**PROJECT REPORT**

**EMPLOYEE ABSENTEEISM**

**Abhirup Bhadra**

**07-03-2020**

**Table of Contents**

Introduction

1.1 Problem Statement. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .3

1.2 Variables. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .3

1.3 Sample Data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

Methodology

2.1 Pre – Processing. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

2.2 Missing Value Analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

2.3 Outlier Analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .5

2.4 Distribution of Continuous variables. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.5 Distribution of Categorical variables. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.6 Feature Selection. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .9

Modelling

3.1 Model Selection. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .10

3.2 Decision Tree. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

3.3 Random Forest. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .10

3.4 Linear Regression. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .11

Conclusion

4.1 Model Evaluation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .11

4.2 Model Selection. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

4.3 Solution of Problem Statement. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..11

Appendix

5.1 Python Code. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

6. R code. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

**Introduction**

* 1. Problem Statement

XYZ is a courier company. As we appreciate that human capital plays an important role in collection, transportation and delivery. The company is passing through genuine issue of Absenteeism. The company has shared it dataset and requested to have an answer on the following areas:

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

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

continues?

* 1. 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)

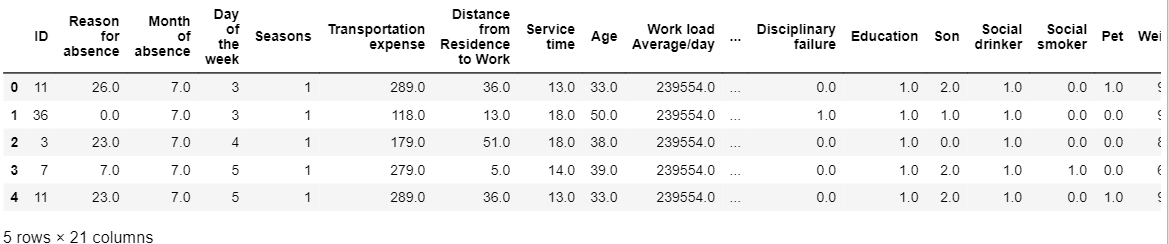
18. Weight

19. Height

20. Body mass index

21. Absenteeism time in hours (target)

* 1. Sample Data



**Methodology**

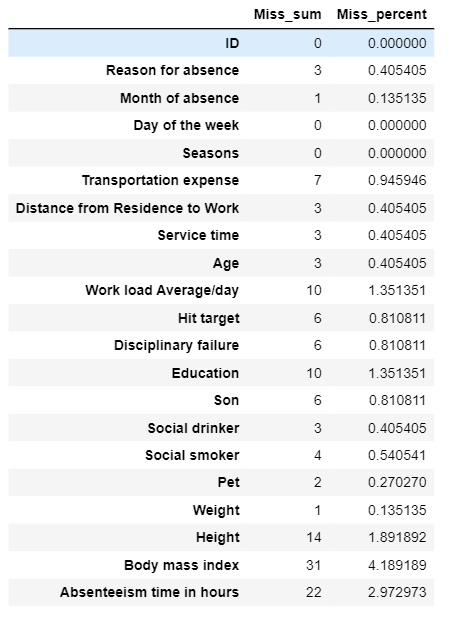
2.1 Pre – Processing

Any predictive modelling requires that we look at the data before we start modeling. However, in data mining terms looking at data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis.

2.2 Missing Value Analysis

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. If a column has more than 30% of data as missing value either we ignore the entire column or we ignore those observations. In the given data the maximum percentage of missing value is 4.189% for **body mass index** column. So, we will compute missing value for all the columns.

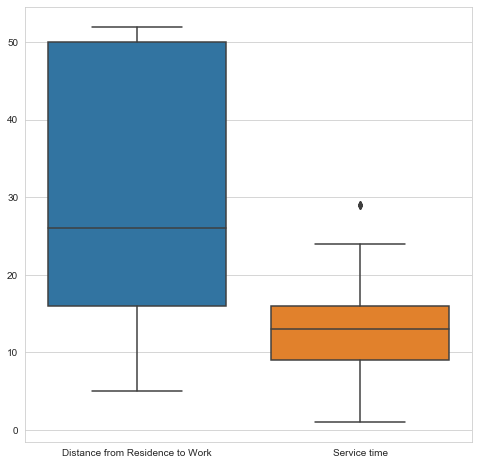
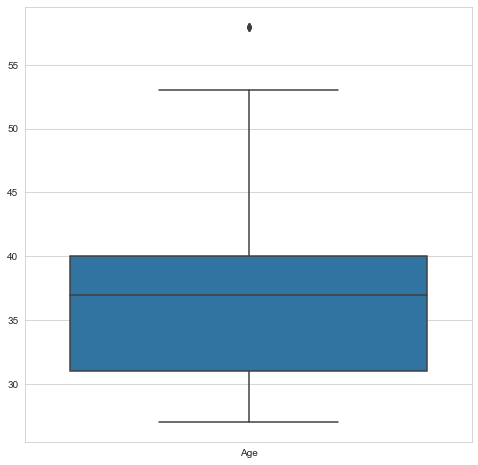
**In this project we have used KNN imputation method to impute missing value**.

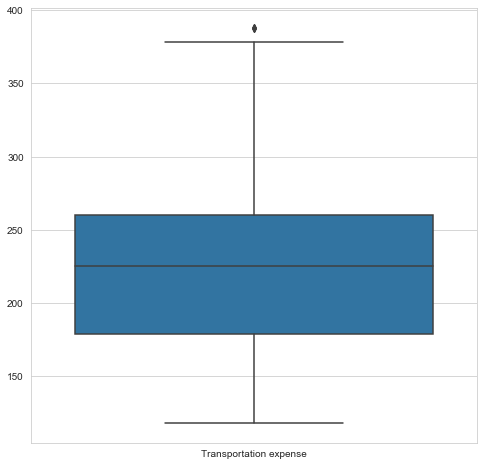
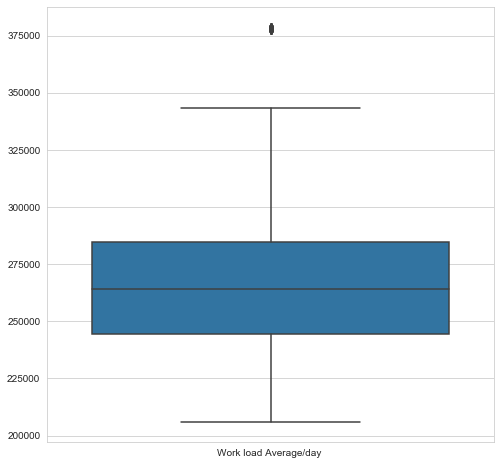


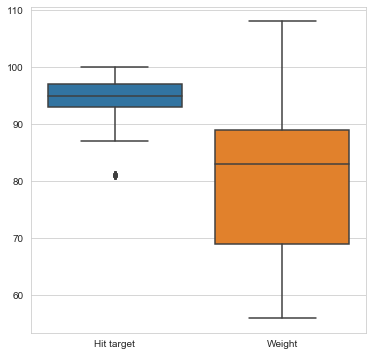
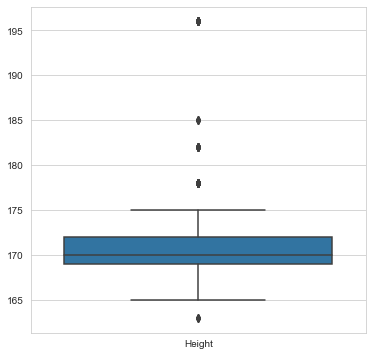
2.3 Outlier Analysis

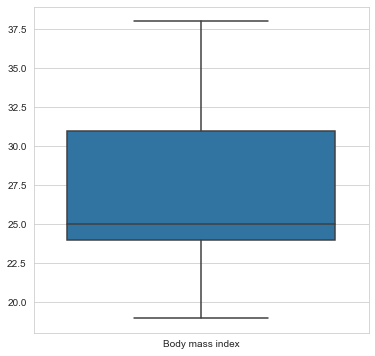
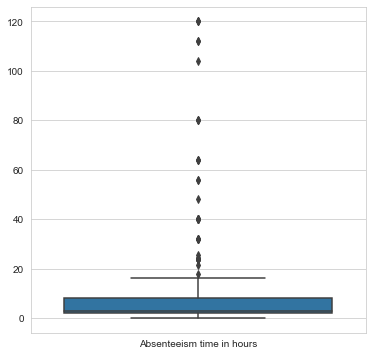
It can be observed from the distribution of variables that almost none of the variables are normally distributed. The skew in these distributions can be explained by the presence of outliers and extreme values in the data. One of the steps in pre-processing involves the detection and removal of such 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.

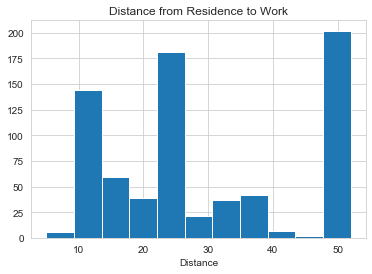
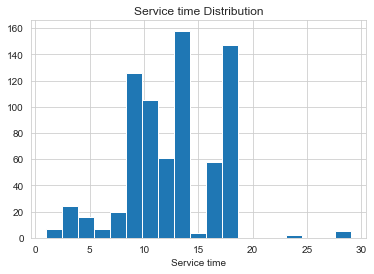
 

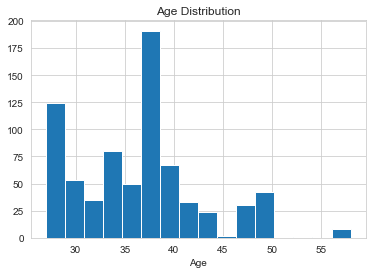
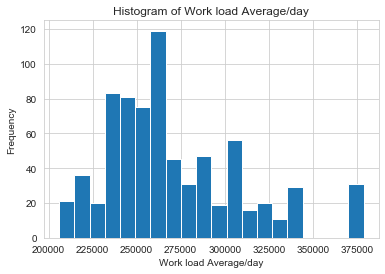
 

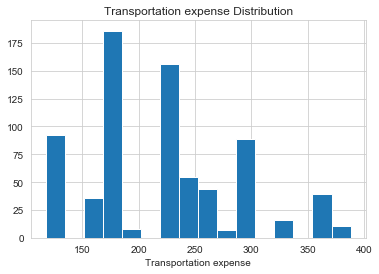
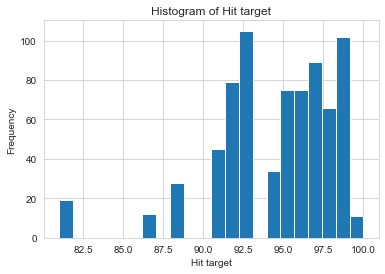
Missing values obtained from boxplots are first converted to have NA values. Then these missing values are imputed using KNN imputation method.

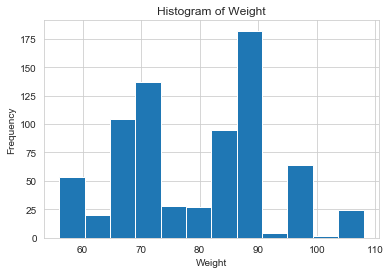
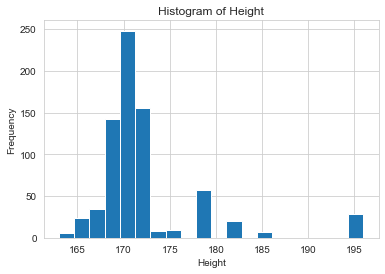
2.4 Distribution of Continuous variables

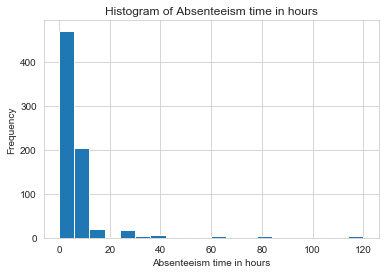
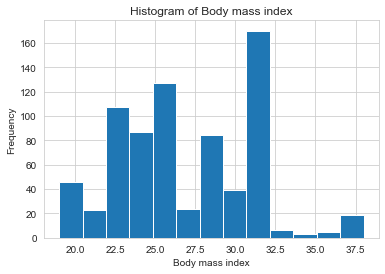
By looking at the distribution of continuous variables, it can be observed that the variables are not normally distributed. Histograms are used to observe the distribution of continuous variables.

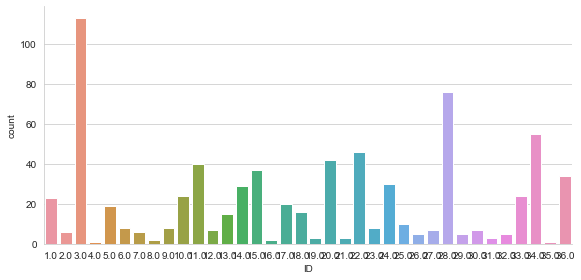
 

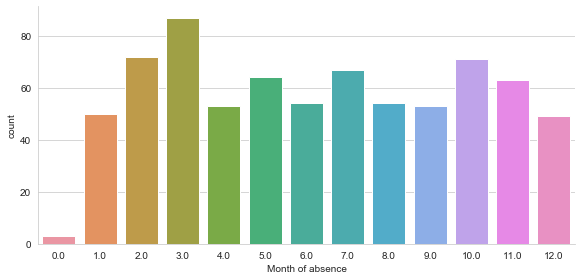


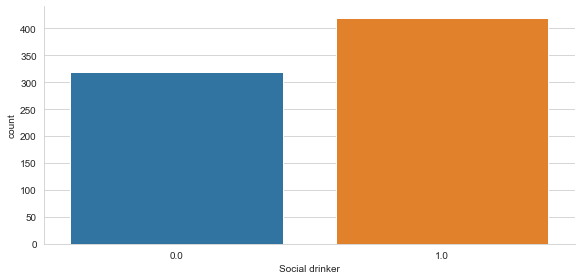
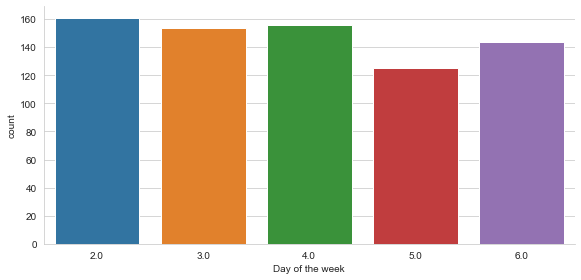
2.5 Distribution of Categorical variables

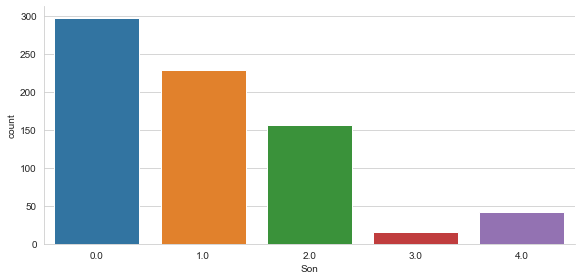
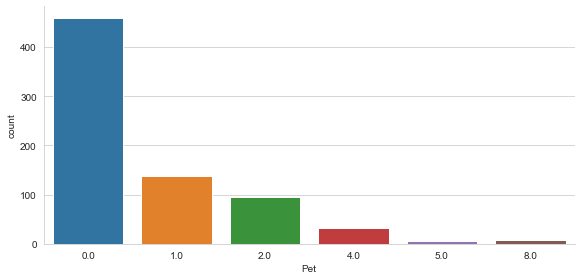
Bar graphs are used to visualize the distribution of categorical variables.

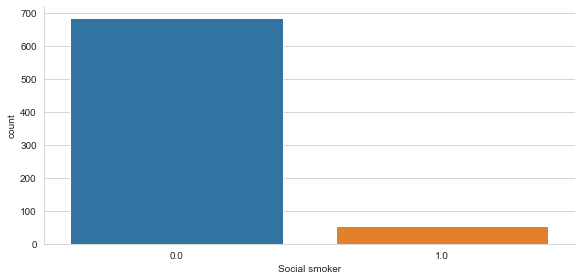
Employees who are social drinkers have more absent hours than those who do not drink. Employees having zero, one or two children have more absent hours. Employees with ID number 3 and 28 are absent the most. Employees are absent the most on Mondays and the least on Thursdays. Reason 23 and 28 are the 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.







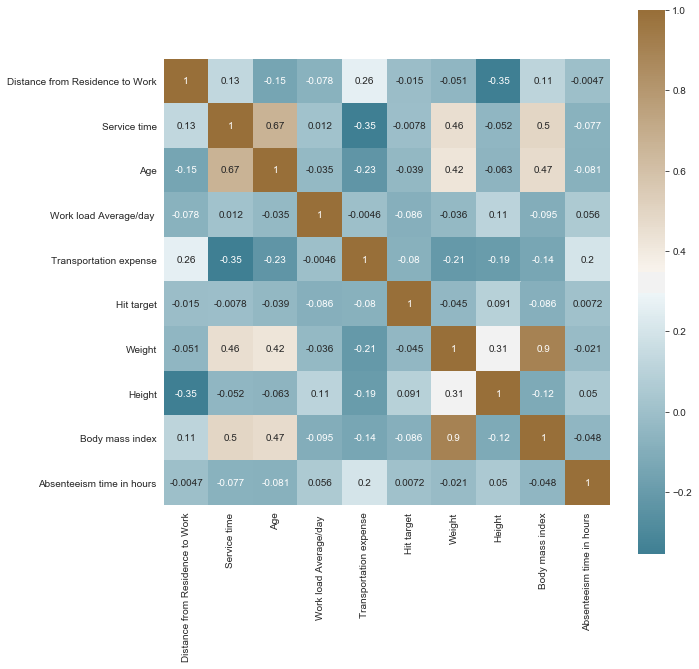




2.6 Feature Selection

Before performing any type of modelling, we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. Selecting subset of relevant columns for the model construction is known as Feature Selection. We cannot use all the features because some features may be carrying the same information or irrelevant information which can increase

overhead. To reduce overhead, we adopt feature selection technique to extract meaningful features out of data. This in turn helps us to avoid the problem of multi collinearity. In this project we have selected **Correlation Analysis** for numerical variable



**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

Using decision tree, we can predict the value target variable. In R RMSE for this model is 2.276. In python the RMSE for this decision tree is 3.648.

3.3 Random Forest

The number of ntrees used for prediction in the forest is 500 in R. RMSE for this model is 2.194. In python the RMSE for this random forest is 3.053.

3.4 Linear Regression

In R RMSE for this model is 2.559. In python the RMSE for this decision tree is 3.099.

**Conclusion**

4.1 Model Evaluation

**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. **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** indicate better fit.

4.2 Model Selection

|  |  |  |
| --- | --- | --- |
| Model | RMSE in R | RMSE in Python |
| Decision Tree | 2.276 | 3.648 |
| Random Forest | 2.194 | 3.053 |
| Linear Regression | 2.559 | 3.099 |

From above observation it is clear that Random Forest model is best suited for this case of validation.

4.3 Solution of Problem Statement

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

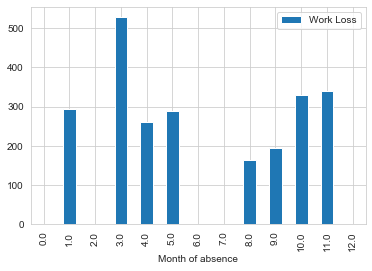
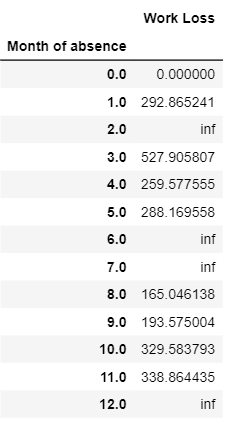
Solutions:

1. It is observed that employee with low education have maximum absentee time. A firm should hire educated staff as staff with low education is having most absenteeism hours.
2. Some employee with **ID 3, 28, 34** are often absent from work, company should take action against them. Or even a warning to them might help
3. Employees who are social smoker have more absentee hour than who are not social smoker. As a smoker is more prone to bad health condition so that causes a lot of absenteeism. So a firm should conduct health campaigns to educated employee about the harmful effects of smoking
4. Most often Reason for absence are medical consultation and dental consultation, company should take care of it. The maximum people taking the absent hours are from category 23 followed by 28 and 27. These category are not attested by doctors. 23: Medical Consultation. 28: Dental consultation 27: Physiotherapy .

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

Ans:

Considering the losses to be the absenteeism time in hours, if the same trend of absenteeism continues, then the total 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 527.90 hours. Employees are absent the least in the month of August, with total Absenteeism hours equal to 165.04 hours.

**Appendix**

5.1 Python Code

import pandas as pd

import numpy as np

import os

#import libraries for plots

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

#Setting working directory

os.chdir("E:\data science\_edwisor\Project 2 Employee Absenteesim")

#Reading dataset

data = pd.read\_excel("Absenteeism\_at\_work\_Project.xls")

data.head()

**Exploratory Data Analysis**[**¶**](http://localhost:8888/notebooks/Project%20Employee%20Absenteeism.ipynb#Exploratory-Data-Analysis)

data.shape

data.dtypes

data.nunique()

# From the EDA and problem statement file categorising the variables in two category " Continuos" and "Categorical"

categorical\_vars = ['ID','Reason for absence','Month of absence','Day of the week',

'Seasons','Disciplinary failure', 'Education', 'Social drinker',

'Social smoker', 'Pet', 'Son']

continuous\_vars = ['Distance from Residence to Work', 'Service time', 'Age', 'Work load Average/day ', 'Transportation expense',

'Hit target', 'Weight', 'Height', 'Body mass index', 'Absenteeism time in hours']

# Trnasforming data type

data['ID']= data['ID'].astype('category')

data['Reason for absence']= data['Reason for absence'].astype('category')

data['Month of absence']= data['Month of absence'].astype('category')

data['Day of the week']= data['Day of the week'].astype('category')

data['Seasons']= data['Seasons'].astype('category')

data['Disciplinary failure']= data['Disciplinary failure'].astype('category')

data['Education']= data['Education'].astype('category')

data['Social drinker']= data['Social drinker'].astype('category')

data['Social smoker']= data['Social smoker'].astype('category')

data['Pet']= data['Pet'].astype('category')

data['Son']= data['Son'].astype('category')

data.isnull().sum()

# Missing value analysis

Miss\_data = pd.DataFrame(data.isnull().sum())

Miss\_data = Miss\_data.rename(columns={0:"Miss\_sum"})

Miss\_data["Miss\_percent"] = (Miss\_data["Miss\_sum"]/len(data))\*100

Miss\_data

# Missing value imputation

data['Body mass index'].iloc[1]

#Set the value of first row in Body mass index as NAN

#create missing value

data['Body mass index'].iloc[1] = np.nan#Apply KNN imputation algorithm

data = pd.DataFrame(KNN(k = 3).fit\_transform(data), columns = data.columns)

data['Body mass index'].iloc[1]

#Round the values of categorical values

for i in categorical\_vars:

data.loc[:,i] = data.loc[:,i].round()

data.loc[:,i] = data.loc[:,i].astype('category')

#Check if any missing values

data.isnull().sum()

# Distribution of data using graphs

#Check the bar graph of categorical Data using factorplot

sns.set\_style("whitegrid")

sns.factorplot(data=data, x='Reason for absence', kind= 'count',size=4,aspect=2)

sns.factorplot(data=data, x='Seasons', kind= 'count',size=4,aspect=2)

sns.factorplot(data=data, x='Education', kind= 'count',size=4,aspect=2)

sns.factorplot(data=data, x='Disciplinary failure', kind= 'count',size=4,aspect=2)

#Check the distribution of numerical data using histogram

plt.hist(data=data, x='Distance from Residence to Work', bins='auto', label='Distance from Residence to Work')

plt.xlabel('Distance')

plt.title("Distance from Residence to Work")

# Outlier Analysis

#Check for outliers in data using boxplot

sns.boxplot(data=data[['Distance from Residence to Work', 'Service time']])

fig=plt.gcf()

fig.set\_size\_inches(8,8)

# list of variables which doesn't have outlier

neglect = ['Distance from Residence to Work', 'Weight', 'Body mass index']

# Looping over all continuou variables to detect and remove Outliers

for i in continuous\_vars:

# Avoiding the variables which doesn't have outlier

if i in neglect:

continue

# Getting 75 and 25 percentile of variable "i"

q75, q25 = np.percentile(data[i], [75,25])

# Calculating Interquartile range

iqr = q75 - q25

# Calculating upper extream and lower extream

minimum = q25 - (iqr\*1.5)

maximum = q75 + (iqr\*1.5)

# Replacing all the outliers value to NA

data.loc[data[i]< minimum,i] = np.nan

data.loc[data[i]> maximum,i] = np.nan

# Impute missing values with KNN

data = pd.DataFrame(KNN(k = 3).fit\_transform(data), columns = data.columns)

# Checking if there is any missing value

data.isnull().sum()

# Feature selection

#Get dataframe with all continuous variables

data\_corr = data.loc[:,continuous\_vars]

#Check for multicollinearity using corelation graph

#Set the width and hieght of the plot

f, ax = plt.subplots(figsize=(10, 10))

#Generate correlation matrix

corr = data\_corr.corr()

#Plot using seaborn library

sns.heatmap(corr, mask=np.zeros\_like(corr, dtype=np.bool),

cmap=sns.diverging\_palette(220, 50, as\_cmap=True),

square=True, ax=ax, annot = True)

plt.plot()

#Variable Reduction

to\_drop = ['Weight']

data = data.drop(to\_drop, axis = 1)

# Updating the Continuous and Categorical Variables

continuous\_vars.remove('Weight')

# Machine Learning Models

#Splitting data into train and test data

from sklearn.model\_selection import train\_test\_split

train, test = train\_test\_split(data, test\_size=0.2,random\_state=123)

# Importing libraries for Decision Tree

from sklearn.tree import DecisionTreeRegressor

#train the model

dt\_data = DecisionTreeRegressor(random\_state=123).fit(train.iloc[:,0:19],train.iloc[:,19])

#Prediction of the hr

dt\_data\_pred = dt\_data.predict(test.iloc[:,0:19])

#Creating dataframe for actual and predicted value

dtpred\_data = pd.DataFrame({'actual': test.iloc[:,19], 'pred': dt\_data\_pred})

RMSE

math.sqrt(mean\_squared\_error(test.iloc[:,19],dt\_data\_pred))

#Import library for RandomForestRegressor

from sklearn.ensemble import RandomForestRegressor

#Train the model

rf\_data = RandomForestRegressor(n\_estimators=200,random\_state=123).fit(train.iloc[:,0:19], train.iloc[:,19])

#Prediction on hr

rf\_data\_prd = rf\_data.predict(test.iloc[:,0:19])

#Creating dataframe for actual and predicted value

rfpred\_data = pd.DataFrame({'actual': test.iloc[:,19], 'pred': rf\_data\_prd})

rfpred\_data.head()

#RMSE

math.sqrt(mean\_squared\_error(test.iloc[:,19],rf\_data\_prd))

#import libraries for Linear regression

import statsmodels.api as sm

#Train the model

lr\_data = sm.OLS(train.iloc[:,19].astype(float), train.iloc[:,0:19].astype(float)).fit()

#Check the summary of model

lr\_data.summary()

#Prediction of hr

lr\_data\_prd = lr\_data.predict(test.iloc[:,0:19])

#Creating dataframe for actual and predicted value

rlpred\_data = pd.DataFrame({'actual': test.iloc[:,19], 'pred': lr\_data\_prd})

rlpred\_data.head()

#RMSE

math.sqrt(mean\_squared\_error(test.iloc[:,19],lr\_data\_prd))

# saving the best model(Random Forest) output data

result=pd.DataFrame(test.iloc[:,0:19])

result['pred\_cnt'] = (rf\_data\_prd)

result.to\_csv("Random forest output python.csv",index=False)

5.2 R Code

#Clean the environment

rm(list = ls())

# Setting the working directory

setwd("E:/data science\_edwisor/Project 2 Employee Absenteesim")

getwd()

#Load the librarires

libraries = c("dummies","caret","rpart.plot","plyr","dplyr", "ggplot2","rpart","dplyr","DMwR","randomForest","usdm","corrgram","DataCombine")

lapply(X = libraries,require, character.only = TRUE)

rm(libraries)

#Read the csv file

data1 = read.csv('Absenteeism\_at\_work\_Project.csv', header = TRUE)

########################################EXPLORE THE DATA########################################

#Check number of rows and columns

dim(data1)

#Structure of variables

str(data1)

#Transform data types

data1$ID = as.factor(as.character(data1$ID))

data1$Reason.for.absence[data1$Reason.for.absence %in% 0] = 20

data1$Reason.for.absence = as.factor(as.character(data1$Reason.for.absence))

data1$Month.of.absence[data1$Month.of.absence %in% 0] = NA

data1$Month.of.absence = as.factor(as.character(data1$Month.of.absence))

data1$Day.of.the.week = as.factor(as.character(data1$Day.of.the.week))

data1$Seasons = as.factor(as.character(data1$Seasons))

data1$Disciplinary.failure = as.factor(as.character(data1$Disciplinary.failure))

data1$Education = as.factor(as.character(data1$Education))

data1$Son = as.factor(as.character(data1$Son))

data1$Social.drinker = as.factor(as.character(data1$Social.drinker))

data1$Social.smoker = as.factor(as.character(data1$Social.smoker))

data1$Pet = as.factor(as.character(data1$Pet))

# From the above EDA and problem statement categorising data in 2 category "continuous" and "catagorical"

#continuous\_vars = c('Distance.from.Residence.to.Work', 'Service.time', 'Age',

'Work.load.Average.day.', 'Transportation.expense',

'Hit.target', 'Weight', 'Height',

'Body.mass.index', 'Absenteeism.time.in.hours')

#catagorical\_vars = c('ID','Reason.for.absence','Month.of.absence','Day.of.the.week',

'Seasons','Disciplinary.failure', 'Education', 'Social.drinker',

'Social.smoker', 'Son', 'Pet')

#######################################MISSING VALUE ANALYSIS########################################

#Get number of missing values

sapply(data1,function(x){sum(is.na(x))})

missing\_values = data.frame(sapply(data1,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(data1)) \*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

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

#data1[["Body.mass.index"]][3]

#data1[["Body.mass.index"]][3] = NA

#data1[["Body.mass.index"]][3] = mean(data1$Body.mass.index, na.rm = T)

#data1[["Body.mass.index"]][3] = median(data1$Body.mass.index, na.rm = T)

data1 = kNN(data = data1, k = 5)

#Check if any missing values

sum(is.na(data1))

########################################EXPLORE DISTRIBUTION USING GRAPHS########################################

#Get numerical data

numeric\_index = sapply(data1, is.numeric)

numeric\_data = data1[,numeric\_index]

#Distribution of factor data using bar plot

bar1 = ggplot(data = data1, aes(x = ID)) + geom\_bar() + ggtitle("Count of ID") + theme\_bw()

bar2 = ggplot(data = data1, aes(x = Reason.for.absence)) + geom\_bar() +

ggtitle("Count of Reason for absence") + theme\_bw()

bar3 = ggplot(data = data1, aes(x = Month.of.absence)) + geom\_bar() + ggtitle("Count of Month") + theme\_bw()

bar4 = ggplot(data = data1, aes(x = Disciplinary.failure)) + geom\_bar() +

ggtitle("Count of Disciplinary failure") + theme\_bw()

bar5 = ggplot(data = data1, aes(x = Education)) + geom\_bar() + ggtitle("Count of Education") + theme\_bw()

bar6 = ggplot(data = data1, aes(x = Son)) + geom\_bar() + ggtitle("Count of Son") + theme\_bw()

bar7 = ggplot(data = data1, 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 = data1, aes(x =Transportation.expense)) +

ggtitle("Transportation.expense") + geom\_histogram(bins = 25)

hist2 = ggplot(data = data1, aes(x =Height)) +

ggtitle("Distribution of Height") + geom\_histogram(bins = 25)

hist3 = ggplot(data = data1, aes(x =Body.mass.index)) +

ggtitle("Distribution of Body.mass.index") + geom\_histogram(bins = 25)

hist4 = ggplot(data = data1, 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(data1, is.numeric)

numeric\_data = data1[,numeric\_index]

#Get the data with only factor columns

factor\_data = data1[,!numeric\_index]

#Check for outliers using boxplots

for(i in 1:ncol(numeric\_data)) {

assign(paste0("box",i), ggplot(data = data1, 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,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 = data1[,i][data1[,i] %in% boxplot.stats(data1[,i])$out]

print(paste(i,length(val)))

data1[,i][data1[,i] %in% val] = NA

}

#Check number of missing values

sapply(data1,function(x){sum(is.na(x))})

#Get number of missing values after replacing outliers as NA

missing\_values\_out = data.frame(sapply(data1,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(data1)) \*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

data1 = kNN(data1, k = 5)

#Check if any missing values

sum(is.na(data1))

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

## Correlation Plot

corr = round(cor(numeric\_data),2)

ggcorrplot(corr,hc.order = T,

type = "full",

lab = T,

lab\_size = 3,

method = "square",

colors = c("blue","white","darkgreen"),

title = "Correlation Plot",

ggtheme = theme\_bw)

## Dimension Reduction

data1 = subset(data1, select = -c(Body.mass.index))

########################################FEATURE SCALING########################################

#Normality check

hist(data1$Absenteeism.time.in.hours)

#Remove dependent variable

numeric\_index = sapply(data1,is.numeric)

numeric\_data = data1[,numeric\_index]

numeric\_columns = names(numeric\_data)

numeric\_columns = numeric\_columns[-9]

#Normalization of continuous variables

for(i in numeric\_columns){

print(i)

data1[,i] = (data1[,i] - min(data1[,i]))/

(max(data1[,i]) - min(data1[,i]))

}

### Modeling

#Dividing into test and train seta

t\_idx = sample(1:nrow(data1), 0.8\*nrow(data1))

train = data1[t\_idx,]

test = data1[-t\_idx,]

# Removing all the custom variables from the memory

rmExcept(c("test","train","data1"))

########################################DECISION TREE########################################

#RMSE: 2.276

#MAE: 1.694

#R squared: 0.44

#Build decsion tree using rpart

dt\_model = rpart(Absenteeism.time.in.hours ~ ., data = train, method = "anova")

#Perdict 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)

#Calcuate MAE, RMSE, R-sqaured for testing data

print(postResample(pred = dt\_predictions, obs = test[,115]))

########################################RANDOM FOREST########################################

#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)

#Calcuate MAE, RMSE, R-sqaured for testing data

print(postResample(pred = rf\_predictions, obs = test[,115]))

########################################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)

#Calcuate MAE, RMSE, R-sqaured for testing data

print(postResample(pred = lr\_predictions, obs = test[,115]))