***Project Report***

***Employee Absenteeism***

***By***

***Sonnet Padayatty Reju***

***14th August 2018***

***Contents***

1 Introduction ………………………………………………..………………… 3

1.1 Project Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.2 Problem Statement. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2 Methodology ……………………………………………………….…………. 4

2.1 Pre Processing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . …. 4

2.1.1 Missing Value Analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . …….4

2.1.2 Outlier Analysis…….. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6

2.1.3 Feature selection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..9

2.1.4 Feature scaling ……. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . 12

3 Modeling ……………….. . . . . . . . . . . . . . . . . . . . . . . . . . ……….. . . . . . . 14

3.1 Model Selection . . . . . . . . . . . . . . . . . . . . . . . . . . . ………. . . . . . . . . . . . 15

3.2 Model evaluation . . . . . . . . . …... . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

Appendix A – Summary of linear regression …………………………… ……..19

Appendix B - Extra Figures of Box plot and normality check(without outliers) 20

Appendix C – R code & Python code (complete)……………………………….28

**Chapter 1**

**Introduction**

Absenteeism can be seen in almost all companies. It is the absence of employees from the work without any proper reason. This trend can adversely affect both Company and Employees. Absenteeism may affect the overall performance of an employee, which eventually could affect the productivity of the company as well.

* 1. Project 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 the 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 loss every month can we expect in 2011 if same trend continues?

* 1. Problem Description

As the target variable in this case is continuous in nature, we can implement **Regression** **Machine learning algorithms** like Decision tree, Random Forest and Linear Regression for the predictive analysis. The independent variables included in the given dataset are the followings:-

**Attribute Information**:

1. Individual identification (ID)

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

I. Certain infectious and parasitic diseases

II. Neoplasms

III. Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism

IV. Endocrine, nutritional and metabolic diseases

V. Mental and 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 (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)

**Chapter 2**

**Methodology**

2.1 Pre-Processing

Pre-processing of data is an indispensable stage in predictive analysis. Since the predictive model needs to handle a big data set, it is always necessary to eliminate unwanted data. There may be many variables whose data type is incorrect and may create complexity while training the predictive model with train dataset. In order to minimize such issues in modeling stage, we conduct data pre-processing and extract important insights from the raw data. We can extract such information by analyzing the independent variables using probability density function or by visualizing how the data points have been distributed in each variable. It can be easily achieved by checking the normality using Tableau or other normality checking functions.

Following are main pre-processing methods used in predictive analysis

**2.1.1 Missing Value Analysis**

Once we are done with pre-processing steps like “Renaming the variables” and “Converting into proper Data types”, we can conduct the missing value analysis. You may use the combination of both train and test data for the imputation of missing values as it makes the model to predict the values more accurately. Mainly, there are 3 methods for imputation of missing values.

1. Mean method
2. Median method
3. KNN imputation

We are not supposed to do imputation if the percentage of missing values in a variable is more than 30%. The below table shows the percentage of missing values in the dataset.

|  |  |
| --- | --- |
| **Variables** | **Missing\_percentage** |
| Body mass index | 4.189189189 |
| Absenteeism time in hours | 2.972972973 |
| Height | 1.891891892 |
| Work load Average/day | 1.351351351 |
| Education | 1.351351351 |
| Transportation expense | 0.945945946 |
| Son | 0.810810811 |
| Disciplinary failure | 0.810810811 |
| Hit target | 0.810810811 |
| Social smoker | 0.540540541 |
| Age | 0.405405405 |
| Reason for absence | 0.405405405 |
| Service time | 0.405405405 |
| Distance from Residence to Work | 0.405405405 |
| Social drinker | 0.405405405 |
| Pet | 0.27027027 |
| Weight | 0.135135135 |
| Month of absence | 0.135135135 |
| Seasons | 0 |
| Day of the week | 0 |
| ID | 0 |

Since the missing value percentages are not greater than 30%, we can perform imputation steps using KNN, median and mean method. Here, median and mean methods bring more accurate values. Hence we are freezing the same for further imputation.

2.1.2 Outlier Analysis

By definition, outliers are the points that are distant from remaining observations. As a result, they can potentially skew or bias any analysis performed on the dataset. It is therefore very important to detect and adequately deal with outliers. In order to show the impact of outliers, we use a technique called box plot in which the distribution of data points is visualized. The box plots and histograms of each independent variable before and after outlier analysis are shown in the Appendix. Here, I would like to share the box plots and histograms of the variable “Height” in order to display the **effect of outliers** in each variable.

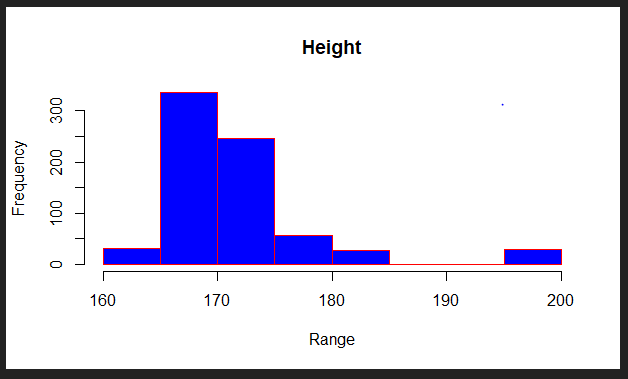
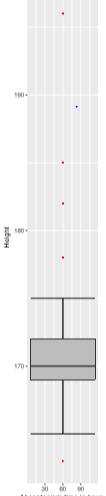
 

Fig:1 - Normality check and Box plot of variable –“Height” with Outliers

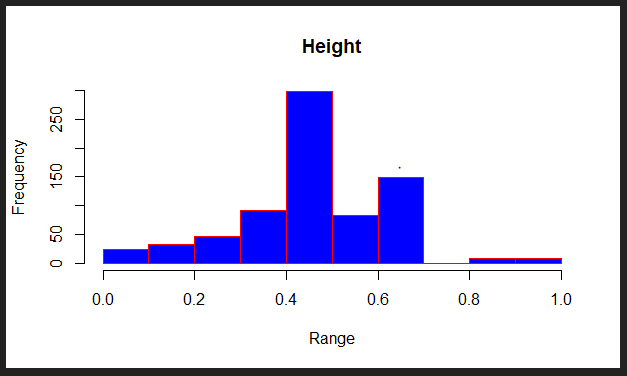
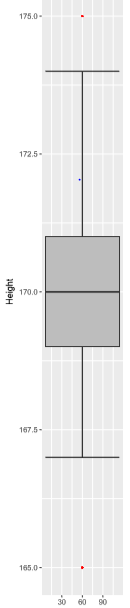
 

Fig: 2 - Normality check and Box plot of variable –“Height” without Outliers

From the above figures, it is evident that the distribution of data became more normal / less skewed after the removal of outliers. In the box plots of variables , we can see red spots- ‘Outliers’, which were more dominant in numbers before outlier analysis, has become very less and negligible after outlier analysis. This illustrates that removal of outliers from the data can improve the accuracy of the predictive model.

2.1.3 Feature Selection

Feature selection is extremely important in machine learning primarily because it serves as a fundamental technique to direct the use of variables to what's most efficient and effective for a given machine learning system. It helps to minimize the curse of dimensionality or help deal with over fitting feature selection helps to give [developers](https://www.techopedia.com/definition/17095/developer) the tools to use only the most relevant and useful data in machine learning [training sets](https://www.techopedia.com/definition/33181/training-data), which dramatically reduces costs and data volume. There are many ways to do feature selection, but in this project we use Anova test and Correlation Analysis for the feature selection of Categorical and Continuous variables respectively.

The correlation plot of numerical variables is shown below. From the figures, it is clear that the following variables are highly correlated to each other.

1. Age and Service time
2. Weight and Body mass index

As these variables are highly correlated, we can eliminate the following numerical variables, so that it helps to minimize the curse of dimensionality.

1. Age
2. Body Mass index

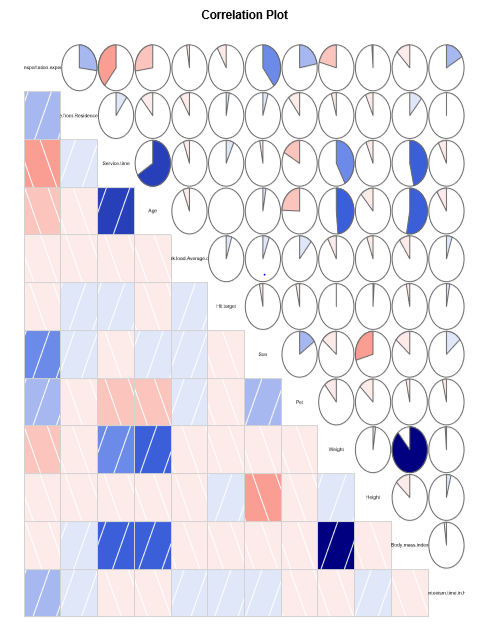


Fig : 7 – Correlation plot used in R & Python programming

When it comes to categorical variables, we need to perform Anova test, in order to extract unwanted variables from the data set. The result of Anova test is shown below.

> res.aov = aov(formula=Absenteeism.time.in.hours ~ ID, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

ID 35 12674 362.1 2.162 0.000149 \*\*\*

Residuals 704 117892 167.5

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Reason.for.absence, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Reason.for.absence 27 25224 934.2 6.314 <2e-16 \*\*\*

Residuals 712 105341 148.0

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Month.of.absence, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Month.of.absence 12 3253 271.1 1.548 0.102

Residuals 727 127313 175.1

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Day.of.the.week, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Day.of.the.week 4 2147 536.8 3.072 0.0159 \*

Residuals 735 128418 174.7

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Seasons, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Seasons 3 540 180.0 1.019 0.384

Residuals 736 130025 176.7

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Disciplinary.failure, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Disciplinary.failure 1 454 454.1 2.575 0.109

Residuals 738 130111 176.3

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Education, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Education 3 257 85.54 0.483 0.694

Residuals 736 130309 177.05

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Social.drinker, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Social.drinker 1 517 516.5 2.931 0.0873 .

Residuals 738 130049 176.2

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> res.aov = aov(formula=Absenteeism.time.in.hours ~ Social.smoker, data = absent)

> summary(res.aov)

Df Sum Sq Mean Sq F value Pr(>F)

Social.smoker 1 206 206.2 1.167 0.28

Residuals 738 130359 176.6

Since the p values of the variables-" Disciplinary.failure, Month.of.absence, Seasons ,Education, Social.drinker, Social.smoker " are greater than 0.05, As per the rule of Anova test, it is found that these variables can be eliminated from modeling stage. Therefore, as part of Dimension reduction, the following variables are removed from the data set.

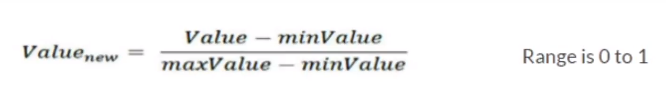
1. Disciplinary.failure
2. Month.of.absence
3. Seasons
4. Education
5. Social.drinker
6. Social.smoker
7. Age
8. Body mass index

2.1.4 Feature Scaling

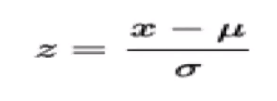
Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Since the range of values of raw data varies widely, in some machine learning 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. Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it.

In this project, we apply feature scaling method- Normalization to the independent variables. Standardization is usually done on the variables whose distribution of data is normal, whereas the Normalization is applied on the variables whose data distribution is not uniform/ Normal. Normality check functions are used for the same. Once we analyze the distribution of data/ Normality check, Standardization and Normalization is performed on the data set using following formulas.

Normalization Formula:-



Standardization Formula:-



Where x = raw value, σ = standard deviation, μ = mean

Z can be either positive or negative. It will be negative if the raw value is less than mean. On the other hand it will be positive when raw value is greater than mean.

**Chapter 3**

**Modeling**

3.1 Model Selection

Model selection is purely based on the type of Machine learning algorithm that we opt. In this case, we follow supervised machine learning. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

It is called supervised learning because the process of algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. Supervised learning problems can be further grouped into regression and classification problems.

* **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

As the problem statement is about Absenteeism and the target variable is Continuous, The predictive model should be a Regression model. We need to follow Trial and Error method in order to find out the most suitable predictive model. The Regression models that we used in this project are followings:-

* 1. Decision tree
  2. Random forest
  3. Linear regression

3.2 Model Evaluation

The evaluation of model can be done by Error metrics. There are two types of error metrics.

1. Classification metrics
2. Confusion matrix
3. Accuracy
4. False positive rate
5. False negative rate
6. Regression metrics
7. RMSE
8. MSE

**Accuracy % = 100-Error**

As we selected Regression model for Predictive Analysis, The error metrics that we opted for model evaluation is Regression metrics. The characteristics of dataset is Time series multivariate. Hence the error metric, used for the accuracy calculation of model, is RMSE(root mean square error).

It is a popular metric used to measure the error rate of time series or transition regression model. It can be only compared between models whose errors are measured in the same units. It can be calculated by squaring the errors, finding their average and taking their square root. It can be mathematically represented as:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Description: http://www.saedsayad.com/images/RMSE.png  Description: http://www.saedsayad.com/images/actual_predicted.png |  |  |

1. **Decision tree**

R code:- Python:-

Accuracy: 89% Accuracy: 89%

Error: 11% Error: 11%

1. **Random Forest**

R code:- Python:-

Accuracy: 90% Accuracy: 89%

Error: 10% Error: 11%

1. **Linear Regression**

R code:- Python:-

Accuracy: 90% Accuracy: 89%

Error: 10% Error: 11%

***From the above results, it is clear that the model based on Random Forest &***

***Linear Regression works better.***

***The summary of linear regression model is mention in Appendix A. Based on the number of stars beside p values of independent variables, we can decide the***

***Influential factors of absenteeism.***

**Conclusion**

From the summary output of linear regression(Appendix I), it is clear that the most influential coefficient / variables are **Reason of Absence,** **Distance.from.Residence.to.Work, Service.time** and the followings are the major reasons of Absenteeism among employees.

1. Diseases of the circulatory system
2. Diseases of the skin and subcutaneous tissue
3. Diseases of the nervous system
4. Diseases of the musculoskeletal system and connective tissue

**Through this study, it is found that most of the absenteeism is due to health issues, mainly diseases of circulatory system. So it is necessary to provide a health awareness campaign for the employees. If possible, The Company should appoint a Doctor who works 24\*7, in order to handle the critical situations. Absenteeism occurred due to the reason- “Distance from residence” can be resolved by implementing free cab service during late working hours or providing travel allowance.**

**If the absenteeism trend continuous, the estimated loss of each month in 2011 is given below.**

|  |  |
| --- | --- |
| ***Month*** | ***Loss*** |
| 1 | 6419082 |
| 2 | 8268540 |
| 3 | 16070855 |
| 4 | 10999488 |
| 5 | 8955656 |
| 6 | 12952902 |
| 7 | 19100222 |
| 8 | 9059830 |
| 9 | 6658574 |
| 10 | 9291178 |
| 11 | 12531829 |
| 12 | 12280056 |

Using the formula,

Loss\_cal = (Work.load.Average.day.\* Absenteeism.time.in.hours)/Service.time

**Appendix A**

lm(formula = Absenteeism.time.in.hours ~ ., data = train)

Residuals:

Min 1Q Median 3Q Max

-0.17636 -0.03384 -0.00789 0.01111 0.91637

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.046e-03 4.996e-02 0.161 0.872110

Reason.for.absence1 6.075e-02 3.207e-02 1.894 0.058759 .

Reason.for.absence2 1.986e-01 1.024e-01 1.939 0.053017 .

Reason.for.absence3 2.930e-02 1.024e-01 0.286 0.774872

Reason.for.absence4 3.865e-02 7.356e-02 0.525 0.599517

Reason.for.absence5 9.519e-03 7.360e-02 0.129 0.897143

Reason.for.absence6 1.410e-01 3.943e-02 3.575 0.000381 \*\*\*

Reason.for.absence7 4.443e-02 3.642e-02 1.220 0.223068

Reason.for.absence8 9.489e-03 4.828e-02 0.197 0.844279

Reason.for.absence9 6.387e-02 7.311e-02 0.874 0.382698

Reason.for.absence10 6.146e-02 2.801e-02 2.194 0.028662 \*

Reason.for.absence11 4.703e-02 2.880e-02 1.633 0.103118

Reason.for.absence12 1.705e-01 4.217e-02 4.044 6.01e-05 \*\*\*

Reason.for.absence13 9.593e-02 2.310e-02 4.153 3.80e-05 \*\*\*

Reason.for.absence14 4.576e-02 3.788e-02 1.208 0.227488

Reason.for.absence15 4.491e-03 7.342e-02 0.061 0.951250

Reason.for.absence16 -2.522e-02 7.333e-02 -0.344 0.731015

Reason.for.absence17 2.455e-03 1.025e-01 0.024 0.980899

Reason.for.absence18 2.316e-02 3.051e-02 0.759 0.448233

Reason.for.absence19 1.209e-01 2.523e-02 4.791 2.13e-06 \*\*\*

Reason.for.absence21 1.699e-02 4.484e-02 0.379 0.704926

Reason.for.absence22 3.174e-02 2.617e-02 1.213 0.225701

Reason.for.absence23 1.887e-03 1.986e-02 0.095 0.924325

Reason.for.absence24 1.776e-02 6.078e-02 0.292 0.770192

Reason.for.absence25 -1.044e-02 2.637e-02 -0.396 0.692189

Reason.for.absence26 2.453e-02 2.579e-02 0.951 0.342015

Reason.for.absence27 8.533e-03 2.366e-02 0.361 0.718531

Reason.for.absence28 2.293e-03 2.062e-02 0.111 0.911495

Day.of.the.week3 -2.264e-03 1.275e-02 -0.178 0.859137

Day.of.the.week4 -1.381e-02 1.268e-02 -1.089 0.276589

Day.of.the.week5 -3.062e-02 1.375e-02 -2.227 0.026350 \*

Day.of.the.week6 -1.860e-02 1.321e-02 -1.408 0.159772

Transportation.expense 3.617e-02 2.233e-02 1.620 0.105801

Distance.from.Residence.to.Work -4.561e-02 1.537e-02 -2.969 0.003122 \*\*

Service.time 8.956e-02 2.934e-02 3.052 0.002380 \*\*

Work.load.Average.day. -1.482e-07 1.348e-07 -1.099 0.272353

Hit.target 2.168e-02 1.866e-02 1.162 0.245772

Son 4.746e-02 1.901e-02 2.496 0.012842 \*

Weight -2.835e-02 1.989e-02 -1.425 0.154764

Height 4.348e-02 2.669e-02 1.629 0.103811

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

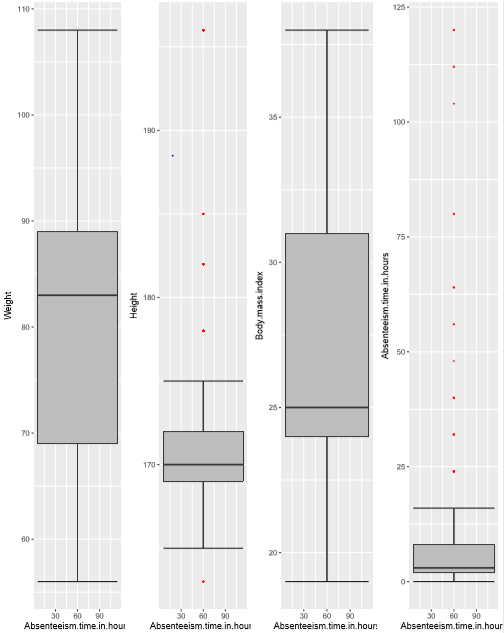
Residual standard error: 0.09966 on 552 degrees of freedom

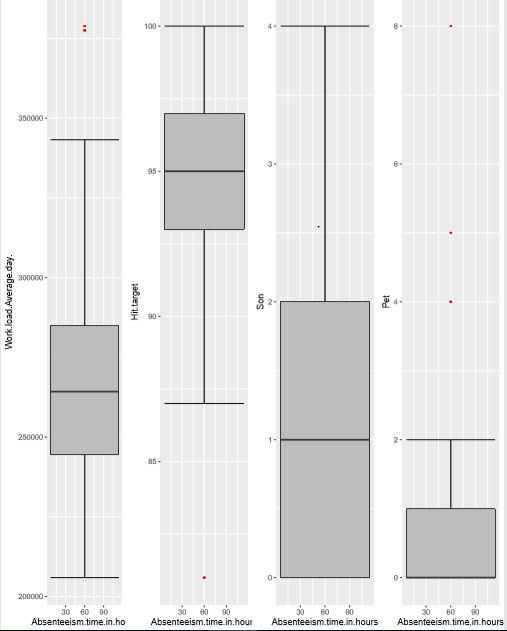
Multiple R-squared: 0.2141, Adjusted R-squared: 0.1586

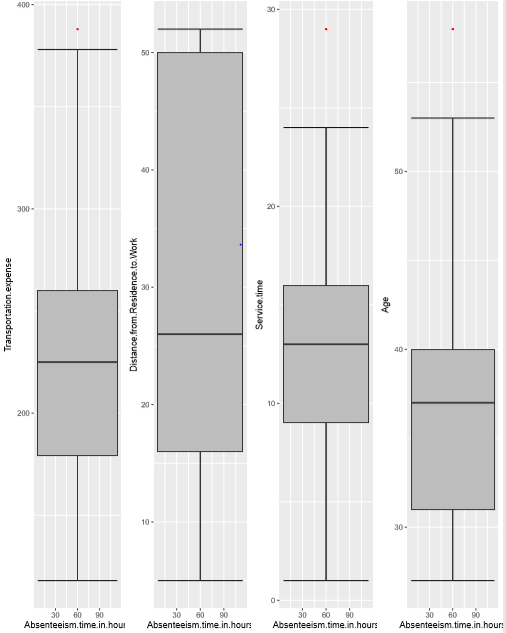
F-statistic: 3.856 on 39 and 552 DF, p-value: 7.089e-13

**Appendix B- Extra figures**

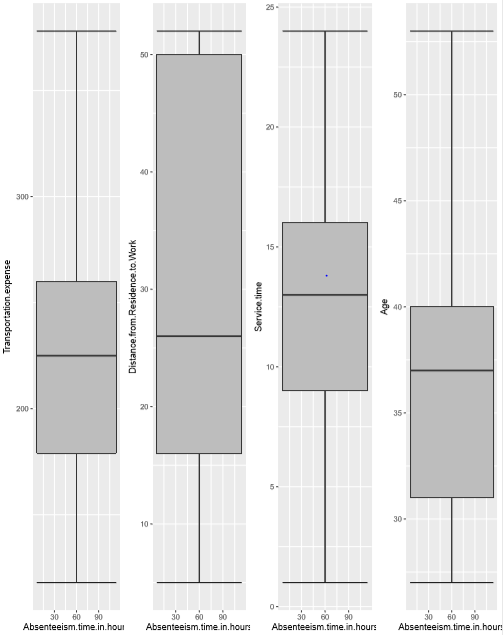
1. **Box plots (with outliers)**

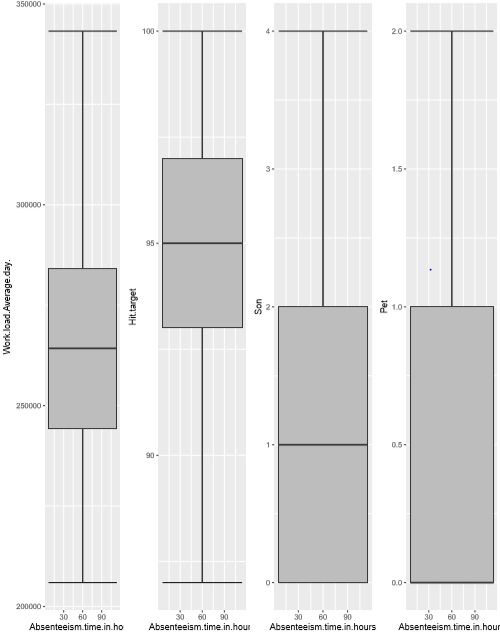


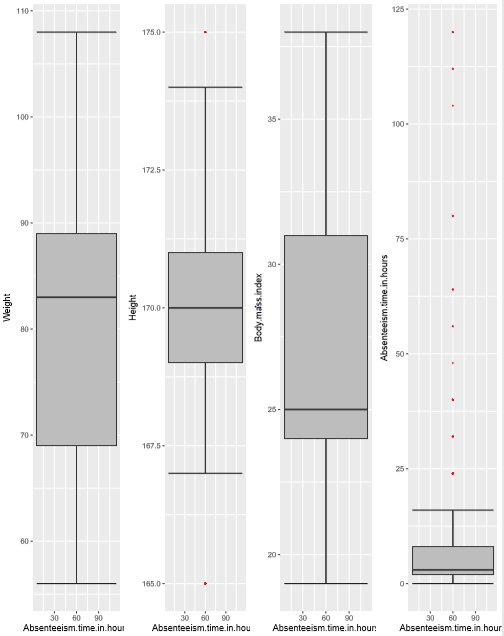




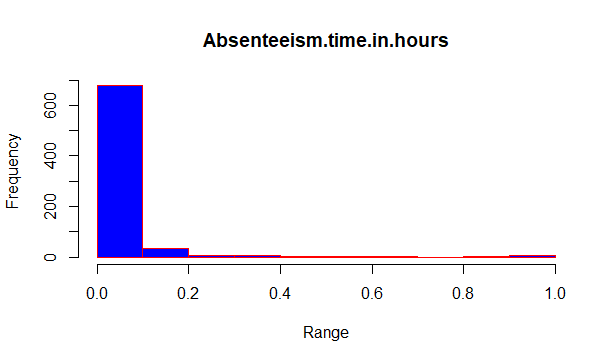
**Box plots(without outliers)**

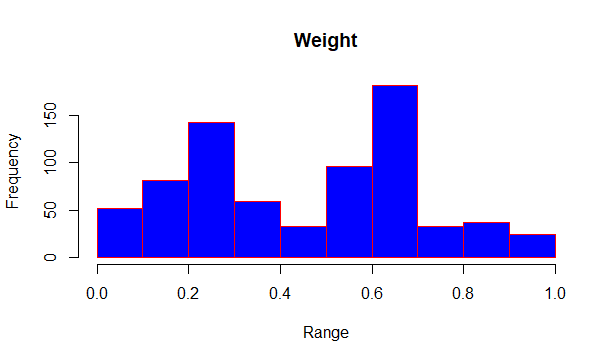


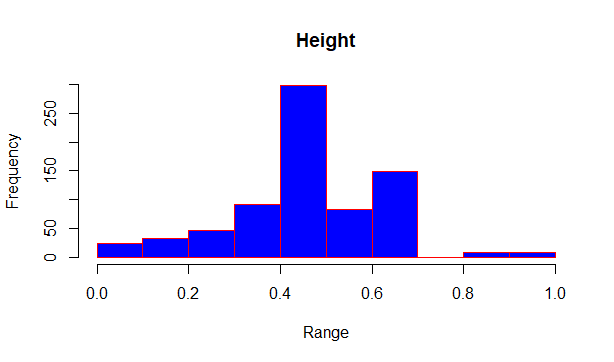


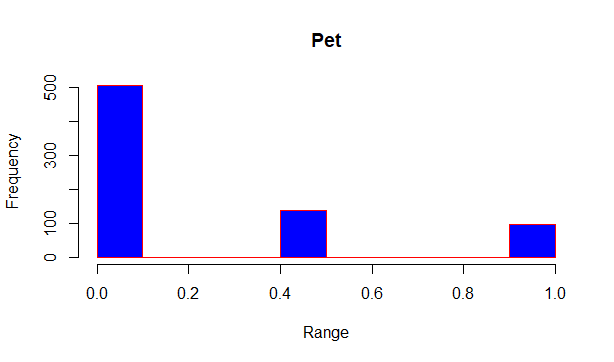


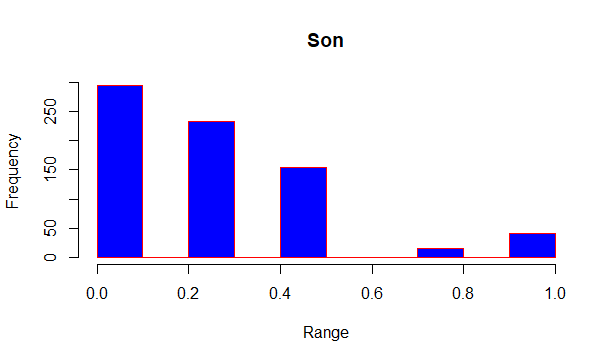
**Normality check of independent variables:-**

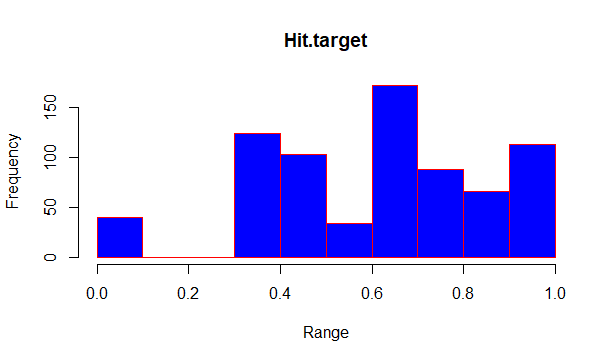
****

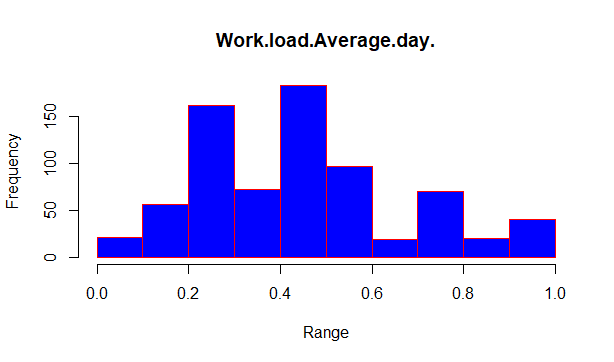
****

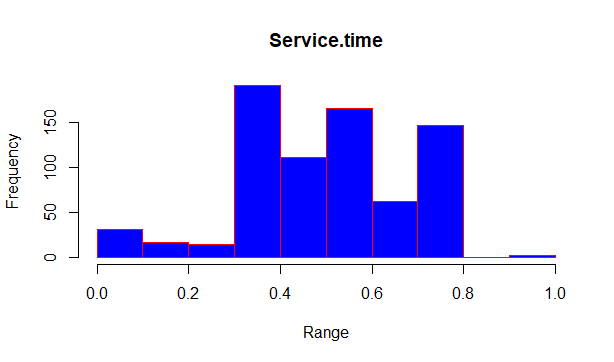
****

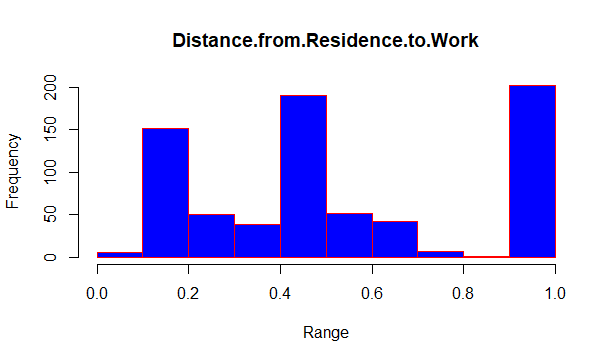
****

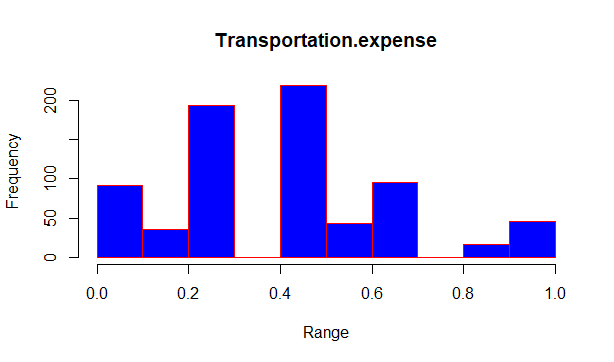
****

****

****

****

****

****

**Appendix C**

**R Code:-**

rm(list=ls(all=T))

setwd("E:/Project 2")

options(warn = -1)

#Load Libraries

x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50", "dummies", "e1071", "Information",

"MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees')

#install.packages(x)

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

rm(x)

## Read the data

library(xlsx)

absent = read.xlsx("Absenteeism\_at\_work\_Project.xls",sheetIndex = 1, header = T)

View(absent)

sum(is.na(absent))

name=subset(absent,select=-c(ID,Day.of.the.week,Seasons,Body.mass.index))

a=colnames(name)

###################################Missing Values Analysis######################

for(i in a){

absent[,i][is.na(absent[,i])] = median(absent[,i], na.rm = T)

}

absent$Body.mass.index[is.na(absent$Body.mass.index)]=mean(absent$Body.mass.index,na.rm = T)

str(absent)

absent$Reason.for.absence=as.factor(absent$Reason.for.absence)

absent$Seasons=as.factor(absent$Seasons)

absent$Social.drinker=as.factor(absent$Social.drinker)

absent$Social.smoker=as.factor(absent$Social.smoker)

absent$Disciplinary.failure=as.factor(absent$Disciplinary.failure)

absent$Day.of.the.week=as.factor(absent$Day.of.the.week)

absent$Education=as.factor(absent$Education)

absent$ID=as.factor(absent$ID)

absent$Month.of.absence=as.factor(absent$Month.of.absence)

###############Outlier Analysis###########################

numeric\_index = sapply(absent,is.numeric) #selecting only numeric

numeric\_data = absent[,numeric\_index]

cnames = colnames(numeric\_data)

for (i in 1:length(cnames))

{

assign(paste0("gn",i), ggplot(aes\_string(y = (cnames[i]), x = "Absenteeism.time.in.hours"), data = subset(absent))+

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

geom\_boxplot(outlier.colour="red", fill = "grey" ,outlier.shape=18,

outlier.size=1, notch=FALSE) +

theme(legend.position="bottom")+

labs(y=cnames[i],x="Absenteeism.time.in.hours")+

ggtitle(paste("Box plot of Absenteeism.time.in.hours for",cnames[i])))

}

## Plotting plots together

gridExtra::grid.arrange(gn1,gn2,gn3,gn4,ncol=4)

gridExtra::grid.arrange(gn5,gn6,gn7,gn8,ncol=4)

gridExtra::grid.arrange(gn9,gn10,gn11,gn12,ncol=4)

#### Removing the outliers and imputing the missing values####

numeric=cnames

numeric=numeric[-11:-12]

for(i in numeric){

val = absent[,i][absent[,i] %in% boxplot.stats(absent[,i])$out]

print(length(val))

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

absent[,i][is.na(absent[,i])] = median(absent[,i], na.rm = T)

}

sum(is.na(absent))

val = absent$Body.mass.index[absent$Body.mass.index %in% boxplot.stats(absent$Body.mass.index)$out]

absent$Body.mass.index[absent$Body.mass.index %in% val] = NA

## Correlation Plot

corrgram(absent[,numeric\_index], order = F,

upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

## Anova test

factor\_index = sapply(absent,is.factor)

factor\_data = absent[,factor\_index]

View(factor\_data)

colnames(factor\_data)

res.aov = aov(formula=Absenteeism.time.in.hours ~ ID, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Reason.for.absence, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Month.of.absence, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Day.of.the.week, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Seasons, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Disciplinary.failure, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Education, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Social.drinker, data = absent)

summary(res.aov)

res.aov = aov(formula=Absenteeism.time.in.hours ~ Social.smoker, data = absent)

summary(res.aov)

### Dimensionality reduction ###

abs\_final=subset(absent,select=-c(ID,Pet,Age,Disciplinary.failure,Month.of.absence,Seasons,Education,Social.drinker,Social.smoker,Body.mass.index))

#Normalisation

cn=c("Transportation.expense" , "Distance.from.Residence.to.Work",

"Service.time", "Hit.target" ,

"Son" ,"Height" , "Weight","Absenteeism.time.in.hours")

#Normality check

for(i in cn){

hist(abs\_final[,i],main = i ,col = "blue",border = "red",xlab="Range")

}

for(i in cn){

print(i)

abs\_final[,i] = (abs\_final[,i] - min(abs\_final[,i]))/

(max(abs\_final[,i] - min(abs\_final[,i])))

}

### Splitting the data set into train and test

set.seed(1234)

train\_ind = sample(1:nrow(abs\_final),0.8\*nrow(abs\_final))

train = abs\_final[train\_ind,]

test = abs\_final[-train\_ind,]

# Decision trees

fit = rpart(Absenteeism.time.in.hours~.,data = train, method = 'anova')

summary(fit)

prediction\_dt = predict(fit,test[,-11])

regr.eval(test[,11],prediction\_dt,stats = "rmse")

# error rate = 11% #accuracy 89%

# Random forest

rf\_mod = randomForest(Absenteeism.time.in.hours~.,train,importance = TRUE,ntree = 500)

rf\_pred = predict(rf\_mod,test[,-11])

regr.eval(test[,11],rf\_pred,stats = "rmse")

#error rate = 10 #accuracy 90

library(rpart)

library(MASS)

#Linear Regression

#check multicollearity

library(usdm)

#droplevels(abs\_final$Reason.for.absence)

#run regression model

lm\_model = lm(Absenteeism.time.in.hours ~., data = train)

#Summary of the model

summary(lm\_model)

#Predict

predictions\_LR = predict(lm\_model, test[,-11])

regr.eval(test[,11],predictions\_LR,stats = "rmse")

#error=10%, accuracy=90%

############Work Loss###############

dim(absent)

loss\_cal=subset(absent,select=c("Month.of.absence","Absenteeism.time.in.hours",

"Work.load.Average.day.","Service.time"))

dim(loss\_cal)

table(absent$Month.of.absence)

loss\_cal$loss=with(loss\_cal,(loss\_cal$Work.load.Average.day.\*loss\_cal$Absenteeism.time.in.hours)/loss\_cal$Service.time)

View(loss\_cal)

month=aggregate(loss\_cal$loss,by=list(loss\_cal$Month.of.absence),sum)[2:13,]

colnames(month)[1]="month"

colnames(month)[2]="loss"

row.names(month)=NULL

View(month)

write.csv(month,"LOSSpermonth.csv",row.names = F)

Python code:

*#Load libraries*

**import** **os**

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**from** **fancyimpute** **import** KNN

**import** **matplotlib.pyplot** **as** **plt**

**from** **scipy.stats** **import** chi2\_contingency

**import** **seaborn** **as** **sns**

**from** **random** **import** randrange, uniform

**from** **scipy** **import** stats

**from** **sklearn** **import** linear\_model

**from** **sklearn.ensemble** **import** RandomForestRegressor

**import** **statsmodels.api** **as** **sm**

**from** **statsmodels.formula.api** **import** ols

In [174]:

*#Set working directory*

os.chdir("E:\Project 2")

In [175]:

*#Load data*

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

In [176]:

*#missing value analysis*

missing\_val=pd.DataFrame(absent.isnull().sum())

*#Reset index*

missing\_val = missing\_val.reset\_index()

*#Rename variable*

missing\_val = missing\_val.rename(columns = {'index': 'Variables', 0: 'Missing\_percentage'})

*#Calculate percentage*

missing\_val['Missing\_percentage'] = (missing\_val['Missing\_percentage']/len(absent))\*100

*#descending order*

missing\_val = missing\_val.sort\_values('Missing\_percentage', ascending = **False**).reset\_index(drop = **True**)

*#save output results*

missing\_val.to\_csv("Missing\_perc.csv", index = **False**)

In [177]:

absent

In [178]:

**for** i **in** range(5,19):

absent.iloc[:,i] = absent.iloc[:,i].fillna(absent.iloc[:,i].median())

**for** i **in** range(1,3):

absent.iloc[:,i] = absent.iloc[:,i].fillna(absent.iloc[:,i].median())

absent["Body mass index"]=absent["Body mass index"].fillna(absent["Body mass index"].mean())

absent["Absenteeism time in hours"]=absent["Absenteeism time in hours"].fillna(absent["Absenteeism time in hours"].median())

In [179]:

lis=["Reason for absence","Seasons","Social drinker","Social smoker","Disciplinary failure","Day of the week","Education","ID","Month of absence"]

**for** i **in** lis:

absent[i]=absent[i].astype(object)

In [180]:

absent\_copy=absent.copy()

In [181]:

lis1=[]

**for** i **in** range(0, absent.shape[1]):

**if**(absent.iloc[:,i].dtypes != 'object'):

lis1.append(absent.columns[i])

lis4=lis1.copy()

**del** lis4[-1]

In [182]:

lis4

Out[182]:

['Transportation expense',

'Distance from Residence to Work',

'Service time',

'Age',

'Work load Average/day ',

'Hit target',

'Son',

'Pet',

'Weight',

'Height',

'Body mass index']

In [183]:

*# #Plot boxplot to visualize Outliers*

%**matplotlib** inline

**for** i **in** lis1:

plt.boxplot(absent[i])

In [184]:

*#Detect outliers and replace with NA*

**for** i **in** lis4:

*# #Extract quartiles*

q75, q25 = np.percentile(absent.loc[:,i], [75,25])

*# #Calculate IQR*

iqr = q75 - q25

*# #Calculate inner and outer fence*

minimum = q25 - (iqr\*1.5)

maximum = q75 + (iqr\*1.5)

*# #Replace with NA*

absent.loc[absent.loc[:,i] < minimum,i] = np.nan

absent.loc[absent.loc[:,i] > maximum,i] = np.nan

In [185]:

pd.DataFrame(absent.isnull().sum())

In [186]:

*#Impute outliers via median method*

**for** i **in** lis4:

absent[i]=absent[i].fillna(absent[i].median())

In [191]:

*#Correlation plot*

absent\_corr = absent.loc[:,lis1]

In [192]:

*#Set the width and hieght of the plot*

f, ax = plt.subplots(figsize=(7, 5))

*#Generate correlation matrix*

corr = absent\_corr.corr()

*#Plot using seaborn library*

sns.heatmap(corr, mask=np.zeros\_like(corr, dtype=np.bool), cmap=sns.diverging\_palette(220, 10, as\_cmap=**True**),

square=**True**, ax=ax)

Out[192]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2ad48c622e8>

In [193]:

**for** i **in** lis:

print(i)

print(stats.f\_oneway(absent["Absenteeism time in hours"],absent[i]))

Reason for absence

F\_onewayResult(statistic=455.3711473915174, pvalue=2.707113523010777e-88)

Seasons

F\_onewayResult(statistic=77.43805808135198, pvalue=3.7436356639463614e-18)

Social drinker

F\_onewayResult(statistic=165.5086864619558, pvalue=5.582372789491098e-36)

Social smoker

F\_onewayResult(statistic=192.82867048087573, pvalue=2.6693616173845234e-41)

Disciplinary failure

F\_onewayResult(statistic=194.00177378775473, pvalue=1.5848851694285747e-41)

Day of the week

F\_onewayResult(statistic=35.90534534236459, pvalue=2.597843591203105e-09)

Education

F\_onewayResult(statistic=129.49899329227654, pvalue=8.030025614992404e-29)

ID

F\_onewayResult(statistic=309.0176938153286, pvalue=5.792879146331886e-63)

Month of absence

F\_onewayResult(statistic=1.1472281864806217, pvalue=0.28430500959032)

In [194]:

**for** i **in** range(0,len(lis)):

**for** j **in** range(i+1,len(lis)):

chi2, p, dof, ex = chi2\_contingency(pd.crosstab(absent[lis[i]], absent[lis[j]]))

**if**(p<=0.05):

print(lis[i],"vs",lis[j])

print(p)

Reason for absence vs Seasons

7.510464854668837e-22

Reason for absence vs Social drinker

3.7248231481975135e-08

Reason for absence vs Social smoker

2.5924166923672465e-09

Reason for absence vs Disciplinary failure

2.6018039430529264e-123

Reason for absence vs Education

1.2324555671642062e-10

Reason for absence vs ID

2.6547906138415276e-61

Reason for absence vs Month of absence

1.0089174328331495e-18

Seasons vs Disciplinary failure

8.428010096294059e-05

Seasons vs ID

2.3159482085127654e-07

Seasons vs Month of absence

0.0

Social drinker vs Social smoker

0.00849237440232344

Social drinker vs Education

7.846131751409077e-35

Social drinker vs ID

1.7935221070986258e-131

Social drinker vs Month of absence

0.009858659892194975

Social smoker vs Disciplinary failure

0.003240642519906935

Social smoker vs Education

3.677653602021038e-21

Social smoker vs ID

5.93573183725426e-133

Social smoker vs Month of absence

0.02370120704079393

Disciplinary failure vs ID

5.921983292784841e-10

Disciplinary failure vs Month of absence

0.00018932044476779238

Day of the week vs ID

3.906507727847447e-05

Education vs ID

0.0

Education vs Month of absence

0.013658150890210824

ID vs Month of absence

6.658724574187815e-71

In [195]:

*#day of week=7, season=4,socil drik=3,smoke=2,displinary=3,edu=3*

In [196]:

abs\_final=absent.drop(['Month of absence','Age','Education','Body mass index','Disciplinary failure','Seasons','Social drinker','Social smoker'],axis=1)

In [197]:

*#abs\_final=absent.drop(['Body mass index','Month of absence'],axis=1)*

In [198]:

lis2=lis1.copy()

**del** lis2[-2]

**del** lis2[3]

In [199]:

lis2

Out[199]:

['Transportation expense',

'Distance from Residence to Work',

'Service time',

'Work load Average/day ',

'Hit target',

'Son',

'Pet',

'Weight',

'Height',

'Absenteeism time in hours']

In [200]:

*#Nomalisation*

**for** i **in** lis2:

print(i)

abs\_final[i] = (abs\_final[i] - min(abs\_final[i]))/(max(abs\_final[i]) - min(abs\_final[i]))

Transportation expense

Distance from Residence to Work

Service time

Work load Average/day

Hit target

Son

Pet

Weight

Height

Absenteeism time in hours

In [201]:

abs\_final

In [220]:

*# ML Algorithm*

*## dividing data into train and test*

*#Import Libraries for decision tree*

**from** **sklearn** **import** tree

**import** **random**

**from** **sklearn.metrics** **import** mean\_squared\_error

**from** **sklearn.tree** **import** DecisionTreeRegressor

**from** **sklearn.metrics** **import** accuracy\_score

**from** **sklearn.cross\_validation** **import** train\_test\_split

train,test = train\_test\_split(abs\_final,test\_size= 0.2)

In [221]:

*# Decision Tree Regression*

fit\_dt = DecisionTreeRegressor(max\_depth = 2).fit(train.iloc[:,0:12],train.iloc[:,12])

In [222]:

predictions\_dt = fit\_dt.predict(test.iloc[:,0:12])

In [223]:

**from** **math** **import** sqrt

rmse\_dt = sqrt(mean\_squared\_error(test.iloc[:,12],predictions\_dt))

print(rmse\_dt)

0.1161471518805244

In [224]:

*# Random Tree Regression*

rf = RandomForestRegressor(n\_estimators = 500, random\_state = 0)

rf.fit(train.iloc[:,0:12],train.iloc[:,12])

predictions\_rf = rf.predict(test.iloc[:,0:12])

In [225]:

rmse\_dt1 = sqrt(mean\_squared\_error(test.iloc[:,12],predictions\_rf))

print(rmse\_dt1)

0.11486909620207172

In [226]:

lis3

Out[226]:

['Reason for absence', 'Day of the week', 'ID']

In [227]:

*# Linear regression*

lis3=lis.copy()

**del** lis3[-1]

**del** lis3[6]

**del** lis3[1:5]

**for** i **in** lis3:

abs\_final[i] = abs\_final[i].astype('float')

train1,test1 = train\_test\_split(abs\_final,test\_size = 0.2)

In [228]:

line\_regression = sm.OLS(train1.iloc[:,12],train1.iloc[:,0:12]).fit()

line\_regression.summary()

predictions\_lr = line\_regression.predict(test1.iloc[:,0:12])

In [229]:

line\_regression.summary()

In [230]:

rmse\_dt = sqrt(mean\_squared\_error(test1.iloc[:,12],predictions\_lr))

print(rmse\_dt)

0.10866112481576658

In [233]:

In [217]:

*########Loss Calculation########################*

loss = absent\_copy[['Month of absence','Service time','Work load Average/day ','Absenteeism time in hours']]

In [218]:

loss["loss/month"]=(loss["Absenteeism time in hours"]\*loss["Work load Average/day "])/loss["Service time"]

C:\Users\sonne\Anaconda3\lib\site-packages\ipykernel\_launcher.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy

"""Entry point for launching an IPython kernel.

In [219]:

loss.columns

Out[219]:

Index(['Month of absence', 'Service time', 'Work load Average/day ',

'Absenteeism time in hours', 'loss/month'],

dtype='object')

In [169]:

month=[]

**for** j **in** range(0,13):

sum=0

**for** i **in** range(0,len(loss)):

**if**(loss["Month of absence"].iloc[i]==j):

sum=sum+loss["loss/month"].iloc[i]

month.append(sum)

z=pd.DataFrame(month)

In [170]:

z["Month"]=z.index

z["Loss"]=z[0]

In [171]:

**del** z[0]

In [172]:

z.iloc[1:len(z),:]

Out[172]:

|  | **Month** | **Loss** |
| --- | --- | --- |
| **1** | 1 | 6.312631e+06 |
| **2** | 2 | 8.268540e+06 |
| **3** | 3 | 1.607086e+07 |
| **4** | 4 | 1.099949e+07 |
| **5** | 5 | 9.693976e+06 |
| **6** | 6 | 1.447528e+07 |
| **7** | 7 | 1.910022e+07 |
| **8** | 8 | 9.059830e+06 |
| **9** | 9 | 6.658574e+06 |
| **10** | 10 | 9.218642e+06 |
| **11** | 11 | 1.253183e+07 |
| **12** | 12 | 1.228006e+07 |

**THANK YOU**