

**Contents**

**1. Introduction**

[1.1 Problem Statement](#_Toc29919) 3

[1.2 Dataset](#_Toc29920) 3

[1.3 Exploratory Data Analysis](#_Toc29921) 5

[**2. Methodology**](#_Toc29922)

[2.1 Data Pre Processing](#_Toc29923) 10

[2.1.1 Missing Value Analysis 10](#_Toc29924)

[2.1.2 Outlier Analysis 11](#_Toc29925)

[2.1.3 Feature Selection](#_Toc29926) 14

[2.1.4 Feature Scaling](#_Toc29927) 17

[2.2 Model Development 17](#_Toc29928)

[2.2.1 Decision Tree 17](#_Toc29929)

[2.2.2 Random Forest 18](#_Toc29930)

[2.2.3 Linear Regression 20](#_Toc29931)

[2.2.4 Gradient Boosting 21](#_Toc29932)

[**3. Conclusion**](#_Toc29933)

[3.1 Model Evaluation](#_Toc29934) 22

[3.2 Model Selection](#_Toc29935) 23

[3.3 Answers of asked questions](#_Toc29936) 23

[**4. Coding**](#_Toc29937)

[4.1 R Coding](#_Toc29938) 31

[4.2 Python Coding 41](#_Toc29939)

[**References**](#_Toc29940) 64

**1. Introduction**

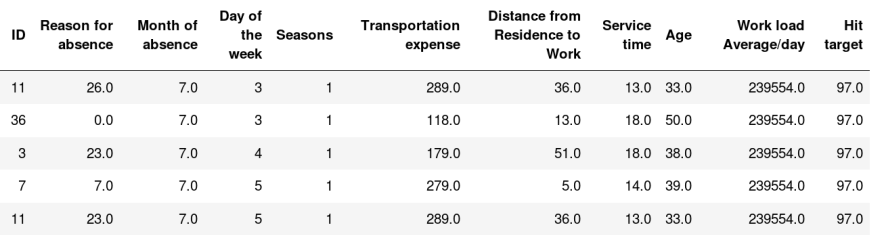
##### **1.1 Problem Statement**

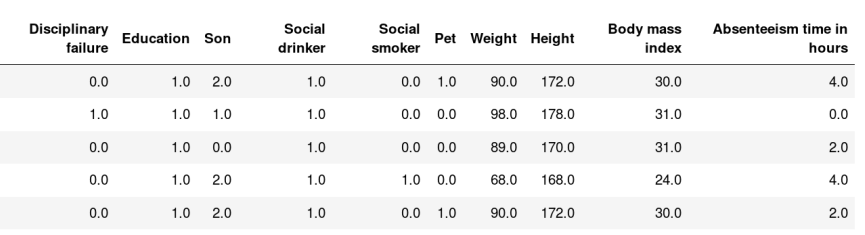
XYZ is a courier company. As we appreciate that human capital plays an important role in collection, transportation and delivery. The company is passing through genuine issue of Absenteeism. The company has shared it dataset and requested to have an answer on the following areas:

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

**1.2 Dataset**

Sample Dataset-





Dataset has 21 variables in which 20 variables are independent and 1 (Absenteeism time in hours) is dependent variable. Since target variable is continuous in nature, this is a regression problem.

**Attribute Information:**

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

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

1. Certain infectious and parasitic diseases
2. Neoplasms
3. Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
4. Endocrine, nutritional and metabolic diseases
5. Mental and behavioural disorders
6. Diseases of the nervous system
7. Diseases of the eye and adnexa
8. Diseases of the ear and mastoid process
9. Diseases of the circulatory system
10. Diseases of the respiratory system
11. Diseases of the digestive system
12. Diseases of the skin and subcutaneous tissue
13. Diseases of the musculoskeletal system and connective tissue
14. Diseases of the genitourinary system
15. Pregnancy, childbirth and the puerperium
16. Certain conditions originating in the perinatal period
17. Congenital malformations, deformations and chromosomal abnormalities
18. Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
19. Injury, poisoning and certain other consequences of external causes
20. External causes of morbidity and mortality
21. 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)**.

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

### **1.3 Exploratory Data Analysis**

Any predictive modelling requires that we look at the data before we start modelling. 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. To start this process we will first try and look at all the probability distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the probability distributions or probability density functions of the variable.

Exploratory Data Analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics. In the given data set there are 21 variables and data types of all variables are either float64 or int64. There are 740 observations and 21 columns in our data set. Missing value is also present in our data.

**ID int64**

**Reason for absence float64**

**Month of absence float64**

**Day of the week int64**

**Seasons int64**

**Transportation expense float64**

**Distance from Residence to Work float64**

**Service time float64**

**Age float64**

**Work load Average/day float64**

**Hit target float64**

**Disciplinary failure float64**

**Education float64**

**Son float64**

**Social drinker float64**

**Social smoker float64**

**Pet float64**

**Weight float64**

**Height float64**

**Body mass index float64**

**Absenteeism time in hours float64**

From EDA we have concluded that there are 10 continuous variable and 11 categorical variables.

They are

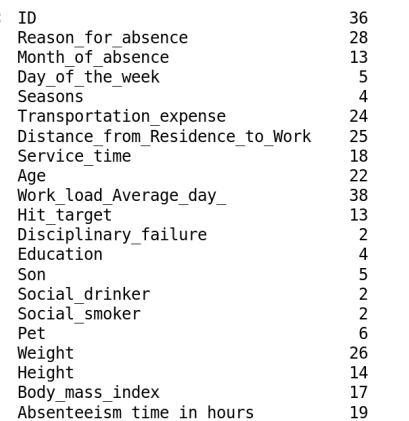
**Continuous variables in dataset:**

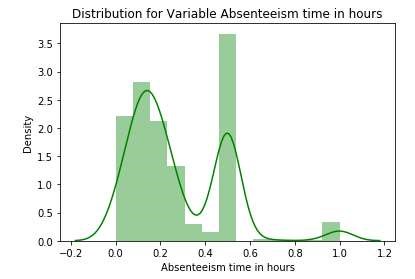
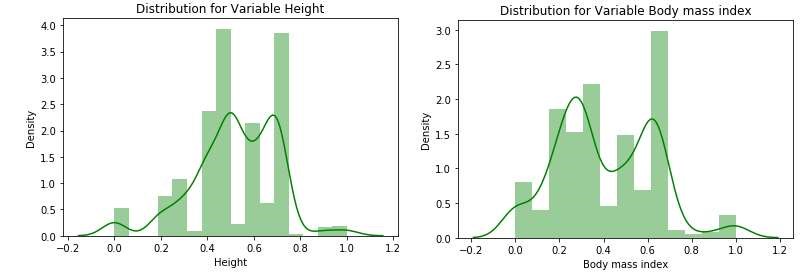
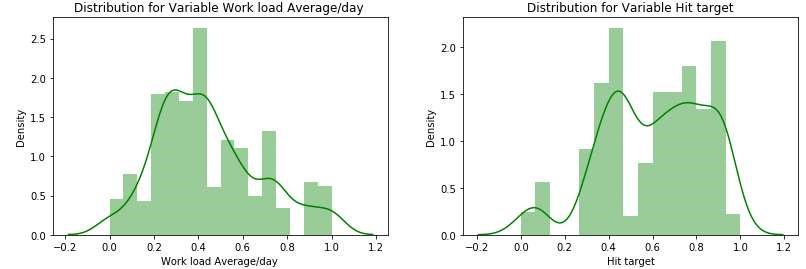
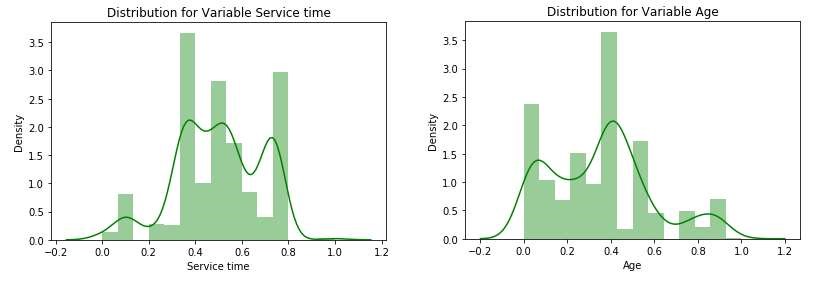
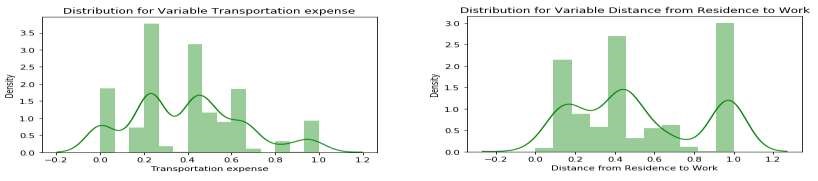
* Transportation expense
* Distance from Residence to Work
* Service time
* Age
* Work load Average/day
* Hit target
* Weight
* Height
* Body mass index
* Absenteeism time in hours

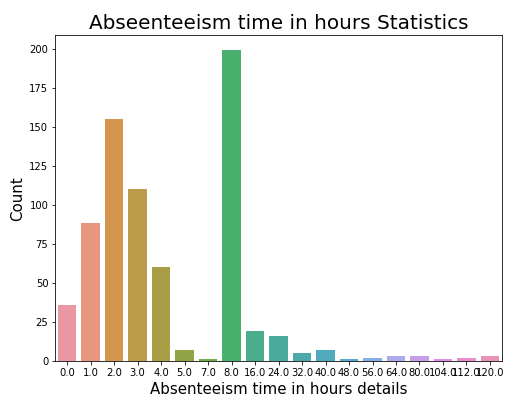
**Categorical variables in dataset:**

* ID
* Reason for absence
* Month of absence
* Day of the week
* Seasons
* Disciplinary failure
* Education
* Son
* Social drinker
* Social smoker
* Pet

From EDA we found the number of unique values for each variable in Dataset







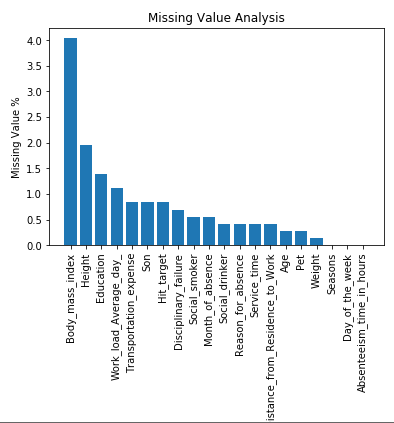
# **2. Methodology**

Before feeding the data to the model we need to clean the data and convert it to a proper format. It is the most crucial part of data science. In this we have to apply different preprocessing techniques to clean the data and to convert it into proper format.

##### **2.1 Data Pre-Processing**

#### **2.1.1 Missing Value Analysis**

In statistics, *missing data*, or *missing values*, occur when no *data value* is stored for the variable in an observation. 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.

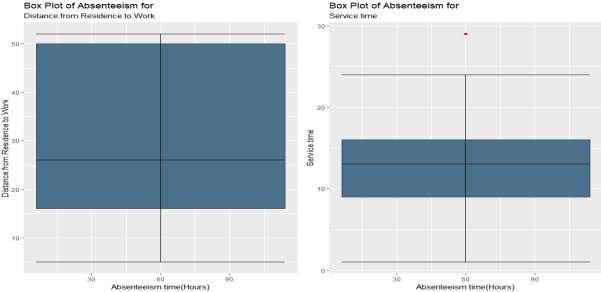


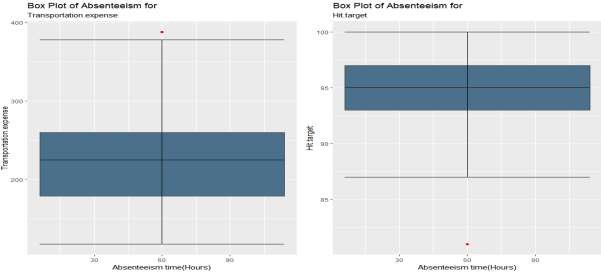
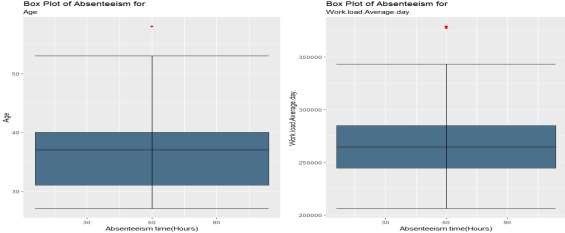
In our project we applied different techniques to impute the numeric missing value like- Mean, Median and KNN Imputation method. To select which method is suitable for imputing missing values we have used two techniques that is taking sample variable and creating missing values manually and checking mean median and KNN and another method is standard deviation method, in this method before imputing we have to check standard deviation and after imputing we have to check for standard deviation. In our data we selected **KNN Imputation,** as we found it more accurate for continuous variable while testing on sample dataset as compare to others. For categorical variables we used mode method to impute the missing values. After applying mode for categorical and KNN for continuous variables we recheck the missing values, now data is free from missing values.

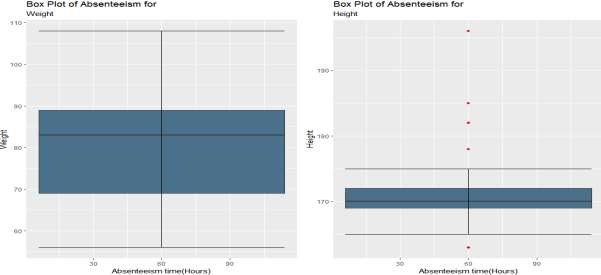
2.1.2 Outlier Analysis

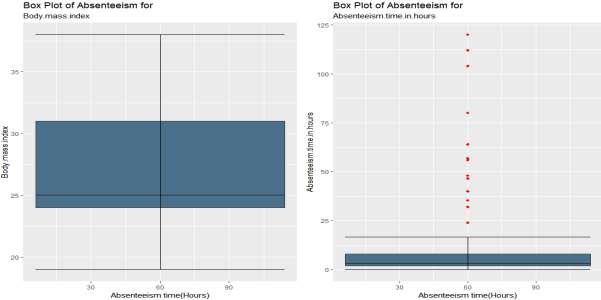
We can clearly observe from these probability distributions that most of the variables are skewed. The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data. One of the other steps of pre-processing apart from checking for normality is the presence of outliers. In this case we use a classic approach of removing outliers. We visualize the outliers using boxplots.

In figure we have plotted the boxplots of the 11 predictor variables with respect to target variable **Absenteeism time in hours** and detect the outliers by visualization.









From the boxplot almost all the variables **except “Distance from residence to work”, “Weight” and “Body mass index”** consists of outliers. From the boxplot visualization we also can see Maximum outliers are present in variables **“Height**” and **“Absenteeism time in hours”** . We have converted the outliers (data beyond minimum and maximum values) as NA i.e. missing values and fill them by **KNN** imputation method.

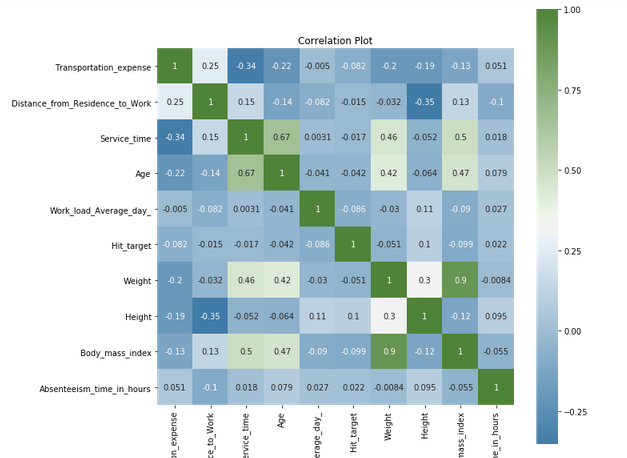
###### **2.1.3 Feature Selection**

Before performing any type of modeling, we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of 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 and **ANOVA** (Analysis of variance) for categorical variables.

**1) Correlation Analysis**

Correlation Analysis is a technique which helps to determine how strongly two features are related to each other (i.e their co-variance). As the co-variance can vary from - infinity to + infinity, the correlation is used as it is a scaled version of the co-variance having values ranging from -1 to +1. A correlation plot is shown in below figure. A correlation threshold of 0.8 is set and feature pairs of which exceeds this threshold, one of them is dropped.

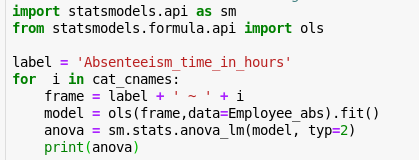
**Correlation Analysis plot for continuous variables-**

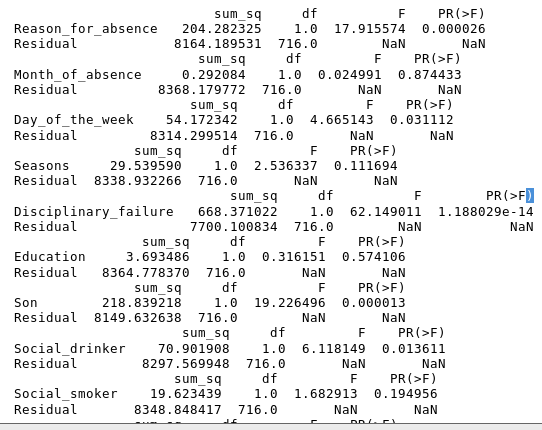


From above correlation analysis we have found that **Weight** and **Body mass index** has high correlation (>0.7), so we have excluded the **Weight** column.

**2) Analysis Of Variance (ANOVA)**

Analysis of Variance (ANOVA) is a statistical technique used to compute and compare the mean between two or more groups of observations. ANOVA makes use of two variables which are categorical variables and numeric variables of the data set. The python code is used to compute the p values of the feature and target variables. Also, the output p- values to the threshold value of 0.05 and saves the feature names to be dropped. Code and the output of the code is shown below:





from ANOVA analysis we have found that in categorical variables **Pet**, **Social smoker**, **Education** and **Month of absence** have the pr(>0.05), so we excluded them. After Correlation Analysis we have remaining variables are-

**Continuous variables in dataset-**

* Transportation expense
* Distance from Residence to Work
* Service time
* Age
* Work load Average/day
* Hit target
* Height
* Body mass index
* Absenteeism time in hours

**Categorical variables in dataset-**

* Reason for absence
* Day of the week
* Disciplinary failure
* Son
* Social drinker

2.1.4 Feature Scaling

**Feature scaling** is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Since our data is not uniformly distributed we will use **Normalization** as Feature Scaling Method.

##### **2.2 Model Development**

After Data pre-processing the next step is to develop a model using a train or historical data Which can perform to predict accurate result on test data or new data. Here we have tried with different model and will choose the model which will provide the most accurate values.

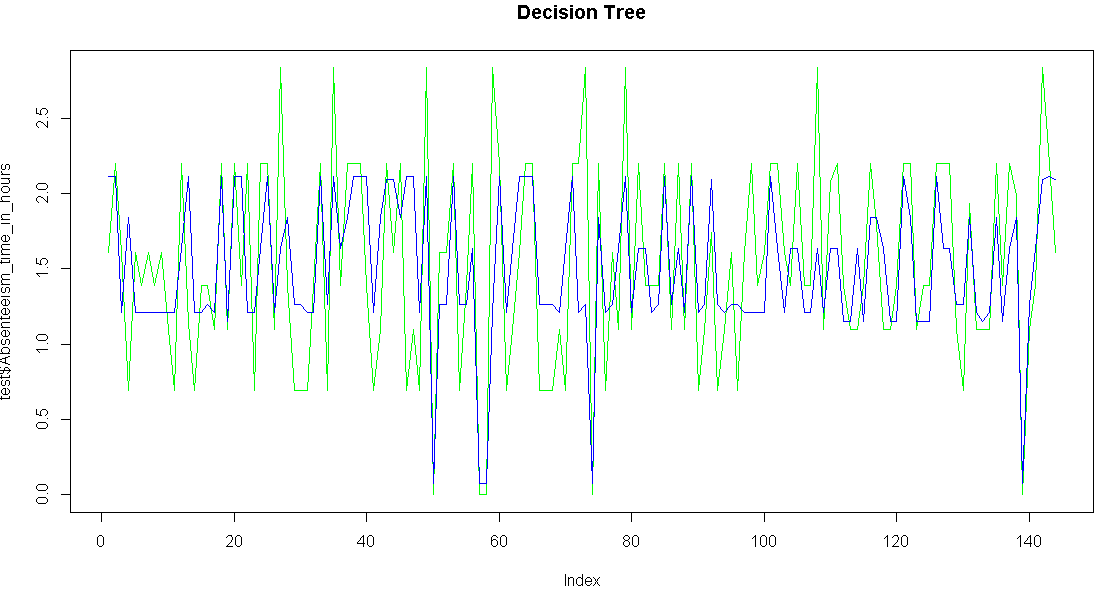
**2.2.1 Decision Tree**

Decision Tree is a supervised machine learning algorithm, which is used to predict the data for classification and regression. It accepts both continuous and categorical variables. A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Each branch connects nodes with “and” and multiple branches are connected by “or”. Extremely easy to understand by the business users. It provides its output in the form of rule, which can easily understood by a non-technical person also.

We have prepared a model by using decision tree algorithm and calculate RMSE value and R^2 value for our project in R and Python are –

|  |  |  |
| --- | --- | --- |
| **Decision Tree** | **R** | **PYTHON** |
| **RMSE Train** | 0.4369 | 0.07884 |
| **RMSE Test** | 0.5030 | 0.5077 |
| **R^2 Train** | 0.5405 | 0.9854 |
| **R^2 Test** | 0.4366 | 0.4375 |

**Visualization for test and predicted test in Decision Tree-**



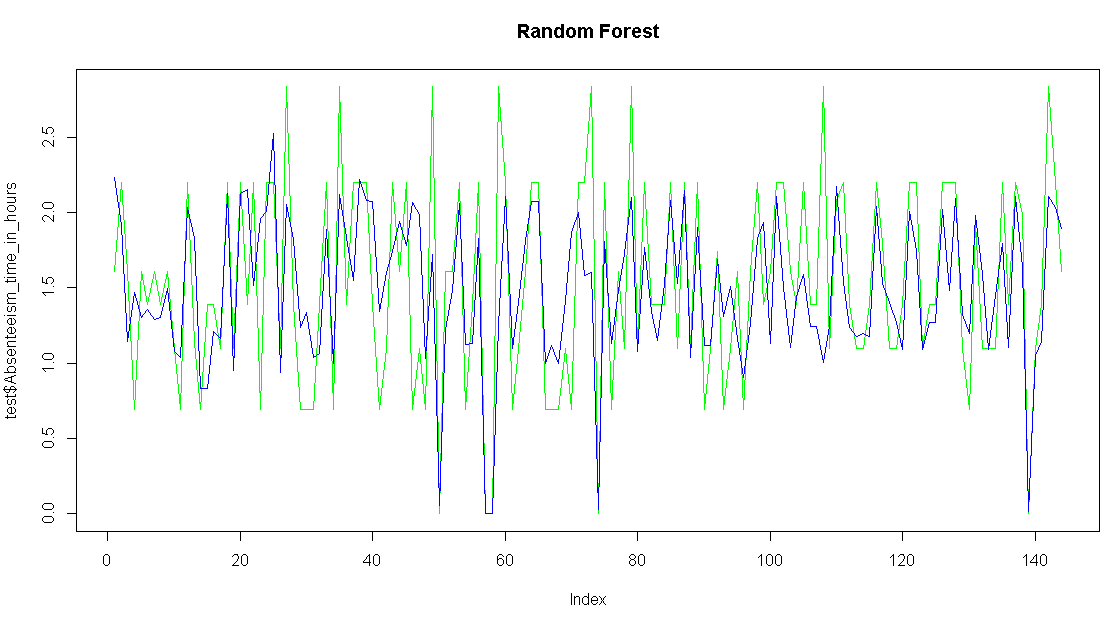
2.2.2 Random Forest

Random Forest is an ensemble technique that consists of many decision trees. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. It is called random forest as we are building n number of trees randomly. In other words,

to build the decision trees it selects randomly n number of variables and n number of observations. It means to build each decision tree on random forest we are not going to use the same data. The higher no of trees in the random forest will give higher no of accuracy, so in random forest we can go for multiple trees. It can handle large no of independent variables without variable deletion and it will give the estimates that what variables are important. The RMSE value and R^2 value for our project in R and Python are –

|  |  |  |
| --- | --- | --- |
| **Random Forest** | **R** | **PYTHON** |
| **RMSE Train** | 0.2390 | 0.1891 |
| **RMSE Test** | 0.4762 | 0.3853 |
| **R^2 Train** | 0.8768 | 0.9742 |
| **R^2 Test** | 0.4940 | 0.6760 |

**Visualization for test and predicted test in Random Forest-**



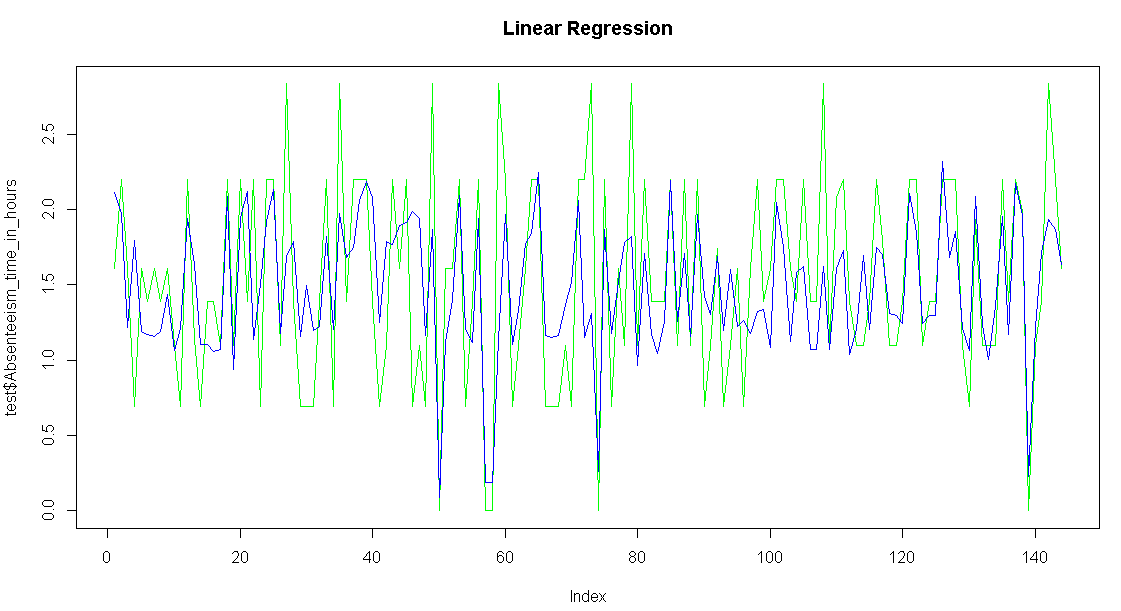
###### **2.2.3 Linear Regression**

Linear Regression is one of the statistical method of prediction. It is most common predictive analysis algorithm. It uses only for regression, means if the target variable is continuous than we can use linear regression machine learning algorithm.

The RMSE value and R^2 value for our project in R and Python are –

|  |  |  |
| --- | --- | --- |
| **Linear Regression** | **R** | **PYTHON** |
| **RMSE Train** | 0.4061 | 0.4190 |
| **RMSE Test** | 0.4922 | 0.4200 |
| **R^2 Train** | 0.6038 | 0.5877 |
| **R^2 Test** | 0.4639 | 0.6148 |

**Visualization for test and predicted test in Linear Regression-**

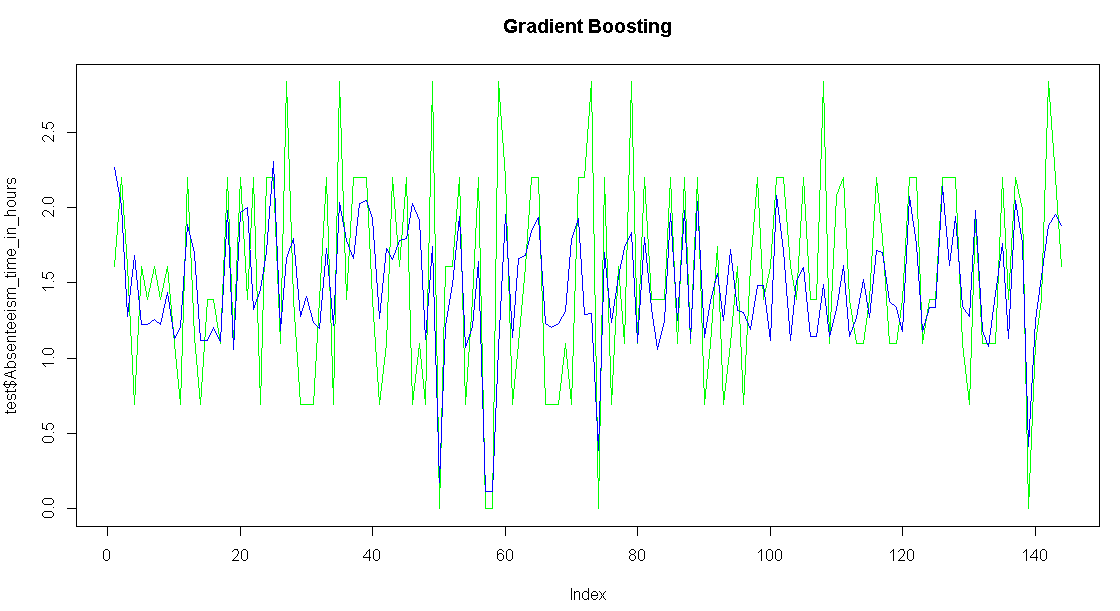


###### **2.2.4 Gradient Boosting**

Gradient boosting is a machine learning technique for regression and classification problems, It produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. The RMSE value and R^2 value for our project in R and Python are –

|  |  |  |
| --- | --- | --- |
| **Gradient Boosting** | **R** | **PYTHON** |
| **RMSE Train** | 0.4081 | 0.3602 |
| **RMSE Test** | 0.5031 | 0.3990 |
| **R^2 Train** | 0.7113 | 0.6954 |
| **R^2 Test** | 0.4405 | 0.6553 |

**Visualization for test and predicted test in Gradient boosting-**



# 

# **3. Conclusion**

In methodology we have done data cleaning and then applied different machine learning algorithms on the data set to check the performance of each model, now in conclusion we will finalize the model of Employee Absenteeism dataset.

3.1 Model Evaluation

In the previous chapter we have applied four algorithms on our dataset and calculate the

Root Mean Square Error (RMSE) and R-Squared Value for all the models. Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are, RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. RMSE is an absolute measure of 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. R-squared is a relative measure of fit. R-squared is basically explains the degree to which input variable explain the variation of the output. In simple words R-squared tells how much variance of dependent variable explained by the independent variable. It is a measure if goodness of fit in regression line. Value of R-squared between 0-1, where 0 means independent variable unable to explain the target variable and 1 means target variable is completely explained by the independent variable. So, Lower values of RMSE and higher value of R-Squared Value indicate better fit of model.

**3.2 Model Selection**

From the observation of all **RMSE Value** and **R-Squared** Value we have concluded that **Random Forest** has minimum value of RMSE (**0.3853)** and it’s **R-Squared** Value is also maximum (**0.67**).Means, By Random forest algorithm predictor are explain 67% to the target variable on the test data. The RMSE value of Test data and Train does not differs a lot this implies that it is not the case of overfitting.

## 

## **3.3 Answers of asked questions**

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

For the above query, we analyzed the data properly and plot some visualization to find the answers for this question.

i) When we studied the reason for absence we observed that absenteeism is frequently for following diseases Diseases of the genitourinary system, Pregnancy, childbirth and the puerperium, Injury, poisoning and certain other consequences of external causes

and External causes of morbidity and mortality. Company can take precaution for avoid these diseases.

ii) We found that the maximum tends of absenteeism people as education wise are those who have only high school degree.

iii) we also observed that that people with no children or no pets tend to be absent more than people who have children or pets, it shows that these people are far from their home town and they are unmarried.

iv)We also observe from visualization that the people who are social drinker tend to be absent more as compare to nondrinker.

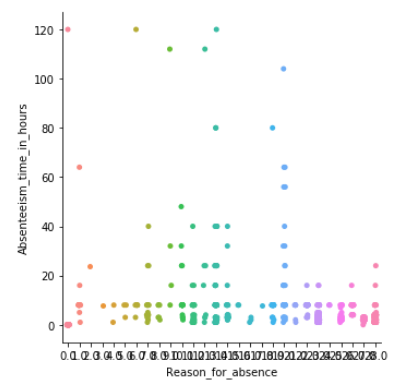
v) We noticed that People with disciplinary failure have maximum absenteeism as compare to non-disciplinary failure.

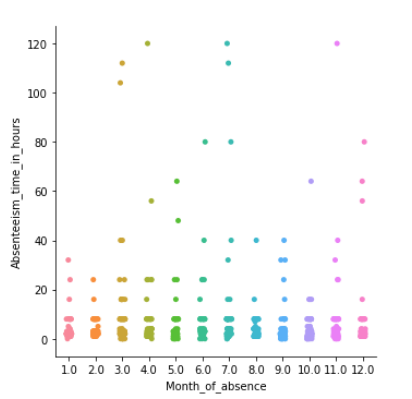
vi) From the predictors also observed that Employee who have work load between 240 to 300 minutes and age below 30 and service time is below 8, these employees tend to absent more frequently.

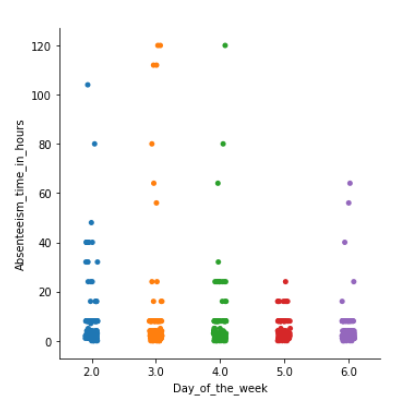
vii)There is no such impact of day of week or month or season because absenteeism trend is same for all these variables.

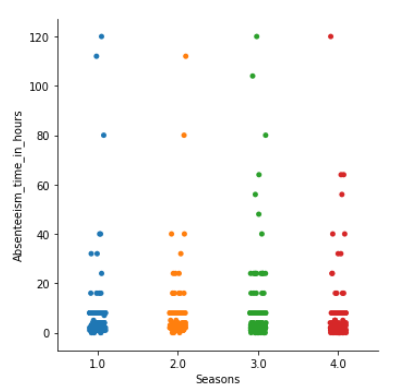
These are the main reason for absenteeism, company should focus on these reasons and find out appropriate solution for this.

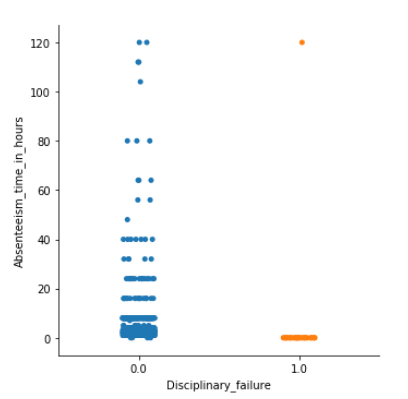
Here below are the visualizations.

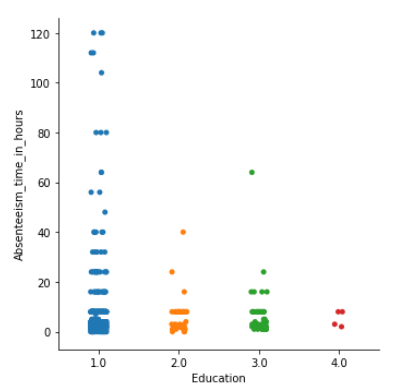


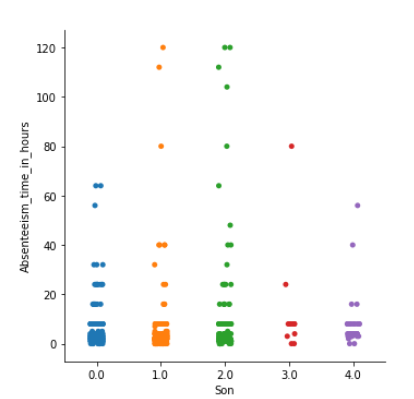


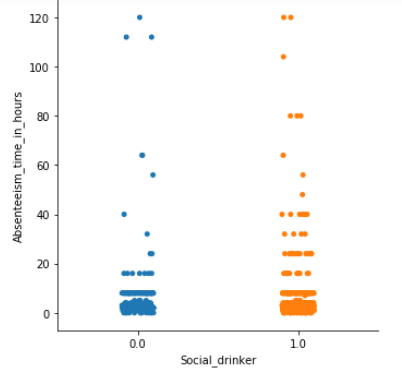


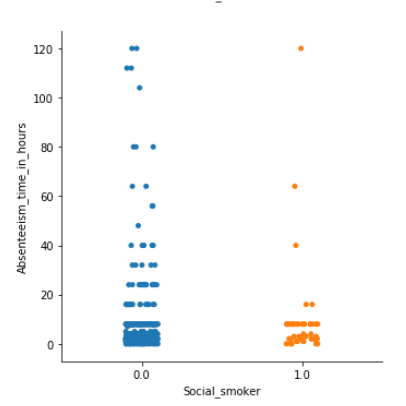


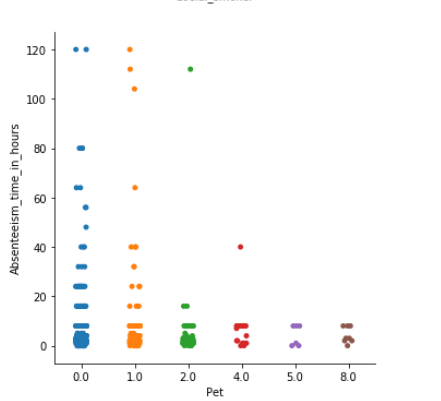


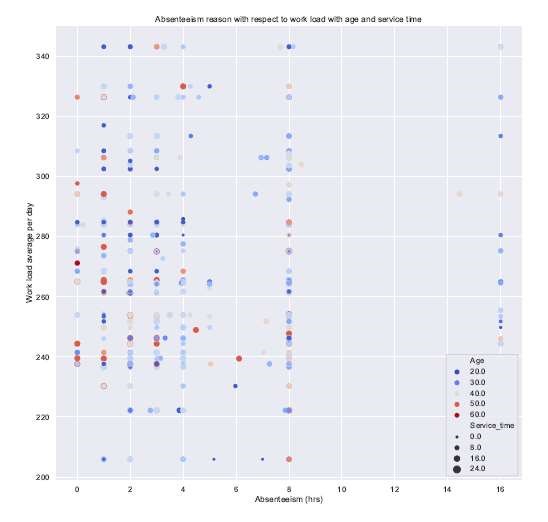












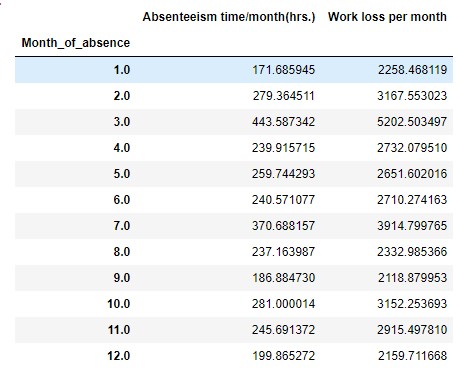
1. **How much losses every month can we project in 2011 if same trend of absenteeism continues?**

As we don’t have the data for 2011, we shall use the given data to calculate the loss in 2011 assuming the trend remains same. For this, we shall calculate the loss of work in time (hrs) which would be the total sum of absenteeism time in hours for each month respectively. We shall also calculate the loss in work load. Assuming work load average per day is the target workload for that day we shall calculate its loss due to absenteeism time in hours by the formula given below.

**Work loss = workload per day\* Absenteeism\_in\_hours**

**24**

By using above formula we can calculate monthly work loss in time (hrs.). Here, in below index we have calculated the work loss monthly by assuming that the trend will be same for the year 2011.



**4.Coding**

**4.1 R coding:**

#Remove all existing environment

rm(list=ls(all=T))

#set the working directory

setwd("C:/Data science/Project/Employee absenteeism")

#check the working directory

getwd()

#load libraries

X=c("ggplot2","corrgram","DMwR","caret","randomForest","unbalanced","C50","dummies","MASS","rpart",

"gbm","ROSE","e1071","Information","sampling","DataCombine","inTrees","readxl")

#Installing packages

install.packages(c("randomForest","unbalanced","c50","dummies","MASS","rpart","gbm","ROSE","e1071","Information","DataCombine"))

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

rm(X)

# loading Employee absenteeisam data set

Employee\_abs=read\_excel("Absenteeism\_at\_work\_Project.xls")

#####Explorataory Data Analysis##########

str(Employee\_abs)

head(Employee\_abs)

summary(Employee\_abs)

names(Employee\_abs)

#Replacing the space b/w collumn name to underscore for easy to use

names(Employee\_abs)=gsub(" ",'\_',names(Employee\_abs))

names(Employee\_abs)=gsub("/",'\_',names(Employee\_abs))

##univariate analysis and variable consolidation

Employee\_abs$Reason\_for\_absence=as.factor(as.character(Employee\_abs$Reason\_for\_absence))

Employee\_abs$Month\_of\_absence =as.factor(as.character(Employee\_abs$Month\_of\_absence))

Employee\_abs$Day\_of\_the\_week=as.factor(as.character(Employee\_abs$Day\_of\_the\_week))

Employee\_abs$Seasons=as.factor(as.character(Employee\_abs$Seasons))

Employee\_abs$Disciplinary\_failure=as.factor(as.character(Employee\_abs$Disciplinary\_failure))

Employee\_abs$Education=as.factor(as.character(Employee\_abs$Education))

Employee\_abs$Social\_drinker=as.factor(as.character(Employee\_abs$Social\_drinker))

Employee\_abs$Son=as.factor(as.character(Employee\_abs$Son))

Employee\_abs$Social\_smoker=as.factor(as.character(Employee\_abs$Social\_smoker))

Employee\_abs$Pet = as.factor(as.character(Employee\_abs$Pet))

str(Employee\_abs)

#from the data summary we can see there is variable ID.will remove it which is not usefull

Employee\_abs=subset(Employee\_abs,select=c(-ID))

#unique value of each count

apply(Employee\_abs, 2,function(x) length(table(x)))

#Since month variable can contain 12 values here we are replacing 0 with NA

Employee\_abs$`Month\_of\_absence`[Employee\_abs$`Month\_of\_absence` %in% 0]=NA

#Divide work load avarege/day variable by 1000(as per support team advise )

Employee\_abs$`Work\_load\_Average\_day`=Employee\_abs$`Work\_load\_Average\_day`/1000

#Extract column names of numeric and categorical variables

Numeric\_cnames = 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')

cat\_cnames = c('Reason\_for\_absence','Month\_of\_absence','Day\_of\_the\_week',

'Seasons','Disciplinary\_failure', 'Education', 'Social\_drinker',

'Social\_smoker', 'Son', 'Pet')

################Data Pre processing#################

#######Missing value analysis###########

missing\_val=data.frame(apply(Employee\_abs,2,function(x){sum(is.na(x))}))

#we found missing values in target variable

#removing obervations in which target variable having missing values

Employee\_abs=Employee\_abs[(!Employee\_abs$`Absenteeism\_time\_in\_hours` %in% NA),]

#remaining missing values

missing\_val=data.frame(apply(Employee\_abs,2,function(x){sum(is.na(x))}))

missing\_val$colnames=row.names(missing\_val)

names(missing\_val)[1]="Missing\_percentage"

missing\_val$`Missing\_percentage`=(missing\_val$`Missing\_percentage`/nrow(Employee\_abs))\*100

missing\_val=missing\_val[order(-missing\_val$`Missing\_percentage`),]

row.names(missing\_val) = NULL

missing\_val = missing\_val[,c(2,1)]

write.csv(missing\_val, "Miising\_perc.csv", row.names = F)

##Visualizing missing value percentage########

ggplot(data = missing\_val[1:3,], aes(x=reorder(colnames, -Missing\_percentage),y = Missing\_percentage))+

geom\_bar(stat = "identity",fill = "grey")+xlab("Parameter")+

ggtitle("Missing data percentage") + theme\_bw()

df=Employee\_abs

#Employee\_abs=df

sum(is.na(Employee\_abs))

#Missing value imputation for categorical variables#

#Mode method-

mode <- function (x, na.rm) {

xtab <- table(x)

xmode <- names(which(xtab == max(xtab)))

if (length(xmode) > 1) xmode <- ">1 mode"

return(xmode)

}

for(i in cat\_cnames){

print(i)

Employee\_abs[,i][is.na(Employee\_abs[,i])] = mode(Employee\_abs[,i])

}

str(Employee\_abs)

#missing values imputation for numeric variables

#lets take one sample variable

Employee\_abs$`Body\_mass\_index`[6]

#actual value=27

#mean method=26.70

#median method=25

#KNN=27

Employee\_abs$`Body\_mass\_index`[6]=NA

Employee\_abs$`Body\_mass\_index`[6]

#mean method

#Employee\_abs$`Body\_mass\_index`[is.na(Employee\_abs$`Body\_mass\_index`)]=mean(Employee\_abs$`Body\_mass\_index`,na.rm = T)

#median method

#Employee\_abs$`Body\_mass\_index`[is.na(Employee\_abs$`Body\_mass\_index`)]=median(Employee\_abs$`Body\_mass\_index`,na.rm = T)

#KNN Imputation # reload the data and perform Knn

library(VIM)

Employee\_abs=kNN(Employee\_abs,variable=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'), k=5)

#In KNN imputaion we get some extra variables (logical) so we delete those junk variable below

Employee\_abs=subset(Employee\_abs,select=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','Reason\_for\_absence','Month\_of\_absence','Day\_of\_the\_week',

'Seasons','Disciplinary\_failure', 'Education', 'Social\_drinker',

'Social\_smoker', 'Son', 'Pet'))

#from the all above methods we can see knn is more accurate so we are imputed with knn method

sum(is.na(Employee\_abs))

#now data is free from missing values

#######outlier Analysis #######

#save data for the reference

df2=Employee\_abs

#Employee\_abs=df

#Box plots-Distributaion and outlier check

Numeric\_index=sapply(Employee\_abs,is.numeric)#selecting only numeric

Numeric\_data=Employee\_abs[,Numeric\_index]

cnames=colnames(Numeric\_data)

cnames

for(i in 1:length(cnames)){

assign(paste0("AB",i),ggplot(aes\_string(y=(cnames[i]),x="Absenteeism\_time\_in\_hours"),

d=subset(Employee\_abs))

+geom\_boxplot(outlier.colour = "Red",outlier.shape = 18,outlier.size = 2,

fill="skyblue4")+theme\_gray()

+stat\_boxplot(geom = "errorbar", width=0.5)

+labs(y=cnames[i],x="Absenteeism\_time\_in\_hours")

+ggtitle("Box Plot of Absenteeism for",cnames[i]))

}

#ploting plots together

gridExtra::grid.arrange(AB1,AB2,ncol=2)

gridExtra::grid.arrange(AB3,AB4,ncol=2)

gridExtra::grid.arrange(AB5,AB6,ncol=2)

gridExtra::grid.arrange(AB7,AB8,ncol=2)

gridExtra::grid.arrange(AB9,AB10,ncol=2)

#remove Outliers using boxplotmethod

#Loop to remove from all variables

#for(i in cnames){

# print(i)

# val=Employee\_abs[,i][Employee\_abs[,i]%in% boxplot.stats(Employee\_abs[,i])$out]

# Employee\_abs=Employee\_abs[which(!Employee\_abs[,i] %in% val),]

#}

#replace all outliers with NA and Impute

for(i in cnames){

print(i)

val=Employee\_abs[,i][Employee\_abs[,i]%in% boxplot.stats(Employee\_abs[,i])$out]

Employee\_abs[,i][Employee\_abs[,i] %in% val]=NA

}

sum(is.na(Employee\_abs))

Employee\_abs=knnImputation(Employee\_abs,k=3)

df1=Employee\_abs

#Employee\_abs=df1

#here the data is free from outliers.

########Feature Selection#########

#Correlation Analysis for continuous variables-

library(corrgram) #Library for correlation plot

corrgram(Employee\_abs[,cnames],order=FALSE,upper.panel = panel.pie,

text.panel = panel.txt,font.labels =1,

main="Correlation plot for Absenteeism")

#Correlated variables are = weight & Body mass index.

#Anova Test for categorical variable-

for(i in cat\_cnames){

print(i)

Anova\_result= summary(aov(formula = Absenteeism\_time\_in\_hours~Employee\_abs[,i],Employee\_abs))

print(Anova\_result)

}

#redudant categorical variables- Social\_smoker,Education,Seasons,Day\_of\_the\_week

#Dimensionity Reduction

Employee\_abs= subset(Employee\_abs,select=-c(Weight,Social\_smoker,Education,Seasons,Day\_of\_the\_week))

dim(Employee\_abs)

#######Feature Scaling#########33

df3=Employee\_abs

#Employee\_abs=df3

# update the numeric and categorical variables

cnames=c('Distance\_from\_Residence\_to\_Work', 'Service\_time', 'Age',

'Work\_load\_Average\_day', 'Transportation\_expense',

'Hit\_target', 'Height',

'Body\_mass\_index', 'Absenteeism\_time\_in\_hours')

cat\_cnames=c('Reason\_for\_absence','Month\_of\_absence','Disciplinary\_failure', 'Social\_drinker', 'Son', 'Pet')

#summary of data to check min and max values of numeric variables-

summary(Employee\_abs)

#Skewness of numeric variables-

library(propagate)

for(i in cnames){

skew = skewness(Employee\_abs[,i])

print(i)

print(skew)

}

#log transform

Employee\_abs$Absenteeism\_time\_in\_hours = log1p(Employee\_abs$Absenteeism\_time\_in\_hours)

#Normality check

qqnorm(Employee\_abs$Transportation\_expense)

hist(Employee\_abs$Absenteeism\_time\_in\_hours,col="blue",main="Histogram of Absenteeism ")

hist(Employee\_abs$Transportation\_expense)

hist(Employee\_abs$Work\_load\_Average\_day)

#from above histogram plots we can say that data is not normally distributed

#so best method is normalization

#Normalization

for (i in cnames) {

if(i != "Absenteeism\_time\_in\_hours")

{

print(i)

Employee\_abs[,i]=(Employee\_abs[,i]-min(Employee\_abs[,i]))/(max(Employee\_abs[,i])-min(Employee\_abs[,i]))

}

}

#Summary of data after all preprocessing-

summary(Employee\_abs)

write.csv(Employee\_abs,"Absenteeism\_Pre\_processed\_Data.csv",row.names=FALSE)

########Model Development##########

#clean the environment

library("DataCombine")

rmExcept("Employee\_abs")

#save the data for reference

df4=Employee\_abs

cat\_cnames=c('Reason\_for\_absence','Month\_of\_absence','Disciplinary\_failure', 'Social\_drinker', 'Son', 'Pet')

#create dummy variable for categorical variables-

library(dummies)

Employee\_abs = dummy.data.frame(Employee\_abs, cat\_cnames)

dim(Employee\_abs)

#Divide the data into train and test-

set.seed(4567)

train\_index= sample(1:nrow(Employee\_abs),0.8\*nrow(Employee\_abs))

train= Employee\_abs[train\_index,]

test= Employee\_abs[-train\_index,]

#########Decision Tree Regression#######

#Model devlopment for train data

library(rpart) #Library for regression model

DT\_model= rpart(Absenteeism\_time\_in\_hours~.,train,method="anova")

DT\_model

#Prediction for train data-

DT\_train=predict(DT\_model,train[-9])

#Prediction for test data-

DT\_test=predict(DT\_model,test[-9])

#Error metrics to calculate the performance of model-

rmse= function(y,y1){

sqrt(mean(abs(y-y1)^2))

}

rmse(train[,9],DT\_train)

#rmse=0.4369

#RMSE calculation for test data-

rmse(test[,9],DT\_test)

#RMSE\_test= 0.5030

#r-square calculation-

#function for r-square-

rsquare=function(y,y1){

cor(y,y1)^2

}

#r-square calculation for train data-

rsquare(train[,9],DT\_train)

#r-square\_train= 0.54

#r-square calculation for test data-

rsquare(test[,9],DT\_test)

#r-square\_test= 0.4366973

#Visulaization to check the model performance on test data-

plot(test$Absenteeism\_time\_in\_hours,type="l",lty=1.8,col="Green",main="Decision Tree")

lines(DT\_test,type="l",col="Blue")

#Write rule into drive-

write(capture.output(summary(DT\_model)),"Decision\_Tree\_Model.txt")

#############Random Forest##########

library(randomForest) #Library for randomforest machine learning algorithm

library(inTrees) #Library for intree transformation

RF\_model= randomForest(Absenteeism\_time\_in\_hours~.,train,ntree=300,method="anova")

#transform ranfomforest object to inTree format-

treelist= RF2List(RF\_model)

#Extract rules-

rules= extractRules(treelist,train[-9])

rules[1:5,]

#Make rules into redable format-

readable\_rules= presentRules(rules,colnames(train))

readable\_rules[1:5,]

#Get Rule metrics-

rule\_metrics= getRuleMetric(rules,train[-9],train$Absenteeism\_time\_in\_hours)

rule\_metrics= presentRules(rule\_metrics,colnames(train))

rule\_metrics[1:10,]

summary(rule\_metrics)

#Check model performance on train data

RF\_train= predict(RF\_model,train[-9])

RF\_train

#Check model performance on test data

RF\_test= predict(RF\_model,test[-9])

RF\_test

#RMSE calculation for train data-

rmse(train[,9],RF\_train)

#RMSE\_train= 0.2390

#RMSE calculation for test data-

rmse(test[,9],RF\_test)

#RMSE\_test= 0.4762

#r-square calculation for train data-

rsquare(train[,9],RF\_train)

#r-square= 0.8768

#r-square calculation for test data-

rsquare(test[,9],RF\_test)

#r-square= 0.4940

#Visulaization to check the model performance on test data-

plot(test$Absenteeism\_time\_in\_hours,type="l",lty=1.8,col="Green",main="Random Forest")

lines(RF\_test,type="l",col="Blue")

#write rule into drive-

write(capture.output(summary(rule\_metrics)),"Random\_Forest\_Model.txt")

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

#recall numeric variables to check the VIF-

numeric\_index1= c("Transportation\_expense","Distance\_from\_Residence\_to\_Work","Service\_time",

"Age","Work\_load\_Average\_day","Hit\_target","Height",

"Body\_mass\_index","Absenteeism\_time\_in\_hours")

numeric\_data1= Employee\_abs[,numeric\_index1]

cnames1= colnames(numeric\_data1)

cnames1

library(usdm) #Library for VIF(Variance Infleation factor)

vif(numeric\_data1)

vifcor(numeric\_data1,th=0.7) #VIF calculation for numeric variables

#Linear regression model-

lr\_model= lm(Absenteeism\_time\_in\_hours~.,train)

summary(lr\_model)

#check model performance on train data-

lr\_train= predict(lr\_model,train[-9])

#check model performance on test data-

lr\_test= predict(lr\_model,test[-9])

#RMSE calculation for train data-

rmse(train[,9],lr\_train)

#RMSE\_train=0.4061

#RMSE calculation for test data-

rmse(test[,9],lr\_test)

#RMSE\_test=0.4922

#r-square calculation for train data-

rsquare(train[,9],lr\_train)

#r-square\_train=0.6038

#r-square calculation for test data-

rsquare(test[,9],lr\_test)

#r-square\_test=0.4639

#Visulaization to check the model performance on test data-

plot(test$Absenteeism\_time\_in\_hours,type="l",lty=1.8,col="Green",main="Linear Regression")

lines(lr\_test,type="l",col="Blue")

write(capture.output(summary(lr\_model)),"Linear\_Regression\_Model.txt")

############Gradient Boosting########

library(gbm)

#Develop Model

GB\_model = gbm(Absenteeism\_time\_in\_hours~., data = train, n.trees = 100, interaction.depth = 2)

#check model performance on train data-

GB\_train = predict(GB\_model, train,n.trees = 100)

#check model performance on test data-

GB\_test = predict(GB\_model, test, n.trees = 100)

#RMSE calculation for train data-

rmse(train[,9],GB\_train)

#RMSE\_train=0.408

#RMSE calculation for test data-

rmse(test[,9],GB\_test)

#RMSE\_test=0.5035

#r-square calculation for train data-

rsquare(train[,9],GB\_train)

#r-square\_train=0.7113

#r-square calculation for test data-

rsquare(test[,9],GB\_test)

#r-square\_test=0.4405

#Visulaization to check the model performance on test data-

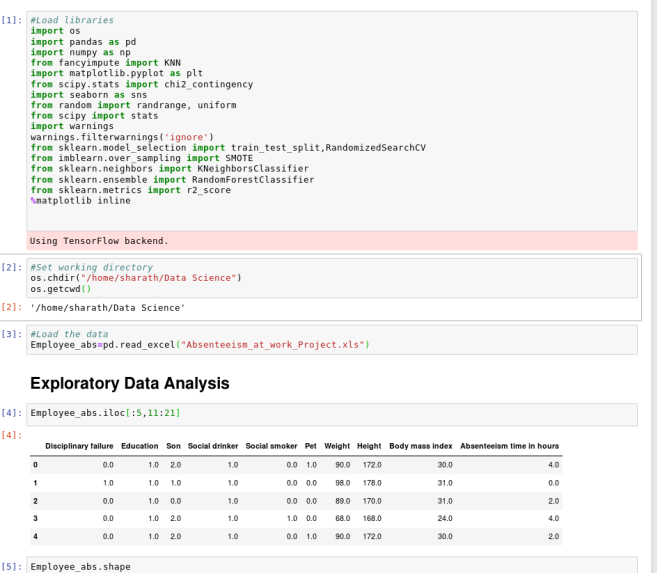
plot(test$Absenteeism\_time\_in\_hours,type="l",lty=1.8,col="Green",

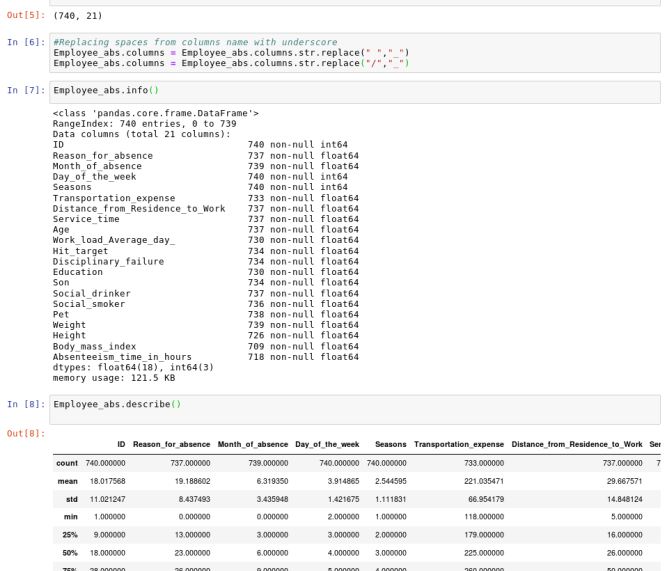
main="Gradient Boosting")

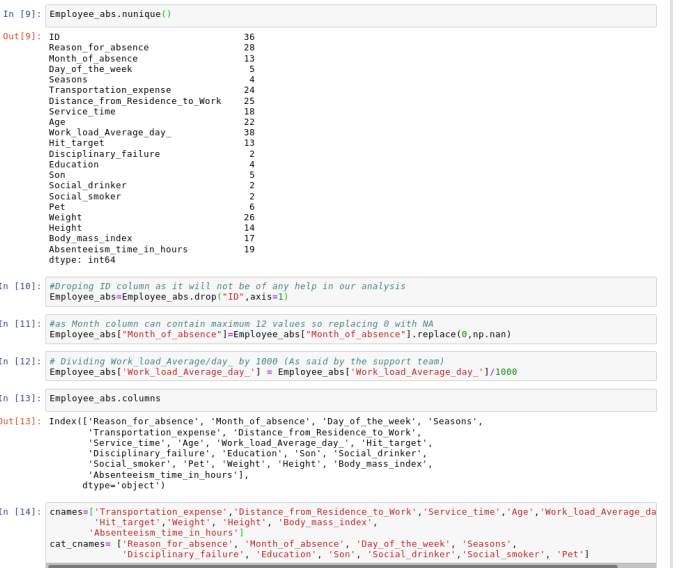
lines(GB\_test,type="l",col="Blue")

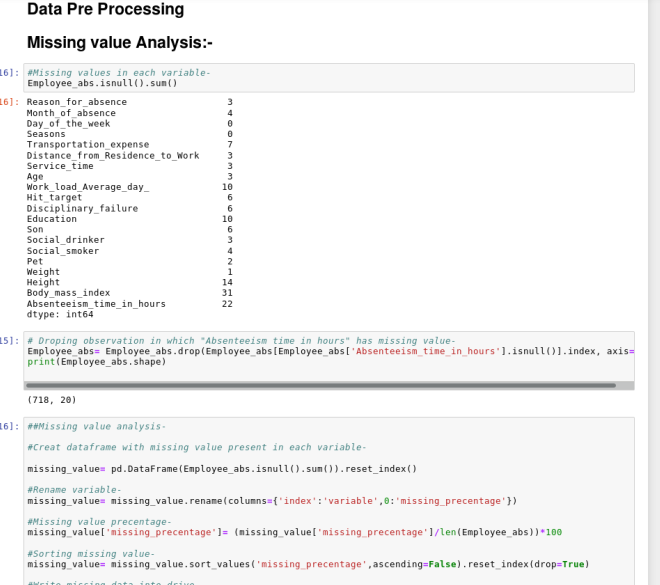
##############Thank you##########

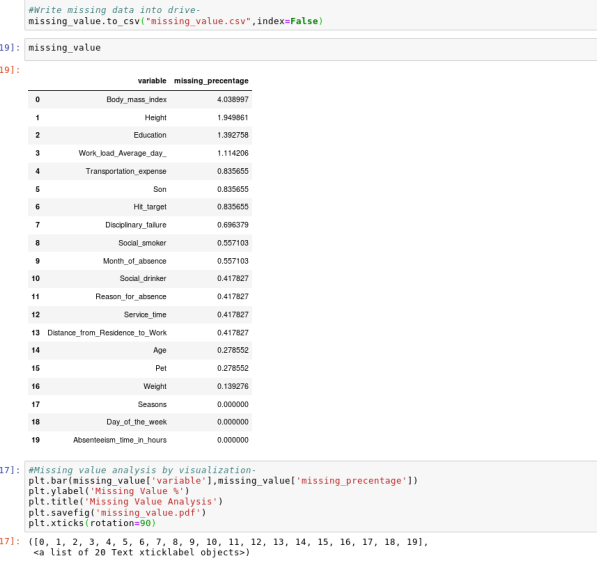
**4.2 Python Coding:**

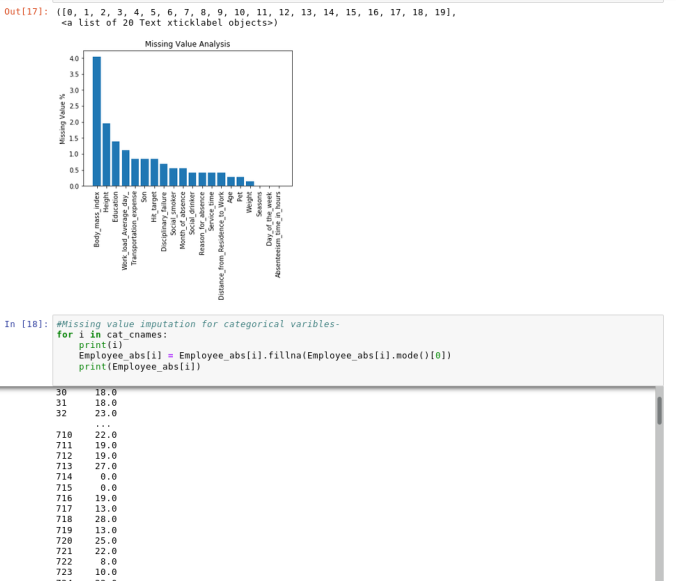


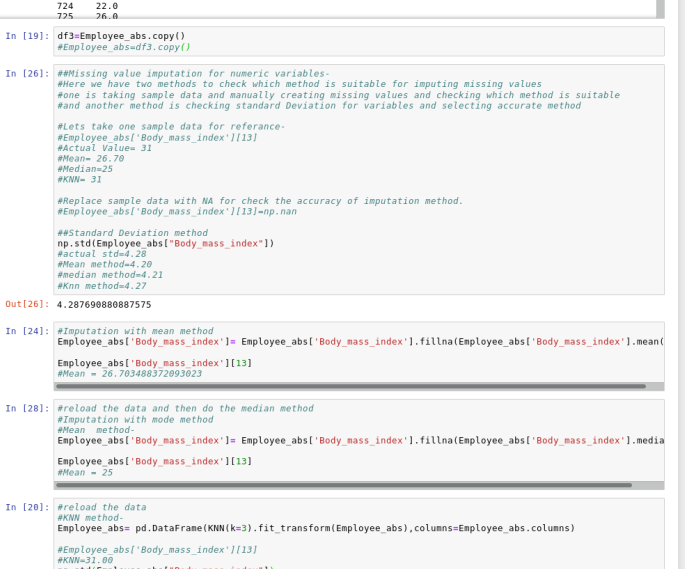


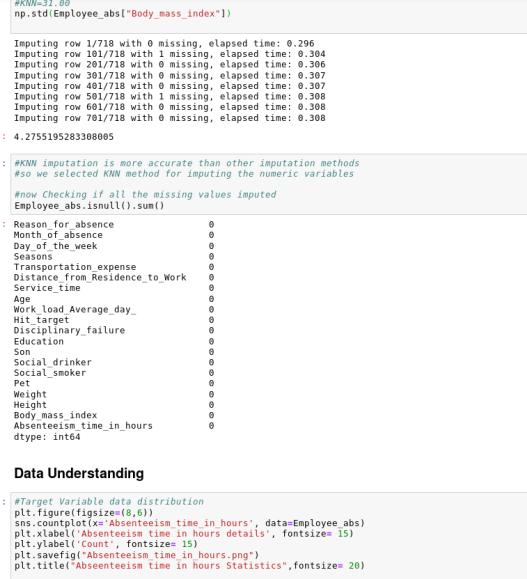




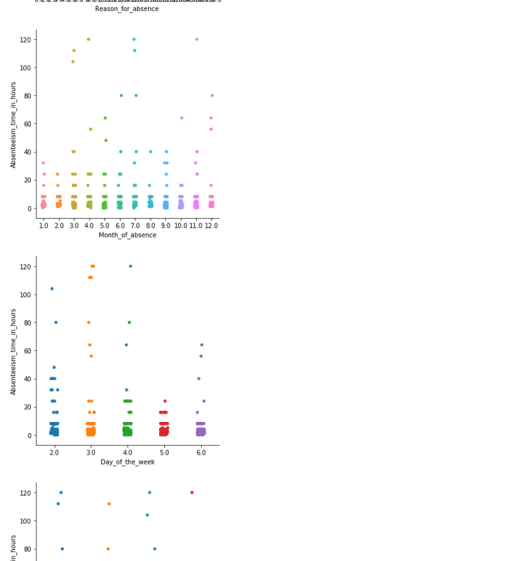


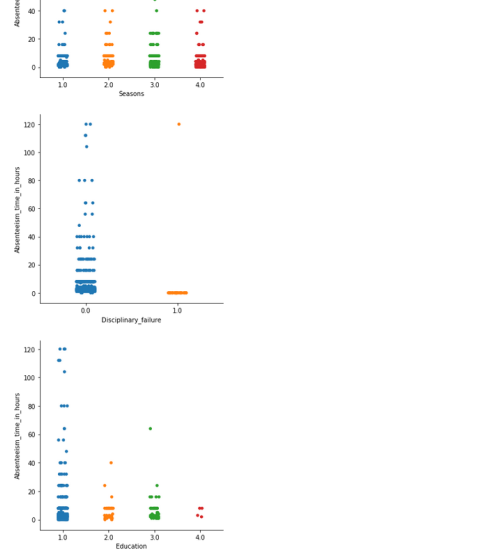




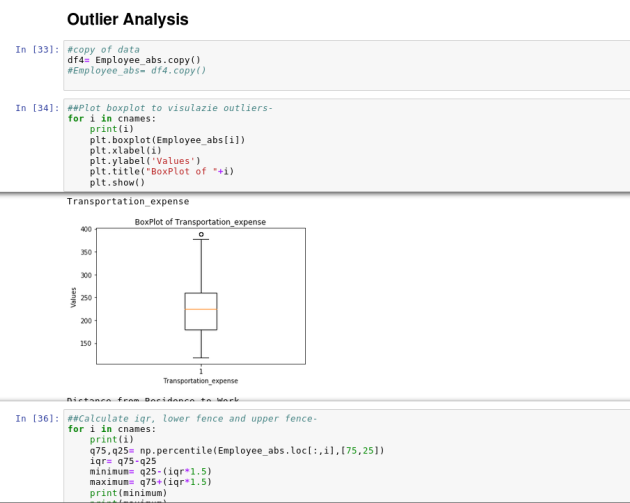


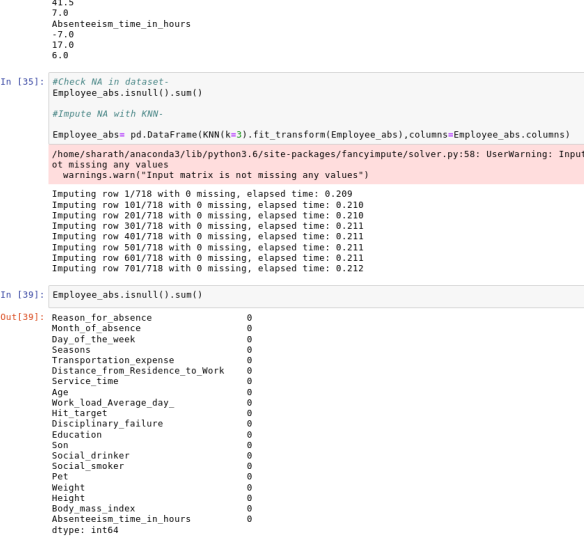


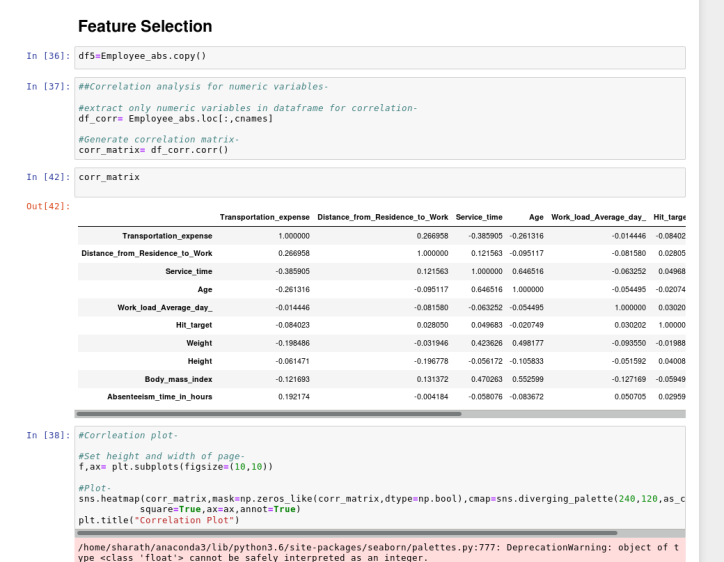


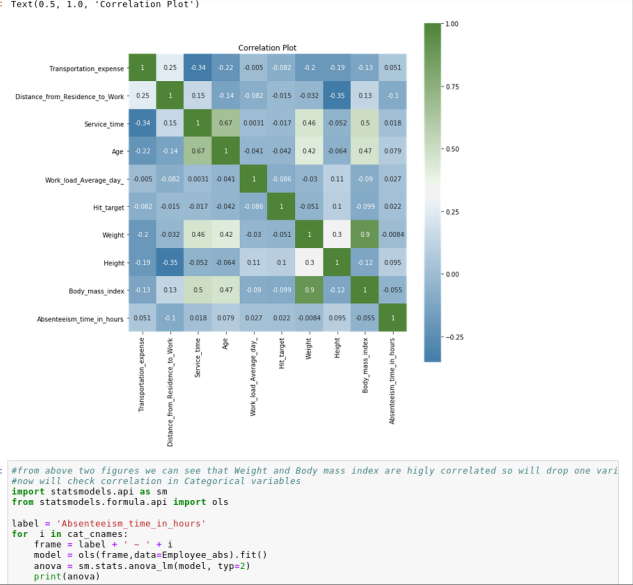


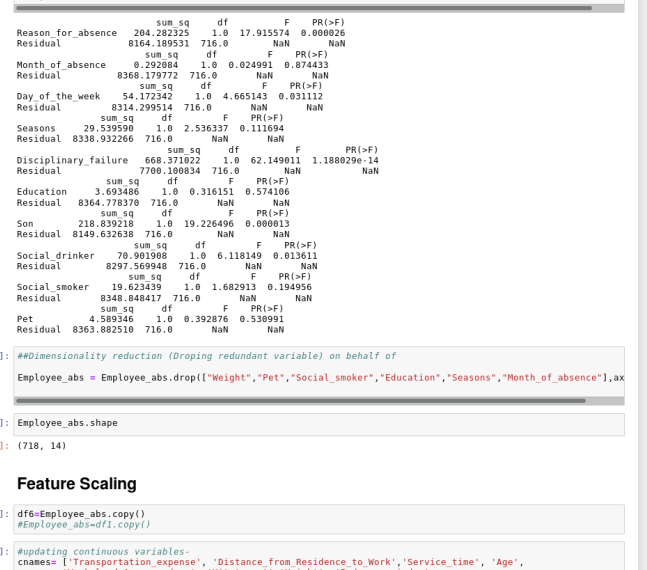




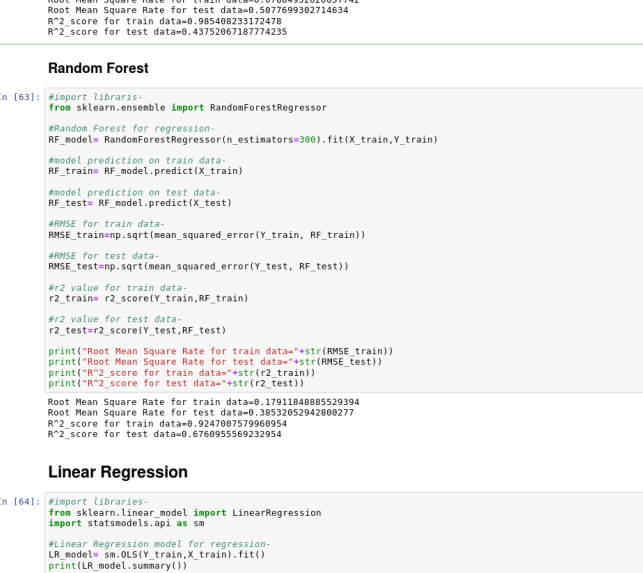


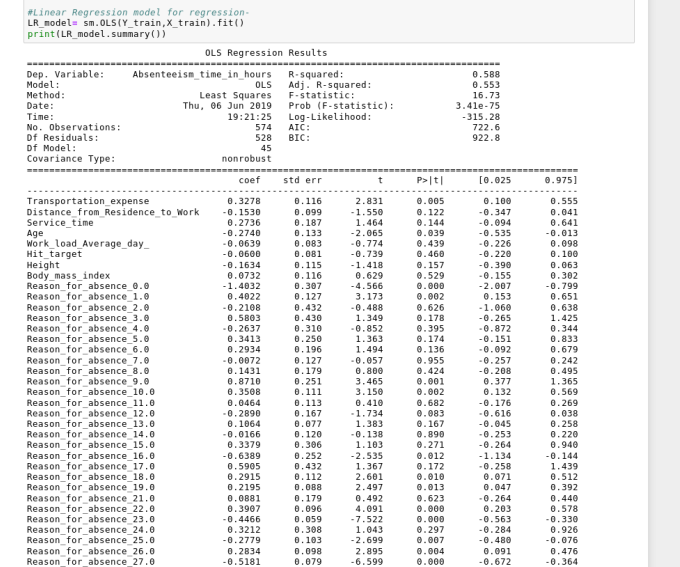


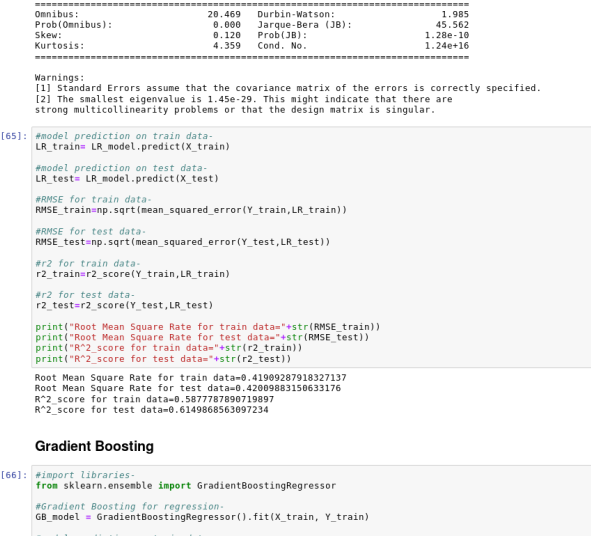


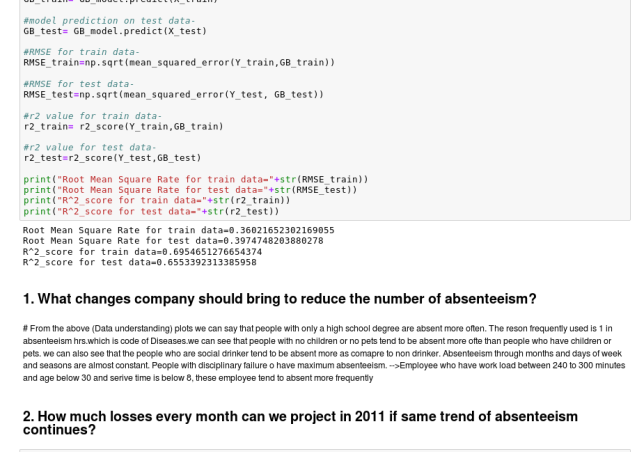




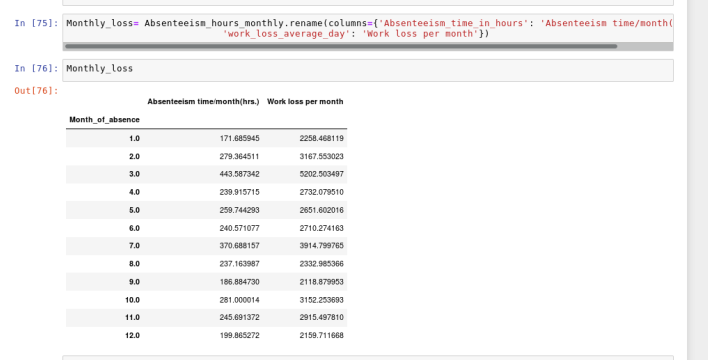












**Reference:**

<https://learning.edwisor.com/>

<https://matplotlib.org/index.html>

<https://stackoverflow.com/>

<https://towardsdatascience.com/>