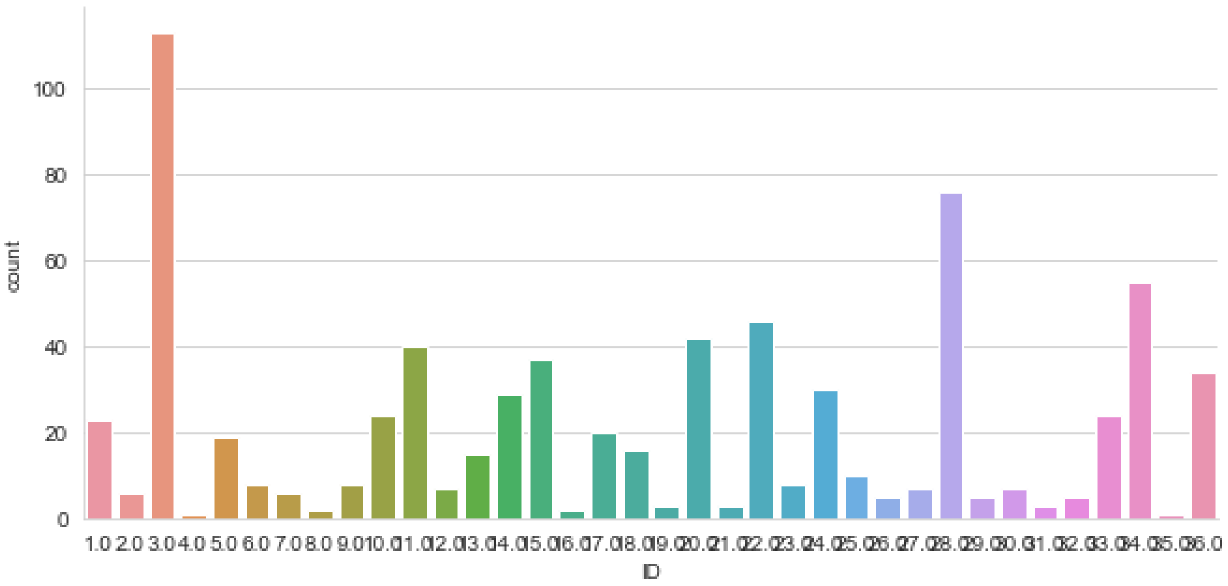
Checking Distribution for Variable Work load Average/day

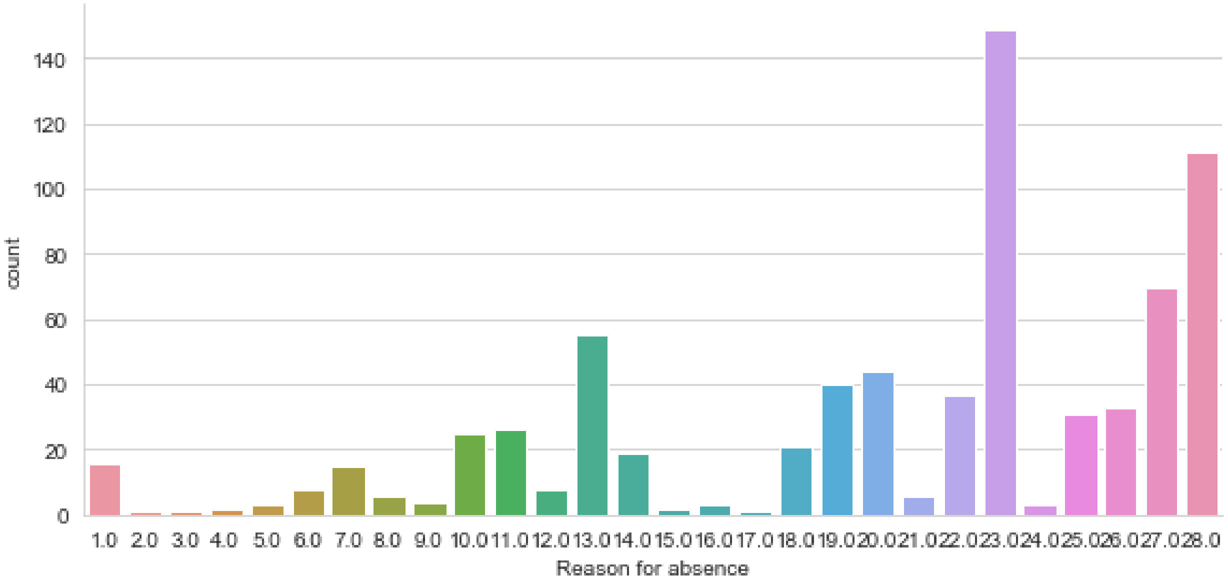


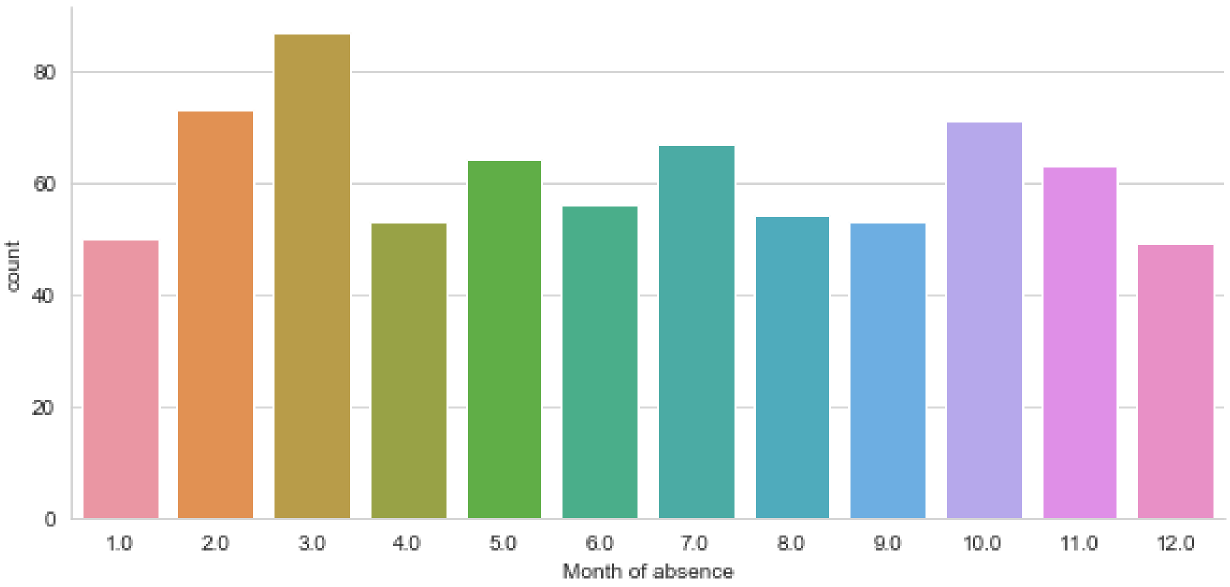
Checking Distribution for Variable Hit target

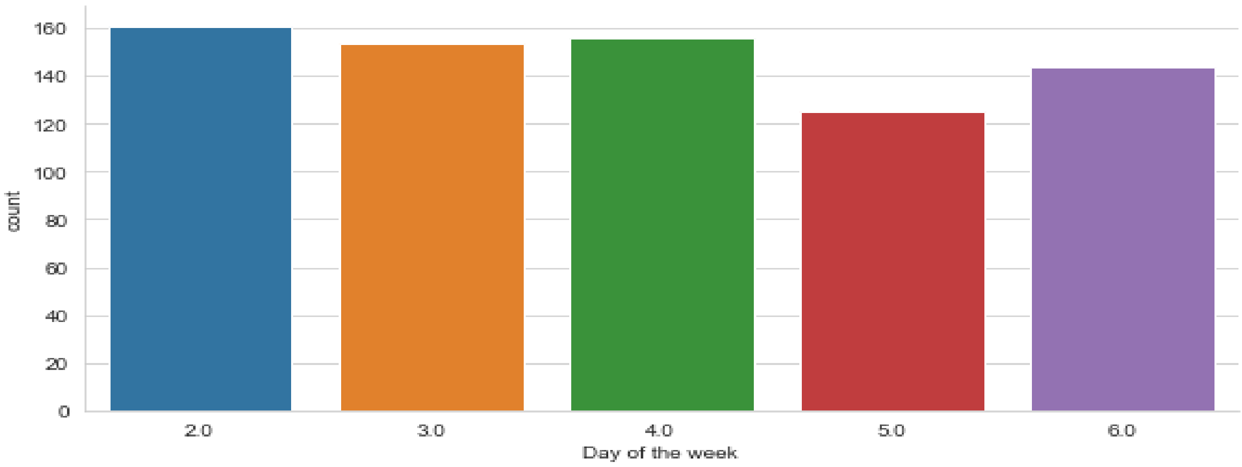


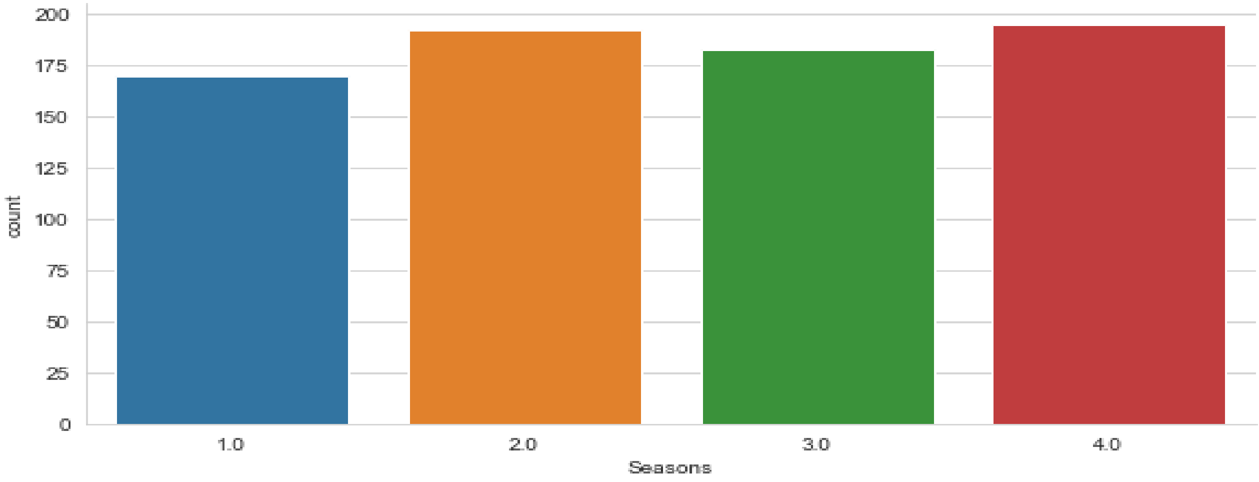
Checking Distribution for Variable Body mass index

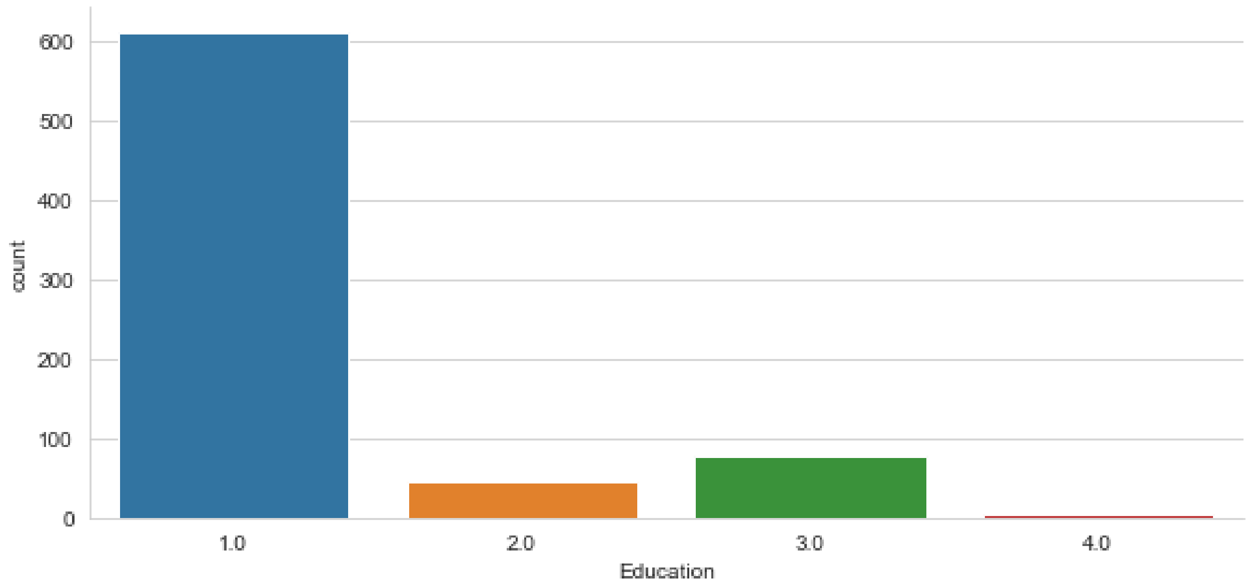

count
0
20
40
60
80
100
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0 30.0 31.0 32.0 33.0 34.0 35.0 36.0
ID

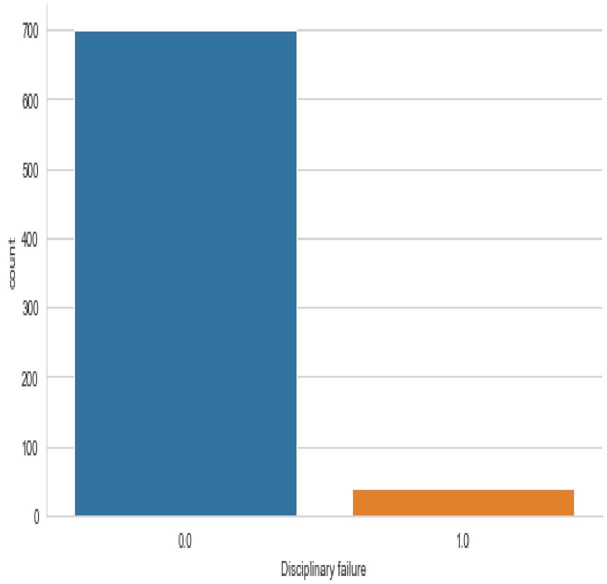

count
0
20
40
60
80
100
120
140
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0
Reason for absence

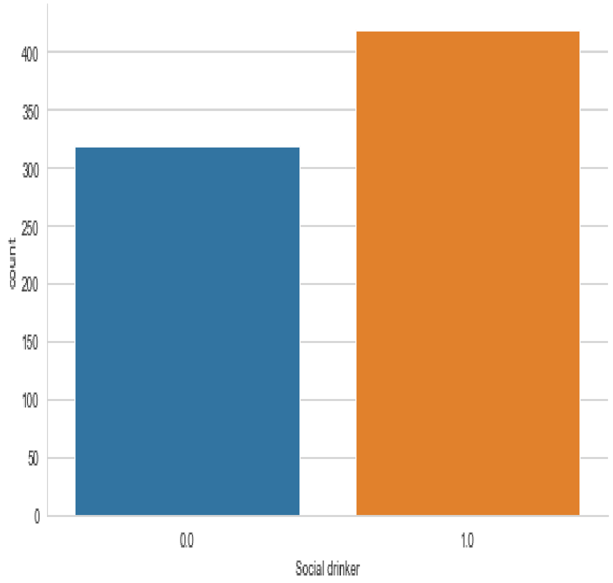

count
0
20
40
60
80
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
Month of absence

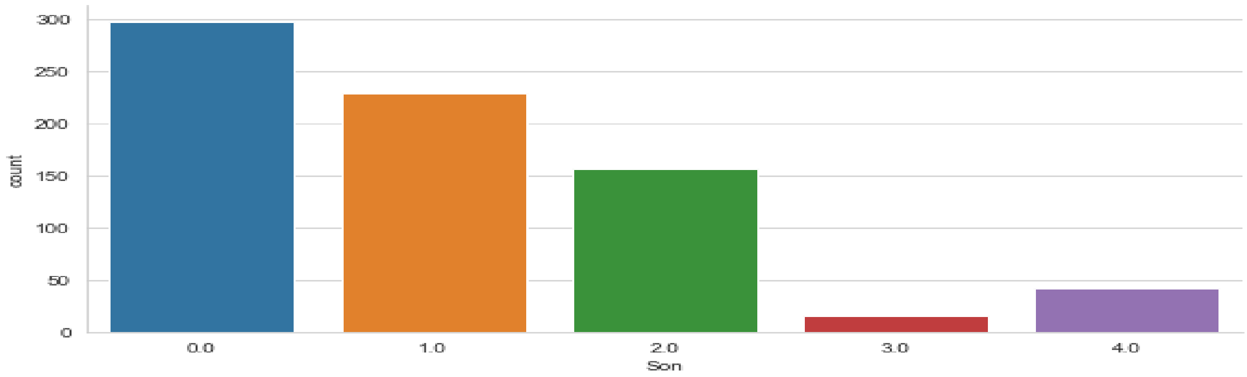

count
0
20
40
60
80
100
120
140
160
2.0
3.0
4.0
5.0
6.0
Day of the week

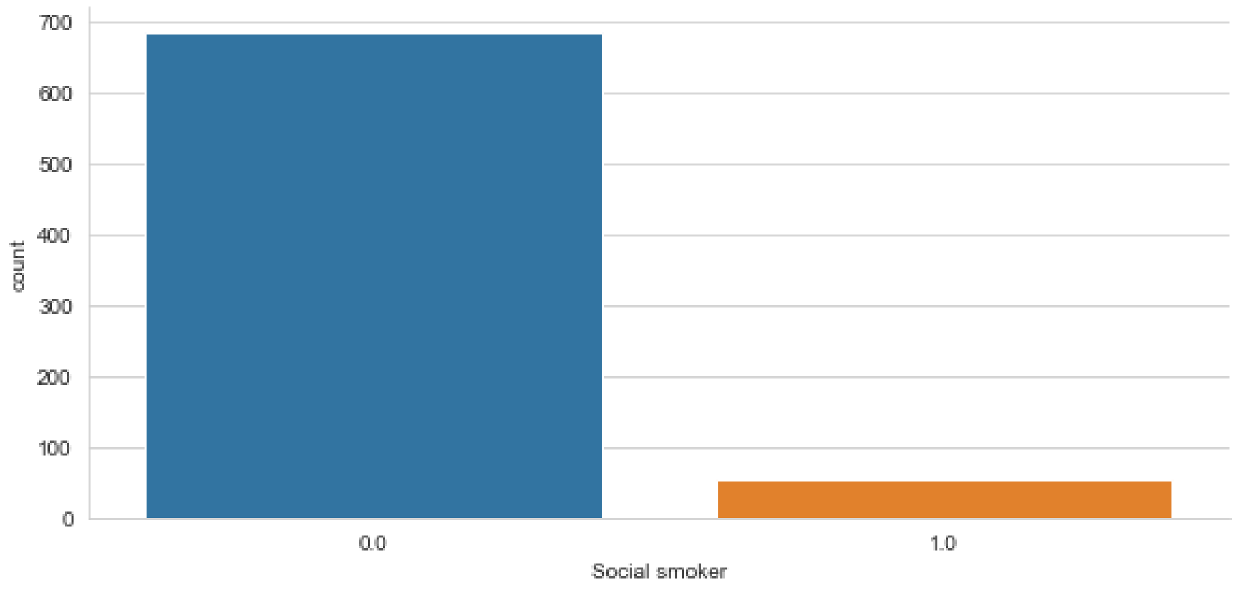

count
0
25
50
75
100
125
150
175
200
1.0
2.0
3.0
4.0
Seasons

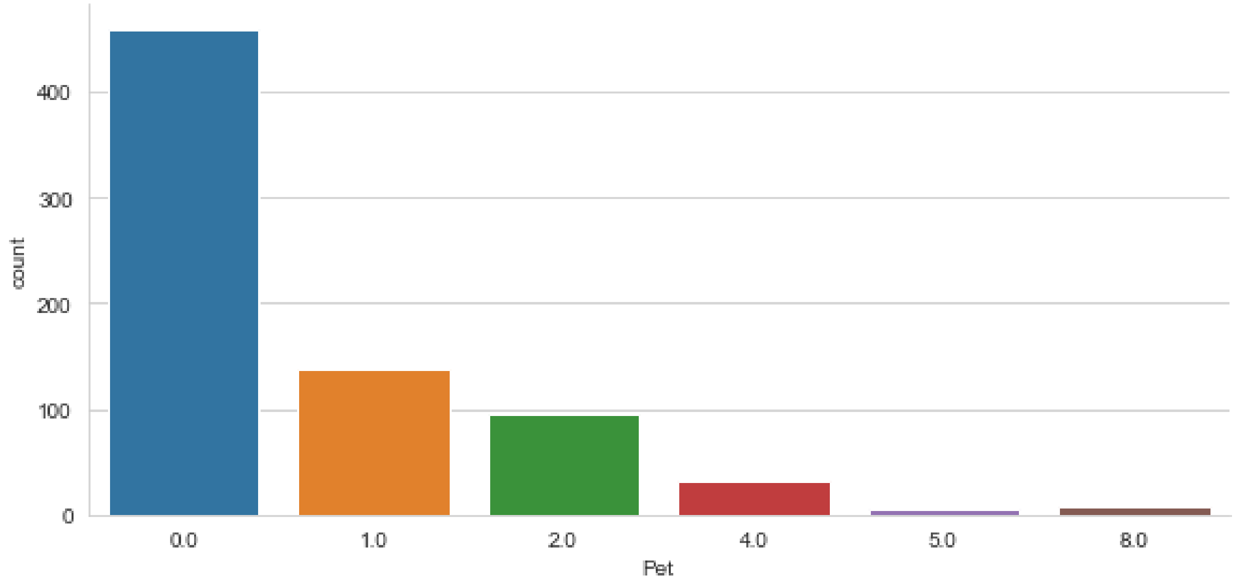

count
600
500
400
300
200
100
0
1.0
2.0
3.0
4.0
Education


count
700
600
500
400
300
200
100
0
0.0
1.0
Disciplinary failure

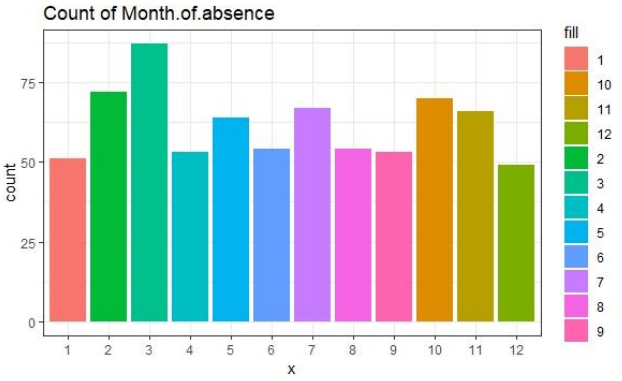

count
400
350
300
250
200
150
100
50
0
0.0
1.0
Social drinker

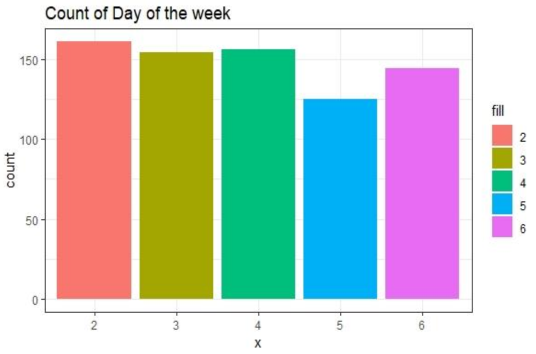

count
300
250
200
150
100
50
0
0.0
1.0
2.0
3.0
4.0
Son

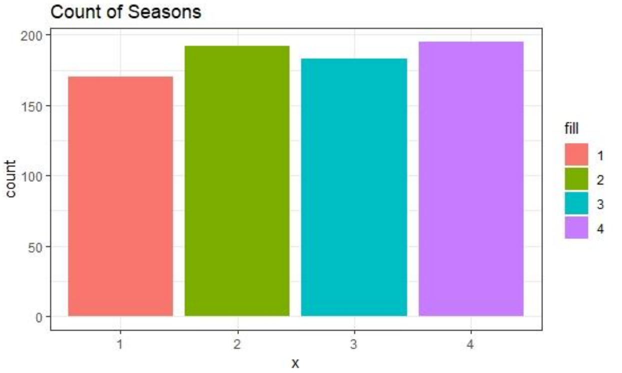

700
600
500
400
300
200
100
0
count
0.0
1.0
Social smoker

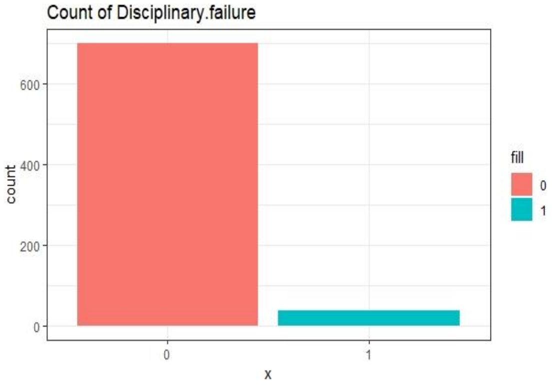

400
300
200
100
0
count
0.0
1.0
2.0
4.0
5.0
8.0
Pet

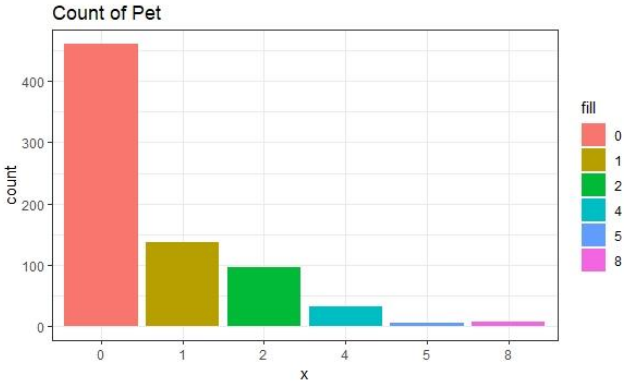
Appendix B – R plots

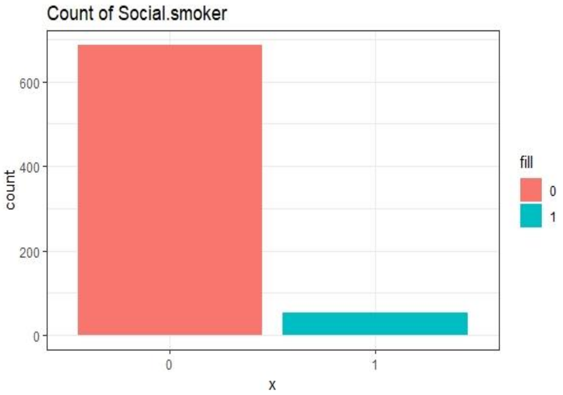












R – codes

```
# Setting working directory
setwd("E:/edWisor/Project")


# Loading libraries

x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced","dummies", "e1071",
"Information",

      "MASS", "rpart","gbm", "ROSE","sampling", "DataCombine", "xlsx","inTrees","usdm")

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


rm(x)



## Read the data
emp_abs = read.xlsx("Absenteeism_at_work_Project.xls",sheetIndex=1, header = T)


#####################################Explore the data  ##################################
dim(emp_abs)
names(emp_abs)
head(emp_abs,5)
str(emp_abs)
summary(emp_abs)
# Numeric to factor transformation

emp_abs$ID = as.factor(as.character(emp_abs$ID))
emp_abs$Reason.for.absence[emp_abs$Reason.for.absence %in% 0]=20
emp_abs$Reason.for.absence = as.factor(as.character(emp_abs$Reason.for.absence))
emp_abs$Month.of.absence[emp_abs$Month.of.absence %in% 0]= NA
emp_abs$Month.of.absence = as.factor(as.character(emp_abs$Month.of.absence))
emp_abs$Day.of.the.week= as.factor(as.character(emp_abs$Day.of.the.week))
emp_abs$Seasons = as.factor(as.character(emp_abs$Seasons))
emp_abs$Disciplinary.failure = as.factor(as.character(emp_abs$Disciplinary.failure))
emp_abs$Education = as.factor(as.character(emp_abs$Education))
emp_abs$Son = as.factor(as.character(emp_abs$Son))
emp_abs$Social.drinker = as.factor(as.character(emp_abs$Social.drinker))
emp_abs$Social.smoker = as.factor(as.character(emp_abs$Social.smoker))

emp_abs$Pet = as.factor(as.character(emp_abs$Pet))

str(emp_abs)
```

```
############################# Missing value analysis ###########################
missing_val = data.frame(apply(emp_abs,2,function(x){sum(is.na(x))}))

# Converting rownames into column names
missing_val$Columns = row.names(missing_val)
row.names(missing_val) = NULL
# Rename the variable
names(missing_val)[1]= "Missing_percentage"
# Calculating percentage
missing_val$Missing_percentage = (missing_val$Missing_percentage/nrow(emp_abs))*100
# Arrange in descending order
missing_val = missing_val[order(-missing_val$Missing_percentage),]
row.names(missing_val) = NULL
#rearrange the column
missing_val= missing_val[,c(2,1)]

# Writing output result back to disk
write.csv(missing_val,"emp_abs_mp.csv",row.names = F)


# Visualization of missing value
ggplot(data = missing_val[1:6,], aes(x=reorder(Columns, -Missing_percentage),y = 
Missing_percentage))+
  geom_bar(stat = "identity",fill = "greenyellow")+xlab("Parameter")+
  ggtitle("Missing data percentage (Train)") + theme_bw()

# Identifying imputation method (mean, median, KNN)
emp_abs[["Body.mass.index"]][5]= NA

# Actual value = 30
# Mean imputed value = 26.67938
# Median imputed value = 25
# KNN imputed value = 30
# Mean method
#emp_abs$Body.mass.index[is.na(emp_abs$Body.mass.index)] = mean(emp_abs$Body.mass.index, 
na.rm = T)

# Median method
#emp_abs$Body.mass.index[is.na(emp_abs$Body.mass.index)] = median(emp_abs$Body.mass.index, 
na.rm = T)

# KNN imputation
emp_abs = knnImputation(emp_abs, k=3)
sum(is.na(emp_abs))


write.csv(emp_abs, 'emp_abs_missing.csv', row.names = F)
##################################### Graphs ###################################
```

```
b1<-ggplot(emp_abs, aes_string(x=reorder(emp_abs$ID),fill =emp_abs$ID)) + geom_bar(stat ='count')+
ggtitle("Count of ID") + theme_bw()
b2<-ggplot(emp_abs,
aes_string(reorder(emp_abs$Reason.for.absence),fill=emp_abs$Reason.for.absence)) + geom_bar(stat
= "count")+ ggtitle("Count of Reason.for.absence") + theme_bw()
b3<-ggplot(emp_abs,
aes_string(reorder(emp_abs$Month.of.absence),fill=emp_abs$Month.of.absence)) + geom_bar(stat =
"count")+ ggtitle("Count of Month.of.absence") + theme_bw()
b4<-ggplot(emp_abs, aes_string(reorder(emp_abs$Day.of.the.week),fill=emp_abs$Day.of.the.week)) +
geom_bar(stat = "count")+ ggtitle("Count of Day of the week") + theme_bw()
b5<-ggplot(emp_abs, aes_string(reorder(emp_abs$Seasons),fill=emp_abs$Seasons)) + geom_bar(stat =
"count")+ ggtitle("Count of Seasons") + theme_bw()
b6<-ggplot(emp_abs,
aes_string(reorder(emp_abs$Disciplinary.failure),fill=emp_abs$Disciplinary.failure)) + geom_bar(stat =
"count")+ ggtitle("Count of Disciplinary.failure") + theme_bw()
b7<-ggplot(emp_abs, aes_string(reorder(emp_abs$Education),fill=emp_abs$Education)) +
geom_bar(stat = "count")+ ggtitle("Count of Education") + theme_bw()
b8<-ggplot(emp_abs, aes_string(reorder(emp_abs$Son),fill=emp_abs$Son)) + geom_bar(stat =
"count")+ ggtitle("Count of Son") + theme_bw()
b9<-ggplot(emp_abs, aes_string(reorder(emp_abs$Social.drinker),fill=emp_abs$Social.drinker)) +
geom_bar(stat = "count")+ ggtitle("Count of Social drinker") + theme_bw()
b10<-ggplot(emp_abs, aes_string(reorder(emp_abs$Social.smoker),fill=emp_abs$Social.smoker)) +
geom_bar(stat = "count")+ ggtitle("Count of Social.smoker") + theme_bw()
b11<-ggplot(emp_abs, aes_string(reorder(emp_abs$Pet),fill=emp_abs$Pet)) + geom_bar(stat =
"count")+ ggtitle("Count of Pet") + theme_bw()
b1
b2
gridExtra::grid.arrange(b3,b4,ncol=2)
gridExtra::grid.arrange(b5,b6,ncol=2)
gridExtra::grid.arrange(b7,b8,ncol=2)
gridExtra::grid.arrange(b9,b10,b11,ncol=3)

################################# Outlier analysis ###################################
# ## Box Plots - Distribution and Outlier Check
numeric_index = sapply(emp_abs,is.numeric) # Selecting only numeric

numeric_data = emp_abs[,numeric_index]
cnames = colnames(numeric_data)

# Selecting categorical data
categorical_data = emp_abs[,!numeric_index]



#
```

```
for(i in 1:ncol(numeric_data)) {
  assign(paste0("gn",i),ggplot(data = emp_abs, 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(gn1,gn2,gn3,gn4,ncol=2)
gridExtra::grid.arrange(gn5,gn6,gn7,gn8,ncol=2)
gridExtra::grid.arrange(gn9,gn10,ncol=2)


# # Replace all outliers with NA and impute
for(i in cnames){
  val = emp_abs[,i][emp_abs[,i] %in% boxplot.stats(emp_abs[,i])$out]
  print(paste(i,length(val)))
  emp_abs[,i][emp_abs[,i] %in% val] = NA
}

# Outlier-missing value calculation
# Get number of missing values after replacing outliers as NA
outlier_missing_values = data.frame(sapply(emp_abs,function(x){sum(is.na(x))}))
outlier_missing_values$Columns = row.names(outlier_missing_values)
row.names(outlier_missing_values) = NULL
names(outlier_missing_values)[1] = "Out_Miss_perc"
outlier_missing_values$Out_Miss_perc =
((outlier_missing_values$Out_Miss_perc/nrow(emp_abs))*100)
outlier_missing_values = outlier_missing_values[,c(2,1)]
outlier_missing_values = outlier_missing_values[order(-outlier_missing_values$Out_Miss_perc),]

outlier_missing_values

# Visualization of missing value
ggplot(data = outlier_missing_values[1:6,], aes(x=reorder(Columns, -Out_Miss_perc),y =
Out_Miss_perc))+
  geom_bar(stat = "identity",fill = "green")+xlab("Parameter")+

  ggtitle("Outlier Missing data percentage (Train)") + theme_bw()

# KNN imputation
emp_abs = knnImputation(emp_abs, k=3)
sum(is.na(emp_abs))

# Data copy
df = emp_abs
# emps_abs = df
```

```
################################### Feature Selection ############################

## Correlation Plot
# corrgram library helps to plot correlation plot
corrgram(emp_abs[,numeric_index], order = F,
        upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

#Check for multicollinearity using VIF
vifcor(numeric_data)

## Dimension Reduction
emp_abs = subset.data.frame(emp_abs,select = -c(Weight))

numeric_data = subset.data.frame(numeric_data,select = -c(Weight))
################################## Feature scaling ###############################

# Norma Q-Q plot
qqnorm(emp_abs$Transportation.expense)
# Normality checks with histogram
h1<-ggplot(emp_abs, aes_string(emp_abs$Transportation.expense))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Transportation.expense")
h2<-ggplot(emp_abs,aes_string(emp_abs$Distance.from.Residence.to.Work))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Distance.from.Residence")
h3<-ggplot(emp_abs,aes_string(emp_abs$Service.time))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Service.time")
h4<-ggplot(emp_abs,aes_string(emp_abs$Age))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Age")
h5<-ggplot(emp_abs,aes_string(emp_abs$Work.load.Average.day.))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Work.load.Average.day.")
h6<-ggplot(emp_abs,aes_string(emp_abs$Hit.target))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Hit.target")
h7<-ggplot(emp_abs,aes_string(emp_abs$Height))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Height")
h8<-ggplot(emp_abs,aes_string(emp_abs$Body.mass.index))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Body.mass.index")
h9<-ggplot(emp_abs,aes_string(emp_abs$Absenteeism.time.in.hours))+
  geom_histogram(fill="DarkSlateBlue",colour ='black',bins = 30)+ggtitle("Absenteeism.time.in.hours")


gridExtra::grid.arrange(h1,h2,h3,ncol=1)
gridExtra::grid.arrange(h4,h5,h6,ncol=1)
gridExtra::grid.arrange(h7,h8,h9,ncol=1)


# Remove dependent variable
numeric_index = sapply(emp_abs,is.numeric) # Selecting only numeric
numeric_data = emp_abs[,numeric_index]
numeric_col = names(numeric_data)
numeric_col = numeric_col[-9]
```

```
for (i in numeric_col) {
  print(i)
  emp_abs[,i] = (emp_abs[,i] - min(emp_abs[,i]))/
    (max(emp_abs[,i]-min(emp_abs[,i])))

}
#################################### Sampling #######################################
# Creating dummy variables for categorical variables
categorical_names = names(categorical_data)
emp_abs = dummy.data.frame(emp_abs,categorical_names)

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

#Principal component analysis
p_c = prcomp(train)
#Standard deviation for principal component
pr_sd = p_c$sdev
#Variance calculation
pr_v = pr_sd^2
#Proportion of variance explained
pr_variance = pr_v/sum(pr_v)



# 97+ % data variance explained by 40 components
plot(cumsum(pr_variance), xlab = "Principal components",ylab = "cumulative proportion of variance",
     type = "b",main = "PCA")

#Adding a training set with principal components
train_1 = data.frame(Absenteeism.time.in.hours = train$Absenteeism.time.in.hours,p_c$x)
#Transforming data
train_1 = train_1[,1:40]


test_1 = predict(p_c,newdata = test)
test_1 = as.data.frame(test_1)
test_1 = test_1[,1:40]
# ######################################## Decision tree model ##############################
#rpart for regression(method = 'anova' for regression model, method = 'class' for classification model)
fit = rpart(Absenteeism.time.in.hours~., data = train_1, method = "anova")
#
# Predict for new test cases
predictions = predict(fit,test_1)
```

```
#
dt_pred = data.frame("actual"= test[,115],"DT_model" = predictions)
head(dt_pred)

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

################################ Random Forest model ##################################
###Random Forest(importance = TRUE , it shows important trees in the model)
RF_model = randomForest(Absenteeism.time.in.hours ~ ., train_1, importance = TRUE, ntree = 500)

#Predict test data using random forest model
RF_Predictions = predict(RF_model, test_1)

#Create data frame for actual and predicted values
dt_pred_RF = data.frame("actual"= test[,115],"RF_model" = RF_Predictions)
head(dt_pred_RF)


#Calculate MAE, RMSE, R-squared for testing data
print(postResample(pred = RF_Predictions, obs = test$Absenteeism.time.in.hours))
################################# Linear regression model ##############################
#Train the model using training data
LR_model = lm(Absenteeism.time.in.hours ~ ., data = train_1)


#Get the summary of the model
summary(LR_model)

#Predict the test cases
LR_predictions = predict(LR_model,test_1)


#Create data frame for actual and predicted values
dt_pred_LR = data.frame("actual"= test[,115],"LR_model" = LR_predictions)
head(dt_pred_LR)

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

Python – codes

```
#Load the libraries
import os
import numpy as np
import pandas as pd
from fancyimpute import KNN
import matplotlib.pyplot as plt
import seaborn as sns
from random import randrange, uniform
from scipy.stats import chi2_contingency
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression

# Set working directory
os.chdir("E:\edWisor\Project")

# Load the data
emp_abs = pd.read_excel("Absenteeism_at_work_Project.xls")

emp_abs.shape
emp_abs.columns
emp_abs.head(5)
emp_abs.dtypes
emp_abs.describe()

# Exploratory data analysis
emp_abs['ID'] = emp_abs['ID'].astype('category')
emp_abs['Reason for absence'] = emp_abs['Reason for absence'].replace(0,20)
emp_abs['Reason for absence'] = emp_abs['Reason for absence'].astype('category')
emp_abs['Month of absence'] = emp_abs['Month of absence'].replace(0,np.nan)
emp_abs['Month of absence'] = emp_abs['Month of absence'].astype('category')
emp_abs['Day of the week'] = emp_abs['Day of the week'].astype('category')
emp_abs['Seasons'] = emp_abs['Seasons'].astype('category')
emp_abs['Disciplinary failure'] = emp_abs['Disciplinary failure'].astype('category')
emp_abs['Education'] = emp_abs['Education'].astype('category')
emp_abs['Son'] = emp_abs['Son'].astype('category')
emp_abs['Social drinker'] = emp_abs['Social drinker'].astype('category')
emp_abs['Social smoker'] = emp_abs['Social smoker'].astype('category')
emp_abs['Pet'] = emp_abs['Pet'].astype('category')
emp_abs.dtypes
```

```
#Missing value analysis

# Creating missing value analysis
missing_val = pd.DataFrame(emp_abs.isnull().sum())

# Resetting index
missing_val = missing_val.reset_index()

# Rename variables
missing_val = missing_val.rename(columns={'index':"Variables",0:"Missing_perc"})

#Calculate percentage
missing_val['Missing_perc'] = (missing_val['Missing_perc']/len(emp_abs))*100

# Descending order
missing_val = missing_val.sort_values('Missing_perc',ascending = False).reset_index(drop = True)

# Save the missing percentage document
missing_val.to_csv("Missing_perc_emp_abs_python.csv",index = False)

missing_val

#Imputation method

emp_abs['Body mass index'].loc[5]
emp_abs['Body mass index'].loc[5] = np.nan
emp_abs['Body mass index'].loc[5]

# Impute with mean
#emp_abs['Body mass index'] = emp_abs['Body mass index'].fillna(emp_abs['Body mass index'].mean())

# Impute with median
#emp_abs['Body mass index'] = emp_abs['Body mass index'].fillna(emp_abs['Body mass
index'].median())

# KNN imputation
emp_abs = pd.DataFrame(KNN(k=3).fit_transform(emp_abs), columns = emp_abs.columns)

# Converting data in proper data type
for i in emp_abs:
   emp_abs.loc[:,i] = emp_abs.loc[:,i].round()

emp_abs['Body mass index'].iloc[5]
emp_abs.isnull().sum()


# Categorizing data based on continuous and categorical variables
```

```
continuous_names =['Transportation expense','Distance from Residence to Work','Service
time','Age','Work load Average/day ',
                'Hit target','Weight','Height','Body mass index','Absenteeism time in hours']

categorical_names =['ID','Reason for absence','Month of absence','Day of the
week','Seasons','Disciplinary failure','Education',
                 'Son','Social drinker','Social smoker','Pet']

for i in categorical_names:
   emp_abs.loc[:,i] = emp_abs.loc[:,i].round()
   emp_abs.loc[:,i] = emp_abs.loc[:,i].astype('category')

emp_abs.dtypes

# Creating backup file
df = emp_abs.copy()
# emp_abs = df.copy()

# Visualization
sns.set_style("whitegrid")
sns.catplot(data=emp_abs, x='ID', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Reason for absence', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Month of absence', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Day of the week', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Seasons', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Education', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Disciplinary failure', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Son', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Social drinker', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Social smoker', kind= 'count',height=4,aspect=2)
sns.catplot(data=emp_abs, x='Pet', kind= 'count',height=4,aspect=2)

#Outlier analysis
# Box plot to visualize outliers
%matplotlib inline
plt.boxplot(emp_abs['Work load Average/day '])
plt.title("Outlier - Work load Average/day")
plt.savefig('plot.png', dpi=300, bbox_inches='tight')

# Box plot to visualize outliers
%matplotlib inline
plt.boxplot(emp_abs['Transportation expense'])
plt.title("Outlier - Transportation expense")
plt.savefig('plot1.png', dpi=300, bbox_inches='tight')


sns.boxplot(data = emp_abs[['Distance from Residence to Work','Body mass index','Weight','Height']])
```

```
fig=plt.gcf()
fig.set_size_inches(12,12)
plt.title("Outlier - 'Distance from Residence to Work','Body mass index','Weight','Height'")
plt.savefig('plot3.png', dpi=300, bbox_inches='tight')

sns.boxplot(data = emp_abs[['Service time','Age','Absenteeism time in hours','Hit target']])
fig=plt.gcf()
fig.set_size_inches(12,12)
plt.title("Outlier - 'Service time','Age','Absenteeism time in hours','Hit target'")
plt.savefig('plot4.png', dpi=300, bbox_inches='tight')

# Detect and replace outlier with NA
#Extract quartiles
Te_q75, Te_q25 = np.percentile(emp_abs['Transportation expense'], [75 ,25])

#Calculate IQR
Te_iqr = Te_q75 - Te_q25

#Calculate inner and outer fence
Te_minimum = Te_q25 - (Te_iqr*1.5)
Te_maximum = Te_q75 + (Te_iqr*1.5)

# Replace outlier with np.nan
emp_abs.loc[emp_abs['Transportation expense']< Te_minimum,'Transportation expense'] = np.nan
emp_abs.loc[emp_abs['Transportation expense']> Te_maximum,'Transportation expense'] = np.nan

#Extract quartiles
Wl_q75, Wl_q25 = np.percentile(emp_abs['Work load Average/day '], [75 ,25])

#Calculate IQR
Wl_iqr = Wl_q75 - Wl_q25

#Calculate inner and outer fence
Wl_minimum = Wl_q25 - (Wl_iqr*1.5)
Wl_maximum = Wl_q75 + (Wl_iqr*1.5)

# Replace outlier with np.nan
emp_abs.loc[emp_abs['Work load Average/day ']< Wl_minimum,'Work load Average/day '] = np.nan
emp_abs.loc[emp_abs['Work load Average/day ']> Wl_maximum,'Work load Average/day '] = np.nan

#Extract quartiles
H_q75, H_q25 = np.percentile(emp_abs['Height'], [75 ,25])

#Calculate IQR
H_iqr = H_q75 - H_q25
```

```
#Calculate inner and outer fence
H_minimum = H_q25 - (H_iqr*1.5)
H_maximum = H_q75 + (H_iqr*1.5)

# Replace outlier with np.nan
emp_abs.loc[emp_abs['Height']< H_minimum,'Height'] = np.nan
emp_abs.loc[emp_abs['Height']> H_maximum,'Height'] = np.nan

#Extract quartiles
St_q75, St_q25 = np.percentile(emp_abs['Service time'], [75 ,25])

#Calculate IQR
St_iqr = St_q75 - St_q25

#Calculate inner and outer fence
St_minimum = St_q25 - (St_iqr*1.5)
St_maximum = St_q75 + (St_iqr*1.5)

# Replace outlier with np.nan
emp_abs.loc[emp_abs['Service time']< St_minimum,'Service time'] = np.nan
emp_abs.loc[emp_abs['Service time']> St_maximum,'Service time'] = np.nan

#Extract quartiles
A_q75, A_q25 = np.percentile(emp_abs['Age'], [75 ,25])

#Calculate IQR
A_iqr = A_q75 - A_q25

#Calculate inner and outer fence
A_minimum = A_q25 - (A_iqr*1.5)
A_maximum = A_q75 + (A_iqr*1.5)

# Replace outlier with np.nan
emp_abs.loc[emp_abs['Age']< A_minimum,'Age'] = np.nan
emp_abs.loc[emp_abs['Age']> A_maximum,'Age'] = np.nan

#Extract quartiles
At_q75, At_q25 = np.percentile(emp_abs['Absenteeism time in hours'], [75 ,25])

#Calculate IQR
At_iqr = At_q75 - At_q25

#Calculate inner and outer fence
At_minimum = At_q25 - (At_iqr*1.5)
At_maximum = At_q75 + (At_iqr*1.5)
```

```
# Replace outlier with np.nan
emp_abs.loc[emp_abs['Absenteeism time in hours']< At_minimum,'Absenteeism time in hours'] =
np.nan
emp_abs.loc[emp_abs['Absenteeism time in hours']> At_maximum,'Absenteeism time in hours'] =
np.nan

#Extract quartiles
Ht_q75, Ht_q25 = np.percentile(emp_abs['Hit target'], [75 ,25])

#Calculate IQR
Ht_iqr = Ht_q75 - Ht_q25

#Calculate inner and outer fence
Ht_minimum = Ht_q25 - (Ht_iqr*1.5)
Ht_maximum = Ht_q75 + (Ht_iqr*1.5)

# Replace outlier with np.nan
emp_abs.loc[emp_abs['Hit target']< Ht_minimum,'Hit target'] = np.nan
emp_abs.loc[emp_abs['Hit target']> Ht_maximum,'Hit target'] = np.nan

emp_abs.isnull().sum()

# KNN imputation
emp_abs = pd.DataFrame(KNN(k=3).fit_transform(emp_abs), columns = emp_abs.columns)

emp_abs.isnull().sum()

# Box plot to visualize outliers
%matplotlib inline
plt.boxplot(emp_abs['Work load Average/day '])
plt.title("Outlier - Work load Average/day")
plt.savefig('plot.png', dpi=300, bbox_inches='tight')

# Box plot to visualize outliers
%matplotlib inline
plt.boxplot(emp_abs['Transportation expense'])
plt.title("Outlier - Transportation expense")
plt.savefig('plot1.png', dpi=300, bbox_inches='tight')


sns.boxplot(data = emp_abs[['Distance from Residence to Work','Body mass index','Weight','Height']])
fig=plt.gcf()
fig.set_size_inches(12,12)
plt.title("Outlier - 'Distance from Residence to Work','Body mass index','Weight','Height'")
plt.savefig('plot3.png', dpi=300, bbox_inches='tight')
```

```
sns.boxplot(data = emp_abs[['Service time','Age','Absenteeism time in hours','Hit target']])
fig=plt.gcf()
fig.set_size_inches(12,12)
plt.title("Outlier - 'Service time','Age','Absenteeism time in hours','Hit target'")
plt.savefig('plot4.png', dpi=300, bbox_inches='tight')

# Feature selection
# Correlation analysis
df_corr = emp_abs.loc[:,continuous_names]

#Set the width and height of the plot
f, ax = plt.subplots(figsize=(10, 10))

#Generate the correlation matrix
corr = df_corr.corr()

#Plot using seaborn library(bottom, top = ax1.get_ylim(),ax1.set_ylim(bottom + 0.5, top - 0.5)) is added
to remove heat map cut-off
ax1 = sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool), cmap=sns.diverging_palette(220, 10,
as_cmap=True),
            square=True, ax=ax, annot =True)
bottom, top = ax1.get_ylim()
ax1.set_ylim(bottom + 0.5, top - 0.5)
plt.title(" Heat map")
plt.savefig('plot9.png', dpi=300, bbox_inches='tight')

# Weight shows strong correlation, hence it would be removed from the data
emp_abs = emp_abs.drop(['Weight'],axis=1)

emp_abs.columns

df = emp_abs.copy()
#emp_abs = df.copy()

#Feature Scaling

#Normality checks
%matplotlib inline
plt.hist(emp_abs['Absenteeism time in hours'], bins='auto')
plt.title("Histogram - Absenteeism time in hours")
plt.savefig('plot10.png', dpi=300, bbox_inches='tight')

cnames = ['Transportation expense','Distance from Residence to Work','Service time','Age','Work load
Average/day ',
                  'Hit target','Height','Body mass index','Absenteeism time in hours']
```

```
#Normality checks
for i in cnames:
    if i == 'Absenteeism time in hours':
        continue
    sns.distplot(emp_abs[i],bins = 'auto')
    plt.title("Checking Distribution for Variable "+str(i))
    plt.ylabel('Density')
    plt.show()

# Normalization
for i in cnames:
    if i == 'Absenteeism time in hours':
        continue
    emp_abs[i] = (emp_abs[i]-emp_abs[i].min())/(emp_abs[i].max()- emp_abs[i].min())

emp_abs.head(5)

#PCA

#Create dummy variables of factor variables
df1 = pd.get_dummies(data = emp_abs, columns = categorical_names)

df1.shape
df1.head(5)

#Data frames with dummy copy
df2 = df1.copy()

#Converting data to numpy array
D_na = df1.values

df1.columns

pca = PCA(n_components=114)
pca.fit(D_na)

#PCA variance explained
var = pca.explained_variance_ratio_
var_1 = np.cumsum(np.round(pca.explained_variance_ratio_, decimals=4)*100)

# PCA plot
plt.plot(var_1)
plt.title("PCA")
plt.savefig('plot11.png', dpi=300, bbox_inches='tight')
plt.show()
```

```
#Selecting 40 components since it explains almost 97+ % data variance
pca = PCA(n_components=40)
pca.fit(D_na)

target_v = df1['Absenteeism time in hours']

#Model development

# Divide the data into test and train (simple random sampling)
X_train, X_test, y_train, y_test = train_test_split(D_na,target_v, test_size=0.2, random_state = 1)

#Decision tree
# Decision tree regression
fit_DT = DecisionTreeRegressor(max_depth=2).fit(X_train,y_train)

#Apply model for test data
predictions_DT = fit_DT.predict(X_test)

#Create data frame for actual and predicted values
df_dt = pd.DataFrame({'actual': y_test, 'pred': predictions_DT})
print(df_dt.head())

emp_abs['Absenteeism time in hours'].iloc[681]

#Define function to calculate RMSE
def RMSE(y_true,y_pred):
    rmse = np.sqrt(mean_squared_error(y_true,y_pred))
    return rmse

#Calculate RMSE and R-squared value
print("RMSE: "+str(RMSE(y_test,predictions_DT)))
print("R^2: "+str(r2_score(y_test,predictions_DT)))

#Random forest
rf_model = RandomForestRegressor(n_estimators = 500, random_state = 1).fit(X_train,y_train)

#Perdict for test cases
predictions_RF = rf_model.predict(X_test)

#Create data frame for actual and predicted values
df_rf = pd.DataFrame({'actual': y_test, 'pred': predictions_RF})
print(df_rf.head())

#Calculate RMSE and R-squared value
print("RMSE: "+str(RMSE(y_test,predictions_RF)))
print("R^2:"+str(r2_score(y_test,predictions_RF)))
```

```
#Linear regression

lr_model = LinearRegression().fit(X_train,y_train)

predictions_LR = lr_model.predict(X_test)

#Create data frame for actual and predicted values
df_lr = pd.DataFrame({'actual': y_test, 'pred': predictions_LR})
print(df_lr.head())

#Calculate RMSE and R-squared value
print("RMSE: "+str(RMSE(y_test,predictions_LR)))
print("R^2: "+str(r2_score(y_test,predictions_LR)))

# Save the model implemented document
df1.to_csv("Dummies_emp_abs_python.csv",index = False)
```

References

1. Edwisor study materials
2. Analytics vidya blogs
3. Github - issues