Project Option 1: “Absenteeism”

Qiyan Fang

Sile Li

Ruchil Barya

Michalina Jablonska

Harrisburg University of Science And Technology

Abstract

This study seeks to perform a prediction to solve the problem of absenteeism amongst employees in a courier company. Absenteeism at the workplace reduces performance and corporate financial results. The researchers in this article, based on an assembly of several kinds of literature and course material from ANLY AD 530, Harrisburg University of Science and Technology, explore the factors contributing to absenteeism as well as a solution to handle the situation. A qualitative approach including was implemented when compiling this paper in order to analyze the topic.

This paper is organized with the below structure. First, we introduce the situation, problem or challenge with the advantage of analytics in this context. In the second part, we illustrate the related works including similar situations in other companies with a possible solution and findings from other work. Then the paper follows with the data cleaning and EDA part as a preparation for our research. In the next part, we describe the algorithms we used in the technical approach as well as the whole process. Then we test and evaluate our model and findings to make sure it is valid and accurate with any potential future work, hoping to better address the absenteeism problem. Finally, we conclude our findings with references provided at the bottom.

Keywords: Absenteeism, Courier company, Data cleaning, Exploratory Data Analysis, Decision trees, Random Forest, Naïve Bayes, SVM, PCA, Linear regression

Introduction, motivation and general description

Situation, problem or challenge

Employee Absenteeism is a manager's nightmare in every courier company. It affects planning, production, efficiency, functioning and managerial effectiveness of the organization with heavy additional expenses involved. The objective of this project is to apply machine learning algorithms in the prediction of absenteeism at work.

The raw database is the information collected records of absenteeism from work during the period of July/07 to July/2010 in a Courier company. Absences certified with the International Classification of Diseases were stratified into 21 categories, the data were tabulated and stored in two datasets (training and testing set).

Concept of absenteeism

'Absenteeism is absence from duty without good reason, indicates poor performance, leads to the breach of an agreement between employee and employer.'(Wikipedia)

Why could analytics help?

Based on machine learning algorithms, we believe analytics could help through performing exploratory analysis to track past progress, making more accurate predictions on the future, given the raw dataset has errors and inconsistencies as part of their shortcomings. In addition, it is not straightforward to explain the past or to predict the future without a complete analysis.

Related work

Similar situations from other companies

Hera Group

Hero group is a Italian multiutility leader in environmental, water and energy services. They implemented Full time equivalent (FTE) calculation, which caused self-discipline and reduced absenteeism. (FTE=TE/YWH, TE: Total effort, YMH: Yearly working hours)

Uva Rossa

Uva Rossa is a wine bar located in New York. They performed one-on-one interviews to uncover the reason and solution. Through the interview, they found Anxiety, Stress and Sleeping problems seems to be the major reason. As a result, they helped their employees set up sessions with occupational health officers to overcome psychological issues. In addition, they also considered giving formal warnings or even a firing if excess absenteeism behavior continues.

RMM Foods Private

RMM Food Private Limited is a food supply company located in India. They suffered from financial loss due to productivity reduction and cost of sick leave benefit. Analysts offered suggestions through research but RMM's final solution was unknown.

Related literatures

A growing literature is casting attention to absenteeism, with analyzing the effects on worker behavior of a large number of variables. Some of these variables are individual characteristics related, while others are institutional and contractual aspects related (Federica Cucchiella, 2014). Some works focused on the relationship between unemployment and absence behavior showing that absenteeism has been shown to be negatively related to unemployment, since the threat of termination tends to be related to labor market conditions (Leigh, 1985, Hesselius, 2007). Then, the effects of several legislative changes in benefit levels on absence was examined using US data. They show that increases in these benefits produce an increase of employees' opportunistic behavior. (Curington, 1994; Meyer et al, 1995). Followed with a finding showed the role of workforce in I-O analysis. (Campisi and Gastaldi, 1996) Similarly, the relation between workforce and risks in supply chain management was analyzed in the next years. (Cucchiella and Gastaldi, 2006; Cucchiella et al., 2010). Notably, firing cost was found significant in the following works. Findings show that workers on fixed term contracts, present lower absenteeism(Arai and Thoursie, 2005; Ichino and Riphahn, 2005; Scoppa, 2009). Then in 2008, Frick and Malo conducted cross sectional data analysis. It showed that in countries with higher sickness benefits levels, individuals tend to have more absent behavior accordingly. (Frick and Malo, 2008)

Critical thinking

While with reviewing the past literature, we have a better understanding where we are and what others had at this stage of the research timeline, the literature on the absenteeism issue provides conclusions only for specific sample size, time period, country, industry, etc. Our

analysis is an attempt to reduce this gap by performing analysis on a courier company that has scarcely been investigated by previous works and contributing to a better understanding accordingly.

Data

Data Summary

We used records of absenteeism from work collected during the period from July 2007 to July 2010 in a courier country. The data includes 21 variables with 740 rows in total, and the data was sliced into 2 separate datasets for training set and testing set separately (666 rows of records for training set and 74 records for testing set). Regarding these 21 attributes, we modified their data types for further analysis. Reason for absence, Month of absence, Day of the week, Seasons, Workload Average day, Disciplinary failure, ID, Education, Son, Social Drinker, Social Smoker and Pet are factor variables. Hit target, Absenteeism time in hours, Transportation expense, Distance from Residence to Work, Service time, Age, Weight, Height, and Body mass index are numeric variables. We also defined that ID, Education, Son, Social Drinker, Social Smoker, Pet, Transportation expense, Distance from Residence to Work, Service time, Age, Weight, Height and, Body mass index are ID related variables.

In Figure 1, we calculated the summary of the training set data. We can see that the mean of our target variable Absenteeism time in hours is 6.752, the median value is 3, and the range of it is 0 to 120. In Figure 2, we calculated the summary of the testing set data. We can see that the mean of our target variable Absenteeism time in hours is 8.473, the median value is also 3, and the range of it is 0 to 120.

Missing Data Imputation

There are no missing values in the testing dataset. There're missing values in 3 attributes in the training set, two of those are in "Weight" while both "Age" and "Hit Target" have one. For "Weight" and "age", these two attributes are ID related variables, so we decided to replace the

missing values with the mean value with the same ID of the variable. As for the “Hit Target”, we replaced the NA values using k Nearest Neighbors of each case with NA values (KNN Imputation).

Data Exploration

We did two sets of visual analysis for the purpose of preliminary study for the dataset. As the first part, we generated a series of histograms for the frequency of key variables. Based on the graphs, we can easily get a better understanding of the distribution of each attribute. As shown in Figure 3, the Absenteeism time displays a noticeable skewed distribution and the majority of data points are concentrated in the “0-10 Hours” range. Since the skewness value is a high positive value, this distribution is an asymmetrical distribution with a long tail to the right. Since the kurtosis is greater than zero, this distribution has heavier tails and is a leptokurtic distribution. This means that more than half of the records have less absenteeism time than the average absenteeism time of all the records. In other words, the few records with most absenteeism time had much more absenteeism time than the rest of the records. As for the Transportation expense, the histogram indicates that it tends to fit better to a normal distribution with the majority of values concentrated between 100 and 300 range. The mean of transportation expense is 205, which is close to its median value 207.

As for the second part, we conducted further investigation into the dataset using the boxplot graphs. As shown in Figure 4, for Absenteeism time, similarly we can see the values tend to be more evenly concentrated in the range from 100 to 300, which also supports our interpretation based on what we saw in the histogram. Also, we noted that for some of the

variables, for example Height and Hit Target, it shows that there are quite some outliers in the boxplot and the data points distribution are more scattered.

Outliers

As discussed in the visual analysis above, we identified the outliers issue in attribute Hit Target, therefore we would need to conduct pre-processing to clean the data. In total, there're 19 outliers which are NA values in "Hit Target", and we decided to replace these NA values (outliers) using k Nearest Neighbors of each case with NA values since it is not an ID related variable.

Model Creation

Model Creation: Task 1

The approach is to run different models and choose a model which provides the best accuracy and least error. 'Absenteeism.time.in.hours' is used as the target variable in a categorical variable with levels 'Group 1' (number of absent hours is equal to 0), 'Group 2' (number of absent hours is less than or equal to 6), and 'Group 3' (number of absent hours are greater than 6), to predict the number of absent hours.

Decision Trees

Since the target variable is categorical, decision trees model is the first method applied to the training set as part of supervised learning. The model provides pretty good results with an accuracy of 74.32%. It also gives an accuracy of 100% for 'Group 1'.The important features are 'Reason.for.absence', 'Transportation.expense', 'Work.load.average.day', 'Height', 'Son', 'Pet'.The detailed results are shown in table.

Naïve Bayes Classifier

It is considered as one of the simplest methods and provides results quickly, the results are based on probabilities. The correlation of predictor variables is checked before the application of the model.'Body.Mass.Index' and 'Service.time' have high correlation value, so these predictor variables are dropped. This model gives an accuracy of 74.32%. The detailed results are shown in table.1

KNN (K-Nearest Neighbor) Clustering

This is a supervised clustering technique, used for classification. The best K value is 25 which is chosen as the integer number of the square root of the number of records in the training set. The model accuracy is 70.27%. The detailed results are shown in table.1

K-means Clustering

This is an unsupervised clustering technique, used for clustering**.** The best values of k is 2 which is chosen as a result of the elbow method**.** The model gives an accuracy of 33% which is the worst among all kinds of modelling techniques.

Support Vector Machine

Support Vector Machine is a supervised machine learning technique, used for classification or regression. It requires less computational power and provides high accuracy. It is applied as a linear, poly and rbf kernel. The best accuracy is achieved with linear function as 73% in the current scenario. The detailed results are shown in table.1

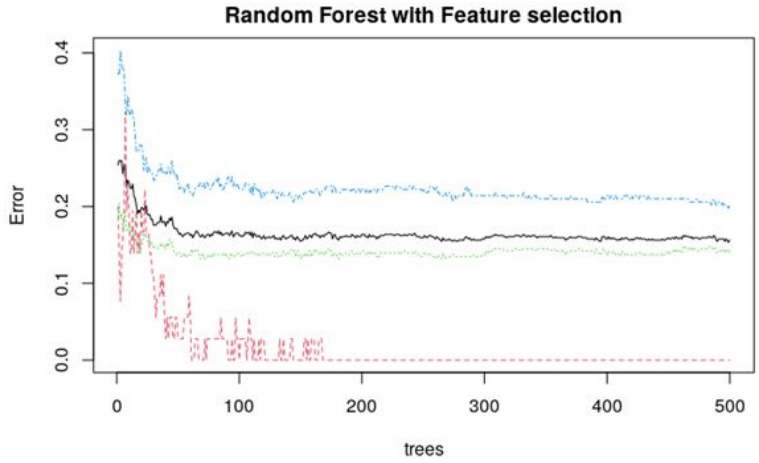
Random Forest

It is an ensemble learning method used for classification and regression, works by generating many decision trees. Overall, 500 trees are used to generate this model. This model provides an accuracy of 75.67%. The important features are ‘Reason.for.absence’,’Transportation.expense’,‘Work.load.average.day’,’Day.of.the.week’,’Month.of.absence’, ’Age’. The detailed results are shown in table.1

Random Forest with feature selection

Based on the feature importance of the above Random Forest Model, this model is created. The predictors in this model are ‘Reason.for.absence’,’Transportation.expense’, ‘Work.load.average.day’,’Day.of.the.week’,’Month.of.absence’, ’Age’ based on the Gini Index value. The best accuracy is 78.4%, achieved in this model. The detailed results are shown in Table. 1.

Fig. 1 Random Forest Error plot



Model creation: Task 2

The approach is to run different models before and after PCA (principal component analysis) and to choose a model which provides best accuracy and least RMSE (root mean square error). 'Absenteeism.time.in.hours' is used as the target variable in a continuous form to predict the number of absent hours.

Principal Component Analysis

When the model is applied directly without any modifications in the independent variables. The accuracy is low, and error is high. This is caused due to many categorical variables. The independent (predictor) variables only consists of two types of variables, 10 numerical variables and 10 categorical variables.'Reason.of.absence' variable contains 28 different levels i.e. reasons of absence. So, the categorical variables are converted by dummy coding and the principal component analysis is applied later to reduce the number of dimensions. As per the elbow method, the first 60 dimensions are required. Models are applied to the new training set with 60 dimensions.

Assumptions for linear regression

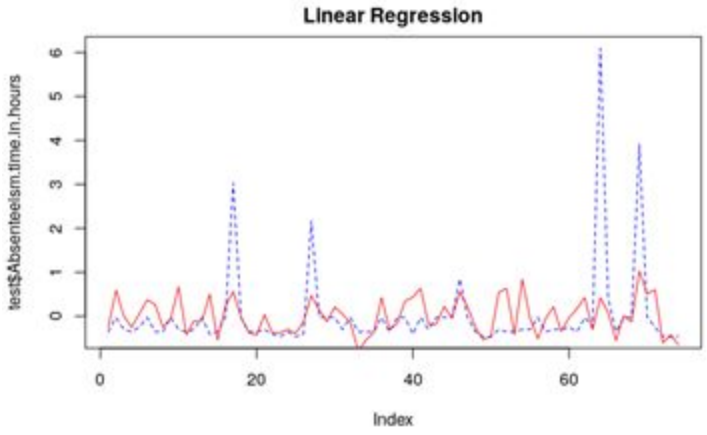
As per Normal Q-Q plot, the assumption of linearity is met as the actual values are close to the regression line. It does not meet the assumption of normality as the histogram of standardized values is very steep. The kurtosis value of 'Height' is quite high, so it has been removed from the analysis. The skewness value for all the independent variables is within the range. As per the standardized plot, it does not meet the assumption of homoscedasticity and homogeneity as the points are spread unevenly. As per the Correlation Matrix, it does not meet

the assumption of Multicollinearity as Body.mass.Index is having high correlation with weight, so to meet the assumption the variable Body.mass.Index is dropped from analysis. Disciplinary.failure is also dropped from the analysis as it does not add anything. To meet the assumption of homoscedasticity and homogeneity, Bootstrapping is used to validate if the results are within the confidence Interval.

Linear Regression

Linear regression is used to predict the absent hours. The model is applied to the training set before and after Principal Component Analysis. Since there are many dimensions, stepwise 'backward' regression is applied. The R2 variance explained by the model before PCA is .10 and after PCA is 0.17. The RMSE (root mean square error) before PCA is .95 and after PCA is 0.90. The main predictors are 'Reason.for.absence','Distance.from.Residence.to.Work','Social.drinker.Linear regression generates the best model after reducing the dimensions by PCA.

Fig.2 Linear Regression Actual v/s Predicted



Random forest with feature selection

This model gives better than linear regression. The R2 variance explained by the model before PCA is 0.15 and after PCA is 0.08. The RMSE (root mean square error) before PCA is 0.92 and after PCA is 0.94. The main predictors are ‘Reason.for.absence’,’Distance.from.Residence.to.Work’,’Social.drinker,’ ‘Month.of.absence’,’Day.of.the.week’,’Age’,’Work.load.Average.day’,’Hit.target’,’Weight’.

Test and evaluation

Model Evaluation parameters like accuracy , precision , recall for all models can be seen in Table 1 (for Task 1)& Table 2 (for Task 2) respectively. Our approach was to test possibly the highest amount of applicable models as a learning curve, we present all relevant ROC related metrics in a form of table to enable easy comparison.

After implementing above described adjustments and calculating ROC metrics for our models, we decided that the model with the best fit for our data is the Random Forest with feature selection. This model reached 78.37% accuracy and AUC of 0.93. The TPR to FPR ratio is illustrated in Fig. 3. We concluded this model has the best area under the curve compared to the other evaluated models.

Fig. 3 ROC curve for the Best Model (Random Forest with Feature selection)

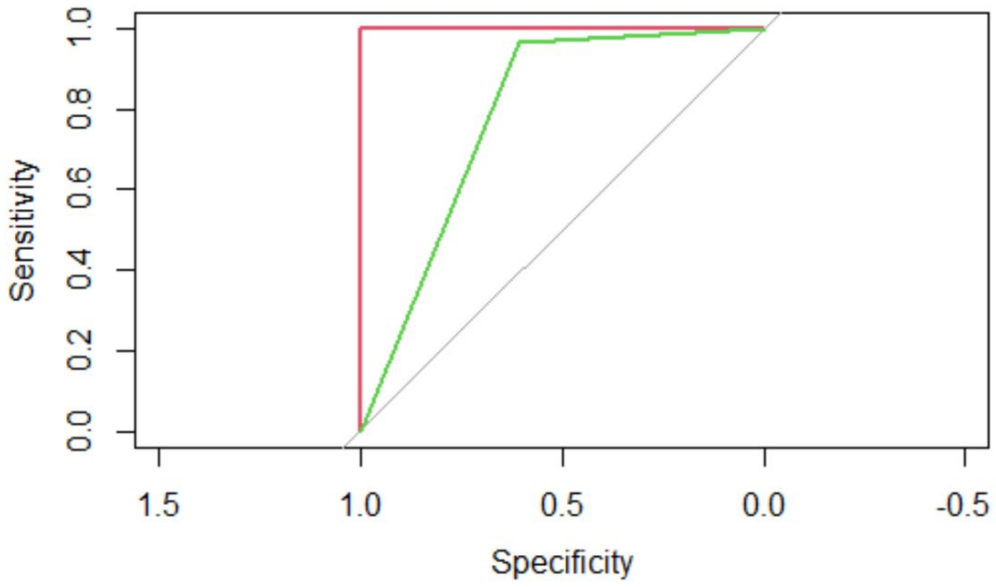


Table 1. Model evaluation comparison for Task 1

Model	Accuracy	Precision	Recall	AUC
Knn with PCA	66.2%	0.45	0.46	NA
Knn	70.27%	0.79	0.64	NA
Decision Trees	74.32%	0.80	0.39	0.91
Random Forest	75.67%	0.82	0.63	0.78
Random Forest with feature selection (Best Model)	**78.37%**	**0.84**	**0.78**	**0.93**
Naive Bayes	74.32%	0.89	0.44	0.87
Naive Bayes with classifier	74.32%	0.79	0.44	0.88
SVM (vanilladot)	72.98%	0.79	0.62	0.90
SVM (polydot)	72.98%	0.79	0.62	0.90
SVM (rbfdot)	75.68%	0.78	0.62	0.83

The Random Forest with feature selection also resulted in a satisfactory precision score of .84. The recall results of 0.78 would still indicate that we could further evaluate this model for testing for fit.

Table 2. Model Comparison for task 2 (Predicting Absenteeism)

	R-2 Score		**RSME**	
Model Name	Before PCA	After PCA	Before PCA	After PCA
Linear Regression	0.10	0.17	0.95	0.90
Random Forest w feature selection	0.15	0.08	0.92	0.94

For Task2 , Linear Regression After PCA gives the best results R2 (variability explained by model) as .17 and RMSE as 0.90 which is least among all.The detailed results can be found in Table2. When we tried to predict the absent hours as a continuous variable results were not good, it is better to split the continuous variable into groups as categorical variable to achieve better accuracy and best precision which was achieved in Task1.

Conclusion

In Conclusion, to predict the number of absent hours, reason for absence is the most important factor.The other important factors are Transportation expense, Day of the week, Month of absence, Age and Work load on average basis.Our results show that most of the employees are aged between 30-40 years. In reason of absence, the major causes of absenteeism are diseases related to musculoskeletal system and connective tissues ,injury, poisoning(digestive diseases), consultation for medical and dental diseases. Higher percentage of Social Drinkers are absent as compared to Non Drinkers and Non Smokers. Many employees are absent in the month of March. During the Spring season,many employees were absent.These results were found as part of the task 1.

However, in task 2 analysis, results are different. First of all, the variance explained by the model is very less which showed that classification as a categorical variable is better. The model predicted that the main reason for absenteeism are reason for absence along with Distance from work and Social Drinker.The last two predictors are not important factor in task1. The reason can be the less number of observations and disparity in the data.Since the task 2 results are not reliable, we have concluded our results based on the task 1.

Limitations

As training data was limited, which resulted in limited variability.This can be overcome by adding more number of observations in the data. The split in the target variable was uneven which resulted in higher bias for Group 2 which can be reduced by adding more features like salary,work experience.

The methods of model evaluation are somewhat limited for multiclass models in R . Based on this challenge, some of the code for model evaluation (precision and recall) had to be done manually which could potentially make it prone to errors.

Many test observations were misclassified as Group 3 but it was Group 2 which led to less accuracy. In other words, the model was not efficient in correctly classifying between the number of absent hours for less than or equal to 6 hours.

Future Aspects

Based on the limitations presented above, it would be better to add more number of features and observations to the data. Neural Network algorithm should be a more efficient method for the prediction of absenteeism, which could lead to higher accuracy.

References

1. Cucchiella, Federica, Gastaldi, Massimo, and Ranieri, Luigi. "Managing Absenteeism in the Workplace: The Case of an Italian Multiutility Company." *Procedia, Social and Behavioral Sciences* 150 (2014): 1157-166.
2. Badubi, R.M. (2017). A Critical Risk Analysis of Absenteeism in the Work Place. *Journal of International Business Research and Marketing,* 2(6), 32-36.
3. Patcha, Bhujanga & Bhujanga Rao, Dr Patcha. (2014). Employee Absenteeism, A Case Study with reference to RMM Foods Private Limited. *Research Journal of Social Science and Management (2014)*, 04-04, 224-231
4. "New Findings from University of Valladolid Describe Advances in Machine Learning (Predicting Absenteeism and Temporary Disability Using Machine Learning: A Systematic Review and Analysis)." *Journal of Engineering (2020)*: 1150.
5. Montano, Isabel Herrera, Marques, Gonçalo, Alonso, Susel Góngora, López-Coronado, Miguel, and De La Torre Díez, Isabel. "Predicting Absenteeism and Temporary Disability Using Machine Learning: A Systematic Review and Analysis." *Journal of Medical Systems 44.9 (2020)*: 162.
6. Singer, Gonen, and Cohen, Izack. "An Objective-Based Entropy Approach for Interpretable Decision Tree Models in Support of Human Resource Management: The Case of Absenteeism at Work." *Entropy (Basel, Switzerland) 22.8 (2020)*: 821.
7. Lary, Maria-Anna, Allsopp, Leslie, Lary, David J, and Sterling, David A. "Using Machine Learning to Examine the Relationship between Asthma and Absenteeism." *Environmental Monitoring and Assessment* 191.S2 (2019): 1-9.

8. Ali Shah, Syed Atif, Uddin, Irfan, Aziz, Furqan, Ahmad, Shafiq, Al-Khasawneh, Mahmoud Ahmad, and Sharaf, Mohamed. "An Enhanced Deep Neural Network for Predicting Workplace Absenteeism." *Complexity (New York, N.Y.) 2020 (2020)*: 1-12.

Appendix I

Figure 1 Summary of Training Set

```
       ID        Reason.for.absence Month.of.absence Day.of.the.week    Seasons      Transportation.expense Distance.from.Residence.to.Work
 Min.   : 1.00   Min.   : 0.00      Min.   : 1.000   Min.   :2.000   Min.   :1.000   Min.   :  0.0          Min.   : 5.00
 1st Qu.: 7.00   1st Qu.:13.00      1st Qu.: 3.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:179.0          1st Qu.:17.00
 Median :18.00   Median :23.00      Median : 7.000   Median :4.000   Median :2.000   Median :225.0          Median :26.00
 Mean   :17.67   Mean   :19.47      Mean   : 6.441   Mean   :3.893   Mean   :2.553   Mean   :222.8          Mean   :30.37
 3rd Qu.:28.00   3rd Qu.:26.00      3rd Qu.:10.000   3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:260.0          3rd Qu.:50.00
 Max.   :36.00   Max.   :28.00      Max.   :12.000   Max.   :6.000   Max.   :4.000   Max.   :388.0          Max.   :52.00

  Service.time       Age      Work.load.Average.day   Hit.target     Disciplinary.failure   Education          Son        Social.drinker
 Min.   : 1.0   38     :112   222,196: 35           Min.   : 81.00   Min.   :0.00000      Min.   :1.000   Min.   :0.000   Min.   :0.0000
 1st Qu.: 9.0   28     :109   264,249: 33           1st Qu.: 92.00   1st Qu.:0.00000      1st Qu.:1.000   1st Qu.:0.000   1st Qu.:0.0000
 Median :13.0   37     : 67   343,253: 29           Median : 95.00   Median :0.00000      Median :1.000   Median :1.000   Median :1.0000
 Mean   :12.7   40     : 50   265,017: 28           Mean   : 94.41   Mean   :0.05405      Mean   :1.246   Mean   :1.029   Mean   :0.5841
 3rd Qu.:16.0   33     : 48   284,853: 25           3rd Qu.: 97.00   3rd Qu.:0.00000      3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:1.0000
 Max.   :29.0   36     : 47   308,593: 24           Max.   :100.00   Max.   :1.00000      Max.   :4.000   Max.   :4.000   Max.   :1.0000
                (Other):233   (Other):492           NA's   :1
 Social.smoker          Pet             Weight          Height      Body.mass.index Absenteeism.time.in.hours
 Min.   :0.00000   Min.   :0.0000   Min.   : 56.00   Min.   :163.0   Min.   :19.00   Min.   :  0.000
 1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.: 69.00   1st Qu.:169.0   1st Qu.:24.00   1st Qu.:  2.000
 Median :0.00000   Median :0.0000   Median : 83.00   Median :170.0   Median :25.00   Median :  3.000
 Mean   :0.06907   Mean   :0.6907   Mean   : 79.21   Mean   :171.9   Mean   :26.82   Mean   :  6.752
 3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.: 89.00   3rd Qu.:172.0   3rd Qu.:31.00   3rd Qu.:  8.000
 Max.   :1.00000   Max.   :8.0000   Max.   :108.00   Max.   :196.0   Max.   :38.00   Max.   :120.000
                                    NA's   :2
```

Figure 2 Summary of Testing Set

```
       ID        Reason.for.absence Month.of.absence Day.of.the.week    Seasons      Transportation.expense Distance.from.Residence.to.Work
 Min.   : 1.00   Min.   : 0.00      Min.   :0.00     Min.   :2.000   Min.   :1.000   Min.   :118.0          Min.   :10.00
 1st Qu.:14.00   1st Qu.:10.00      1st Qu.:5.00     1st Qu.:3.000   1st Qu.:1.250   1st Qu.:155.0          1st Qu.:13.00
 Median :22.00   Median :19.00      Median :5.00     Median :4.000   Median :3.000   Median :207.0          Median :21.00
 Mean   :21.11   Mean   :16.93      Mean   :5.27     Mean   :4.108   Mean   :2.473   Mean   :205.8          Mean   :22.97
 3rd Qu.:28.75   3rd Qu.:25.00      3rd Qu.:6.00     3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:235.0          3rd Qu.:26.00
 Max.   :36.00   Max.   :28.00      Max.   :7.00     Max.   :6.000   Max.   :3.000   Max.   :378.0          Max.   :52.00
  Service.time       Age        Work.load.Average.day   Hit.target    Disciplinary.failure   Education          Son
 Min.   : 1.00   Min.   :28.00   237,656:32           Min.   :91.00   Min.   :0.00000      Min.   :1.000   Min.   :0.0000
 1st Qu.: 9.00   1st Qu.:30.25   246,288: 8           1st Qu.:93.00   1st Qu.:0.00000      1st Qu.:1.000   1st Qu.:0.0000
 Median :10.50   Median :36.50   264,604:12           Median :96.00   Median :0.00000      Median :1.000   Median :1.0000
 Mean   :11.24   Mean   :36.82   271,219: 3           Mean   :96.23   Mean   :0.05405      Mean   :1.703   Mean   :0.9324
 3rd Qu.:14.00   3rd Qu.:40.00   275,089:19           3rd Qu.:99.00   3rd Qu.:0.00000      3rd Qu.:3.000   3rd Qu.:2.0000
 Max.   :24.00   Max.   :58.00                        Max.   :99.00   Max.   :1.00000      Max.   :4.000   Max.   :3.0000
 Social.drinker   Social.smoker         Pet             Weight          Height      Body.mass.index Absenteeism.time.in.hours
 Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   : 56.00   Min.   :163.0   Min.   :19.00   Min.   :  0.000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.: 68.00   1st Qu.:171.0   1st Qu.:22.00   1st Qu.:  2.000
 Median :0.0000   Median :0.0000   Median :0.000   Median : 75.00   Median :172.0   Median :25.00   Median :  3.000
 Mean   :0.4189   Mean   :0.1081   Mean   :1.243   Mean   : 77.38   Mean   :174.3   Mean   :25.43   Mean   :  8.473
 3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:2.000   3rd Qu.: 88.00   3rd Qu.:178.0   3rd Qu.:28.00   3rd Qu.:  8.000
 Max.   :1.0000   Max.   :1.0000   Max.   :8.000   Max.   :106.00   Max.   :196.0   Max.   :38.00   Max.   :120.000
```

Figure 3 Histograms of Multiple Variables

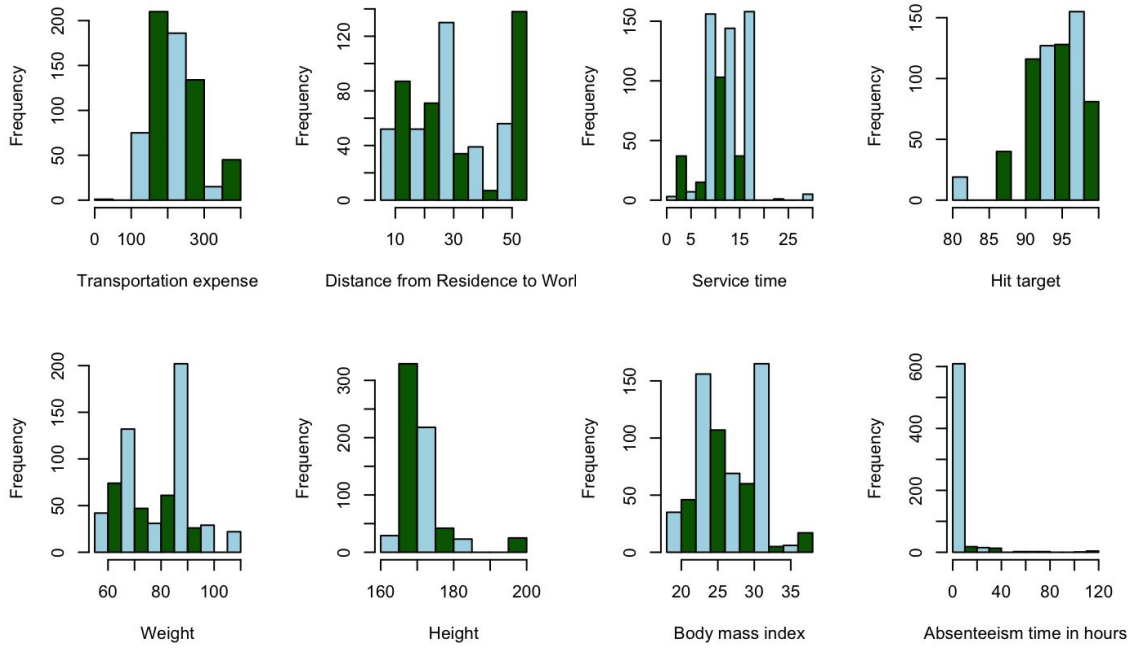


Figure 4 Boxplots of Multiple Variables

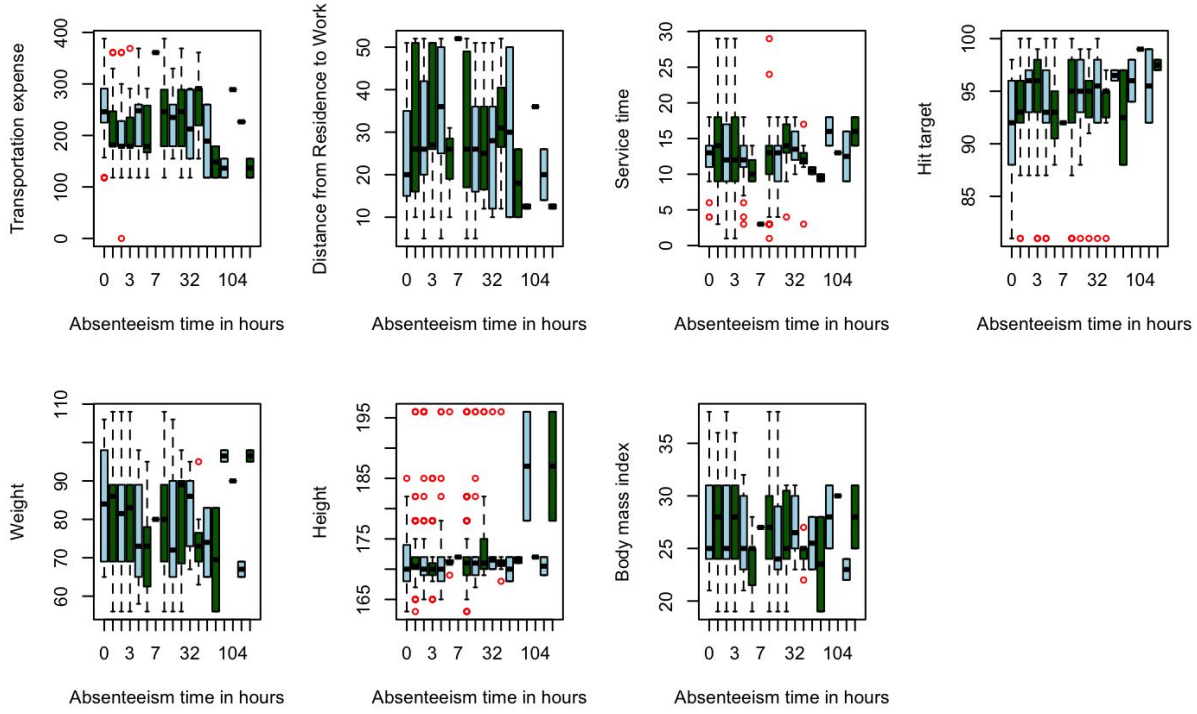


Figure 5 Decision Trees

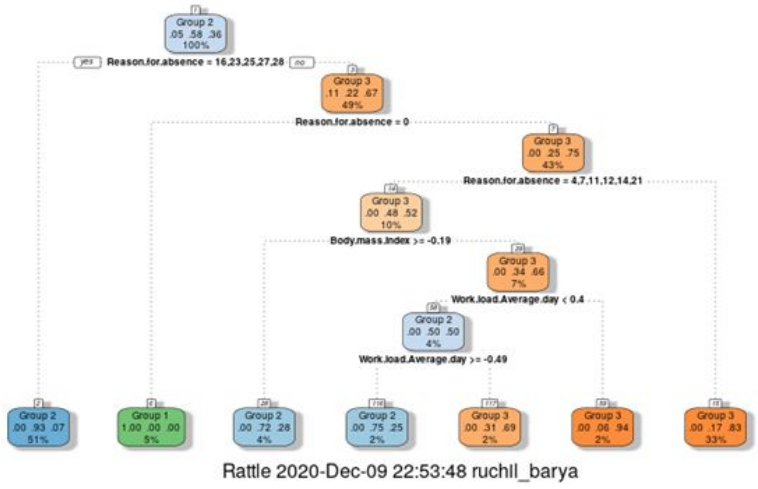


Figure 6 Number of cluster for k-means Analysis

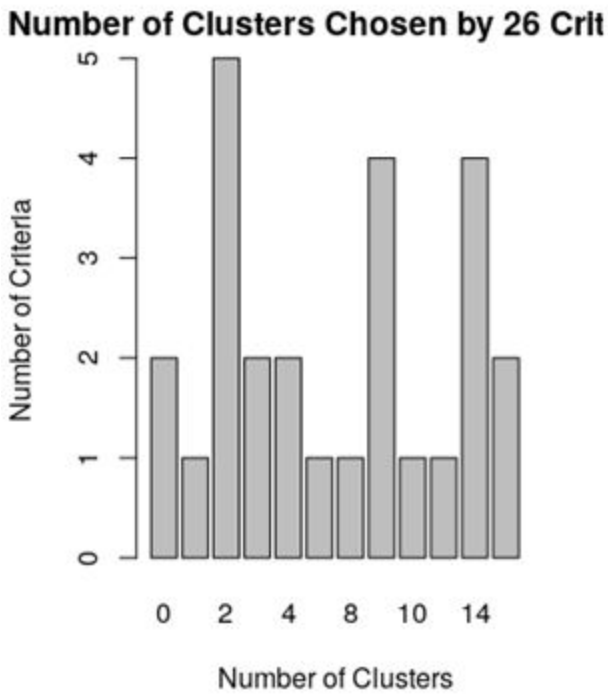


Figure 7 Random forest Prediction

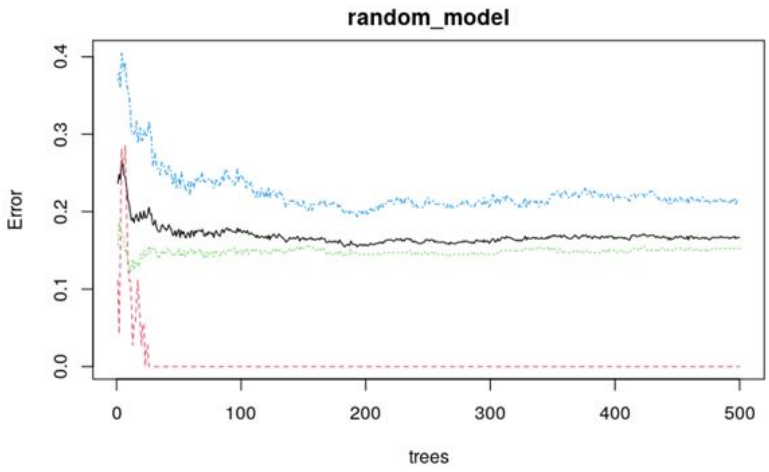


Figure 8 Principal Component Analysis

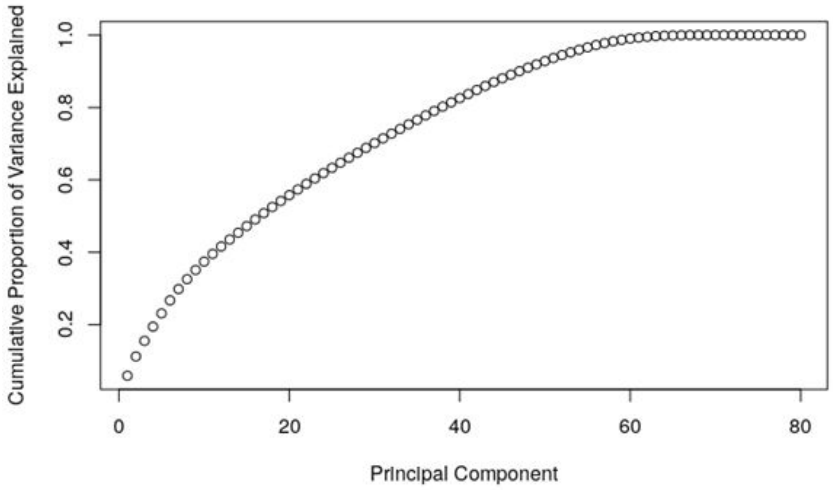


Figure 9 Assumptions for Linear Regression

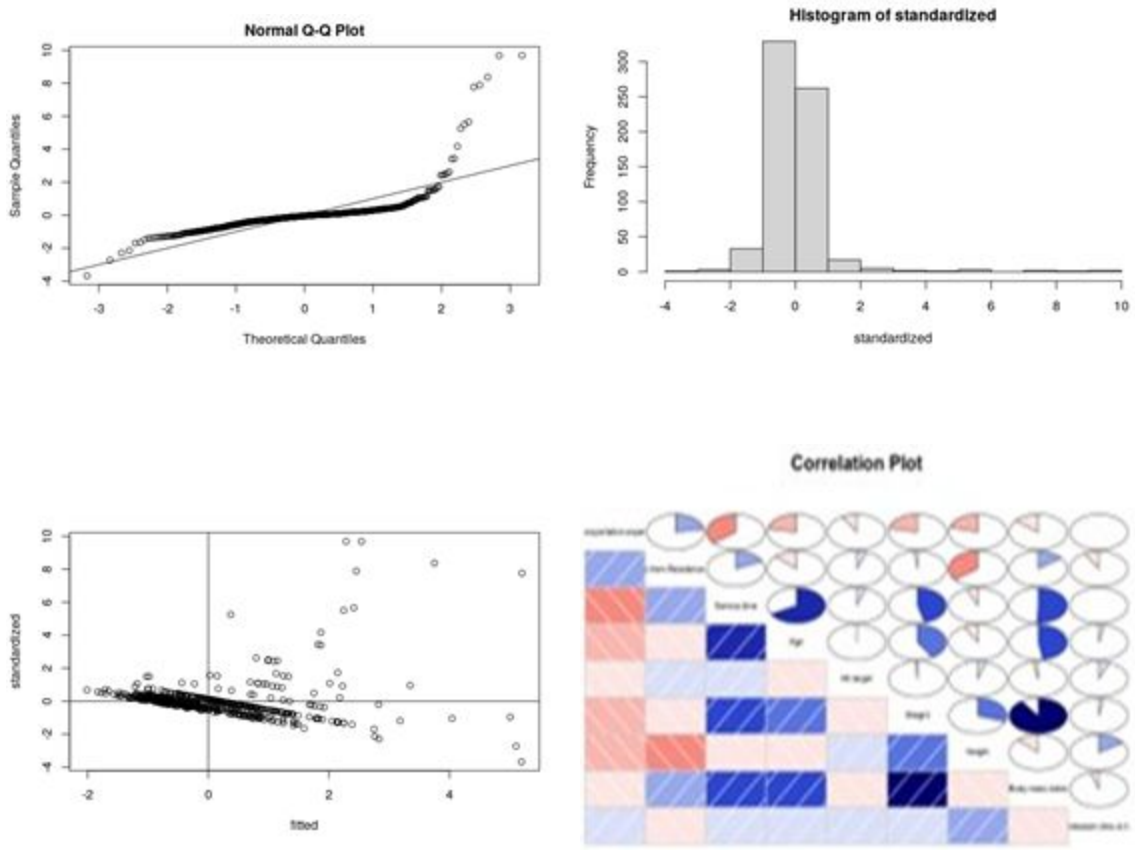
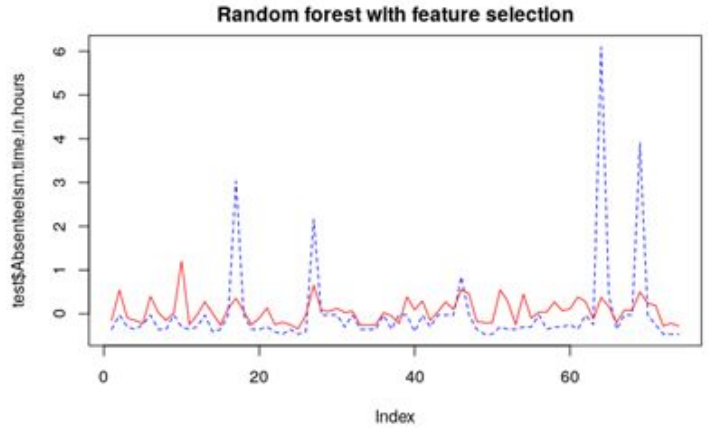


Figure 10 Prediction with Random Forest



Appendix II - R Codes

```
#Loading required libraries


library(corrgram)
library(plyr)

library(dplyr)

library(ggplot2)

library(gridExtra)

library(corrplot)

library(DMwR)

#Loading train data
train = read.csv("Absenteeism_at_work_train.csv")
test = read.csv("Absenteeism_at_work_test.csv")
```

Training Data Processing

Summary

```
##        ID        Reason.for.absence Month.of.absence Day.of.the.week
##  Min.   : 1.00   Min.   : 0.00      Min.   : 1.000   Min.   :2.000
##  1st Qu.: 7.00   1st Qu.:13.00      1st Qu.: 3.000   1st Qu.:3.000
##  Median :18.00   Median :23.00      Median : 7.000   Median :4.000
##  Mean   :17.67   Mean   :19.47      Mean   : 6.441   Mean   :3.893
##  3rd Qu.:28.00   3rd Qu.:26.00      3rd Qu.:10.000   3rd Qu.:5.000
##  Max.   :36.00   Max.   :28.00      Max.   :12.000   Max.   :6.000
##
##     Seasons      Transportation.expense Distance.from.Residence.to.Work
##  Min.   :1.000   Min.   :  0.0          Min.   : 5.00
##  1st Qu.:2.000   1st Qu.:179.0          1st Qu.:17.00
##  Median :2.000   Median :225.0          Median :26.00
##  Mean   :2.553   Mean   :222.8          Mean   :30.37
##  3rd Qu.:4.000   3rd Qu.:260.0          3rd Qu.:50.00
##  Max.   :4.000   Max.   :388.0          Max.   :52.00
##
##   Service.time       Age      Work.load.Average.day   Hit.target
##  Min.   : 1.0   38     :112   222,196: 35           Min.   : 81.00
##  1st Qu.: 9.0   28     :109   264,249: 33           1st Qu.: 92.00
##  Median :13.0   37     : 67   343,253: 29           Median : 95.00
##  Mean   :12.7   40     : 50   265,017: 28           Mean   : 94.41
##  3rd Qu.:16.0   33     : 48   284,853: 25           3rd Qu.: 97.00
##  Max.   :29.0   36     : 47   308,593: 24           Max.   :100.00
##                 (Other):233   (Other):492           NA's   :1
##  Disciplinary.failure   Education          Son        Social.drinker
```

```
##  Min.   :0.00000      Min.   :1.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.00000      1st Qu.:1.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.00000      Median :1.000   Median :1.000   Median :1.0000
##  Mean   :0.05405      Mean   :1.246   Mean   :1.029   Mean   :0.5841
##  3rd Qu.:0.00000      3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.00000      Max.   :4.000   Max.   :4.000   Max.   :1.0000
##
##  Social.smoker          Pet             Weight          Height
##  Min.   :0.00000   Min.   :0.0000   Min.   : 56.00   Min.   :163.0
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.: 69.00   1st Qu.:169.0
##  Median :0.00000   Median :0.0000   Median : 83.00   Median :170.0
##  Mean   :0.06907   Mean   :0.6907   Mean   : 79.21   Mean   :171.9
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.: 89.00   3rd Qu.:172.0
##  Max.   :1.00000   Max.   :8.0000   Max.   :108.00   Max.   :196.0
##                                     NA's   :2
##  Body.mass.index Absenteeism.time.in.hours
##  Min.   :19.00   Min.   :  0.000
##  1st Qu.:24.00   1st Qu.:  2.000
##  Median :25.00   Median :  3.000
##  Mean   :26.82   Mean   :  6.752
##  3rd Qu.:31.00   3rd Qu.:  8.000
##  Max.   :38.00   Max.   :120.000
##
```

Pre-processing

```
train <- train %>% mutate(
    Age = as.character(Age),
    Age = ifelse(Age == "R" , NA, Age),
    Age = ifelse(Age == "0" , NA, Age),
    Age = as.numeric(Age)
  )


  train$ID <- as.factor(train$ID)
  train$Reason.for.absence <- as.factor(train$Reason.for.absence)
  train$Month.of.absence <- as.factor(train$Month.of.absence)
  train$Day.of.the.week <- as.factor(train$Day.of.the.week)
  train$Seasons <- as.factor(train$Seasons)
  train$Disciplinary.failure <- as.factor(train$Disciplinary.failure)
  train$Education <- as.factor(train$Education)
  train$Son <- as.factor(train$Son)
  train$Social.drinker <- as.factor(train$Social.drinker)
  train$Social.smoker <- as.factor(train$Social.smoker)
  train$Pet <- as.factor(train$Pet)

#Selecting factor columns
get_cat <- function(data)
  {
  return(colnames(data[,sapply(data, is.factor)]))
```

```
  }
cat_cnames <- get_cat(train)

#Selecting numeric columns
get_num <- function(data)
  {
  return(colnames(data[,sapply(data, is.numeric)]))
  }
num_cnames <- get_num(train)

summary(train)
```

```
##        ID      Reason.for.absence Month.of.absence Day.of.the.week Seasons
##  3      :113   23     :145        3      : 87      2:147           1:151
##  28     : 71   28     :105        2      : 72      3:144           2:191
##  34     : 46   27     : 61        10     : 71      4:136           3:129
##  20     : 42   13     : 51        11     : 63      5:111           4:195
##  11     : 39   0      : 36        7      : 55      6:128
##  22     : 35   19     : 32        8      : 54
##  (Other):320   (Other):236        (Other):264
##  Transportation.expense Distance.from.Residence.to.Work  Service.time
##  Min.   :  0.0          Min.   : 5.00                   Min.   : 1.0
##  1st Qu.:179.0          1st Qu.:17.00                   1st Qu.: 9.0
##  Median :225.0          Median :26.00                   Median :13.0
##  Mean   :222.8          Mean   :30.37                   Mean   :12.7
##  3rd Qu.:260.0          3rd Qu.:50.00                   3rd Qu.:16.0
##  Max.   :388.0          Max.   :52.00                   Max.   :29.0
##
##       Age        Work.load.Average.day   Hit.target
Disciplinary.failure
##  Min.   :27.00   222,196: 35           Min.   : 81.00   0:630
##  1st Qu.:31.00   264,249: 33           1st Qu.: 92.00   1: 36
##  Median :37.00   343,253: 29           Median : 95.00
##  Mean   :36.41   265,017: 28           Mean   : 94.41
##  3rd Qu.:40.00   284,853: 25           3rd Qu.: 97.00
##  Max.   :58.00   308,593: 24           Max.   :100.00
##  NA's   :2       (Other):492           NA's   :1
##  Education Son     Social.drinker Social.smoker Pet         Weight
##  1:566     0:272   0:277          0:620         0:417   Min.   : 56.00
##  2: 37     1:201   1:389          1: 46         1:127   1st Qu.: 69.00
##  3: 62     2:137                                2: 86   Median : 83.00
##  4:  1     3: 14                                4: 28   Mean   : 79.21
##            4: 42                                5:  5   3rd Qu.: 89.00
##                                                 8:  3   Max.   :108.00
##                                                         NA's   :2
##      Height      Body.mass.index Absenteeism.time.in.hours
##  Min.   :163.0   Min.   :19.00   Min.   :  0.000
##  1st Qu.:169.0   1st Qu.:24.00   1st Qu.:  2.000
##  Median :170.0   Median :25.00   Median :  3.000
##  Mean   :171.9   Mean   :26.82   Mean   :  6.752
```

```
##  3rd Qu.:172.0   3rd Qu.:31.00   3rd Qu.:  8.000
##  Max.   :196.0   Max.   :38.00   Max.   :120.000
##
```

str(train)

```
## 'data.frame':    666 obs. of  21 variables:
##  $ ID                              : Factor w/ 34 levels
"1","2","3","5",..: 10 34 3 6 10 3 9 19 13 1 ...
##  $ Reason.for.absence              : Factor w/ 28 levels
"0","1","2","3",..: 26 1 23 8 23 23 22 23 20 22 ...
##  $ Month.of.absence                : Factor w/ 12 levels
"1","2","3","4",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ Day.of.the.week                 : Factor w/ 5 levels "2","3","4","5",..:
2 2 3 4 4 5 5 5 1 1 ...
##  $ Seasons                         : Factor w/ 4 levels "1","2","3","4": 1
1 1 1 1 1 1 1 1 ...
##  $ Transportation.expense          : int  289 118 179 279 289 179 361 260
155 235 ...
##  $ Distance.from.Residence.to.Work: int  36 13 51 5 36 51 52 50 12 11 ...
##  $ Service.time                    : int  13 18 18 14 13 18 3 11 14 14 ...
##  $ Age                             : num  33 50 38 39 33 38 28 36 34 37 ...
##  $ Work.load.Average.day           : Factor w/ 36 levels
"0","12","205,917",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ Hit.target                      : int  97 97 97 97 97 97 97 97 97 97 ...
##  $ Disciplinary.failure            : Factor w/ 2 levels "0","1": 1 2 1 1 1
1 1 1 1 1 ...
##  $ Education                       : Factor w/ 4 levels "1","2","3","4": 1
1 1 1 1 1 1 1 1 3 ...
##  $ Son                             : Factor w/ 5 levels "0","1","2","3",..:
3 2 1 3 3 1 2 5 3 2 ...
##  $ Social.drinker                  : Factor w/ 2 levels "0","1": 2 2 2 2 2
2 2 2 2 1 ...
##  $ Social.smoker                   : Factor w/ 2 levels "0","1": 1 1 1 2 1
1 1 1 1 1 ...
##  $ Pet                             : Factor w/ 6 levels "0","1","2","4",..:
2 1 1 1 2 1 4 1 1 2 ...
##  $ Weight                          : int  90 98 89 68 90 89 80 65 95 88 ...
##  $ Height                          : int  172 178 170 168 172 170 172 168
196 172 ...
##  $ Body.mass.index                 : int  30 31 31 24 30 31 27 23 25 29 ...
##  $ Absenteeism.time.in.hours       : int  4 0 2 4 2 2 8 4 40 8 ...
```

Missing Value

```
sapply(train, function(x) sum(is.na(x)))
```

```
##                              ID             Reason.for.absence
##                               0                              0
##               Month.of.absence                 Day.of.the.week
##                               0                              0
```

```
##                            Seasons          Transportation.expense
##                                  0                               0
## Distance.from.Residence.to.Work                    Service.time
##                                  0                               0
##                                Age           Work.load.Average.day
##                                  2                               0
##                         Hit.target            Disciplinary.failure
##                                  1                               0
##                          Education                             Son
##                                  0                               0
##                     Social.drinker                  Social.smoker
##                                  0                               0
##                                Pet                          Weight
##                                  0                               2
##                             Height                 Body.mass.index
##                                  0                               0
##       Absenteeism.time.in.hours
##                                  0
```

```
#ID related Variables
Dependent_ID <-
c("ID","Transportation.expense","Service.time","Age","Height",

"Distance.from.Residence.to.Work","Education","Son","Weight",
                    "Social.smoker","Social.drinker","Pet","Body.mass.index")
Dependent_ID_data <- train[,Dependent_ID]
Dependent_ID_data <- aggregate(. ~ ID, data = Dependent_ID_data,
                               FUN = function(e) c(x = mean(e)))
for (i in Dependent_ID)
  {
  for (j in (1:nrow(train)))
    {
    ID <- train[j,"ID"]
    if(is.na(train[j,i]))
      {
      train[j,i] <- Dependent_ID_data[ID,i]
      }
    }
  }


sapply(train, function(x) sum(is.na(x)))
```

```
##                                ID             Reason.for.absence
##                                 0                               0
##                 Month.of.absence                Day.of.the.week
##                                 0                               0
##                          Seasons          Transportation.expense
##                                 0                               0
## Distance.from.Residence.to.Work                    Service.time
```

```
##                                0                                0
##                              Age          Work.load.Average.day
##                                0                                0
##                       Hit.target           Disciplinary.failure
##                                1                                0
##                        Education                              Son
##                                0                                0
##                   Social.drinker                   Social.smoker
##                                0                                0
##                              Pet                           Weight
##                                0                                0
##                           Height                 Body.mass.index
##                                0                                0
##       Absenteeism.time.in.hours
##                                0
```

```
#Other Variables
train = knnImputation(train, k = 7)
sapply(train, function(x) sum(is.na(x)))
```

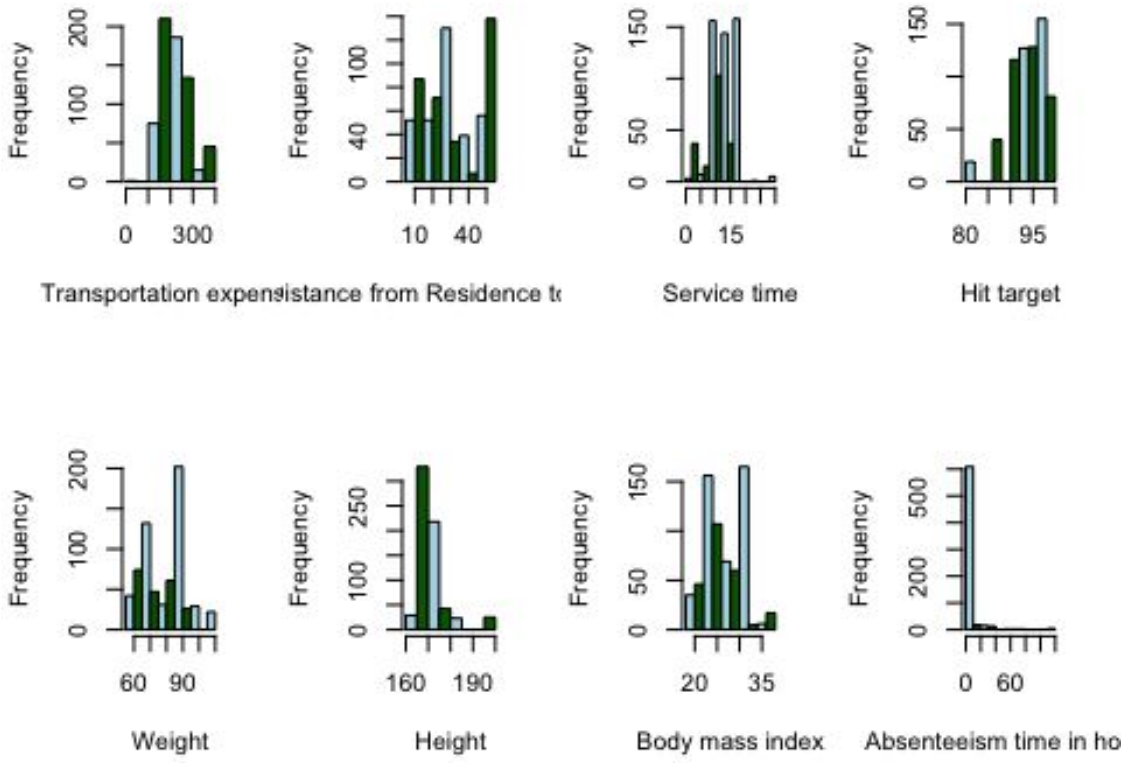
```
##                               ID             Reason.for.absence
##                                0                                0
##                 Month.of.absence                Day.of.the.week
##                                0                                0
##                          Seasons          Transportation.expense
##                                0                                0
## Distance.from.Residence.to.Work                     Service.time
##                                0                                0
##                              Age          Work.load.Average.day
##                                0                                0
##                       Hit.target           Disciplinary.failure
##                                0                                0
##                        Education                              Son
##                                0                                0
##                   Social.drinker                   Social.smoker
##                                0                                0
##                              Pet                           Weight
##                                0                                0
##                           Height                 Body.mass.index
##                                0                                0
##       Absenteeism.time.in.hours
##                                0
```

Outliers

Histograms

```
par(mfrow=c(2,4))
hist(train$Transportation.expense, xlab="Transportation expense", main=" ",
col=(c("lightblue","darkgreen")))
hist(train$Distance.from.Residence.to.Work, xlab="Distance from Residence to
```

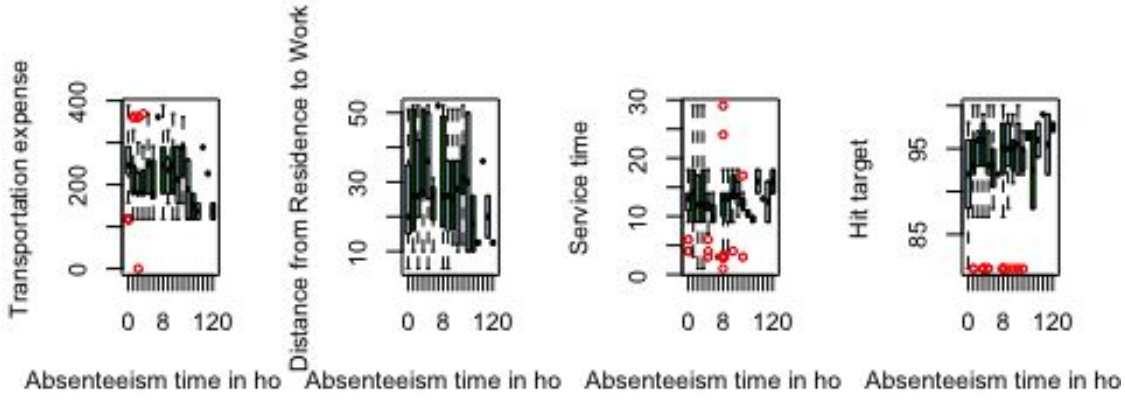
```
Work", main=" ", col=(c("lightblue","darkgreen")))
hist(train$Service.time, xlab="Service time", main=" ",
col=(c("lightblue","darkgreen")))
hist(train$Hit.target, xlab="Hit target", main=" ",
col=(c("lightblue","darkgreen")))
hist(train$Weight, xlab="Weight", main=" ", col=(c("lightblue","darkgreen")))
hist(train$Height, xlab="Height", main=" ", col=(c("lightblue","darkgreen")))
hist(train$Body.mass.index, xlab="Body mass index", main=" ",
col=(c("lightblue","darkgreen")))
hist(train$Absenteeism.time.in.hours, xlab="Absenteeism time in hours",
main=" ", col=(c("lightblue","darkgreen")))
```

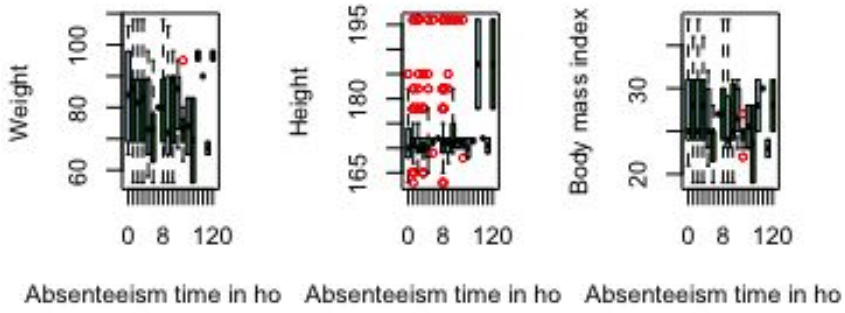


Boxplots

```
par(mfrow=c(2,4))
boxplot(train$Transportation.expense ~ train$Absenteeism.time.in.hours, data
= train, xlab = "Absenteeism time in hours", ylab="Transportation expense",
main=" ", col=(c("lightblue","darkgreen")), outcol="red")
boxplot(train$Distance.from.Residence.to.Work ~
train$Absenteeism.time.in.hours, data = train, xlab = "Absenteeism time in
hours", ylab="Distance from Residence to Work", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(train$Service.time ~ train$Absenteeism.time.in.hours, data = train,
```

```
xlab = "Absenteeism time in hours", ylab="Service time", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(train$Hit.target ~ train$Absenteeism.time.in.hours, data = train,
xlab = "Absenteeism time in hours", ylab="Hit target", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(train$Weight ~ train$Absenteeism.time.in.hours, data = train, xlab =
"Absenteeism time in hours", ylab="Weight", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(train$Height ~ train$Absenteeism.time.in.hours, data = train, xlab =
"Absenteeism time in hours", ylab="Height", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(train$Body.mass.index ~ train$Absenteeism.time.in.hours, data =
train, xlab = "Absenteeism time in hours", ylab="Body mass index", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
```






```
#Replacing all outliers with NA

num_cnames <- num_cnames[num_cnames != "Absenteeism.time.in.hours"]
for (i in Dependent_ID)
  {
  num_cnames <- num_cnames[num_cnames != i]
}
```

```
for(i in num_cnames)
  {
  val = train[,i][train[,i] %in% boxplot.stats(train[,i])$out]
  train[,i][train[,i] %in% val] = NA
}

sapply(train, function(x) sum(is.na(x)))
```

```
##                               ID              Reason.for.absence
##                                0                               0
##                 Month.of.absence                 Day.of.the.week
##                                0                               0
##                          Seasons          Transportation.expense
##                                0                               0
## Distance.from.Residence.to.Work                    Service.time
##                                0                               0
##                              Age           Work.load.Average.day
##                                0                               0
##                       Hit.target            Disciplinary.failure
##                               19                               0
##                        Education                             Son
##                                0                               0
##                   Social.drinker                   Social.smoker
##                                0                               0
##                              Pet                           Weight
##                                0                               0
##                           Height                 Body.mass.index
##                                0                               0
##       Absenteeism.time.in.hours
##                                0
```

```
# Impute NA

train <- knnImputation(train, k = 7)

sapply(train, function(x) sum(is.na(x)))
```

```
##                               ID              Reason.for.absence
##                                0                               0
##                 Month.of.absence                 Day.of.the.week
##                                0                               0
##                          Seasons          Transportation.expense
##                                0                               0
## Distance.from.Residence.to.Work                    Service.time
##                                0                               0
##                              Age           Work.load.Average.day
##                                0                               0
##                       Hit.target            Disciplinary.failure
##                                0                               0
##                        Education                             Son
```

```
##                                0                                0
##                  Social.drinker                   Social.smoker
##                                0                                0
##                             Pet                           Weight
##                                0                                0
##                          Height                  Body.mass.index
##                                0                                0
##       Absenteeism.time.in.hours
##                                0

train_na <- train
```

Testing Data Processing

Summary

```
##        ID        Reason.for.absence Month.of.absence Day.of.the.week
##  Min.   : 1.00   Min.   : 0.00      Min.   :0.00     Min.   :2.000
##  1st Qu.:14.00   1st Qu.:10.00      1st Qu.:5.00     1st Qu.:3.000
##  Median :22.00   Median :19.00      Median :5.00     Median :4.000
##  Mean   :21.11   Mean   :16.93      Mean   :5.27     Mean   :4.108
##  3rd Qu.:28.75   3rd Qu.:25.00      3rd Qu.:6.00     3rd Qu.:5.000
##  Max.   :36.00   Max.   :28.00      Max.   :7.00     Max.   :6.000
##     Seasons      Transportation.expense Distance.from.Residence.to.Work
##  Min.   :1.000   Min.   :118.0          Min.   :10.00
##  1st Qu.:1.250   1st Qu.:155.0          1st Qu.:13.00
##  Median :3.000   Median :207.0          Median :21.00
##  Mean   :2.473   Mean   :205.8          Mean   :22.97
##  3rd Qu.:3.000   3rd Qu.:235.0          3rd Qu.:26.00
##  Max.   :3.000   Max.   :378.0          Max.   :52.00
##   Service.time        Age        Work.load.Average.day   Hit.target
##  Min.   : 1.00   Min.   :28.00   237,656:32            Min.   :91.00
##  1st Qu.: 9.00   1st Qu.:30.25   246,288: 8            1st Qu.:93.00
##  Median :10.50   Median :36.50   264,604:12            Median :96.00
##  Mean   :11.24   Mean   :36.82   271,219: 3            Mean   :96.23
##  3rd Qu.:14.00   3rd Qu.:40.00   275,089:19            3rd Qu.:99.00
##  Max.   :24.00   Max.   :58.00                         Max.   :99.00
##  Disciplinary.failure   Education          Son         Social.drinker
##  Min.   :0.00000      Min.   :1.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000      1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000      Median :1.000   Median :1.0000   Median :0.0000
##  Mean   :0.05405      Mean   :1.703   Mean   :0.9324   Mean   :0.4189
##  3rd Qu.:0.00000      3rd Qu.:3.000   3rd Qu.:2.0000   3rd Qu.:1.0000
##  Max.   :1.00000      Max.   :4.000   Max.   :3.0000   Max.   :1.0000
##  Social.smoker         Pet            Weight           Height
##  Min.   :0.0000   Min.   :0.000   Min.   : 56.00   Min.   :163.0
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.: 68.00   1st Qu.:171.0
##  Median :0.0000   Median :0.000   Median : 75.00   Median :172.0
##  Mean   :0.1081   Mean   :1.243   Mean   : 77.38   Mean   :174.3
##  3rd Qu.:0.0000   3rd Qu.:2.000   3rd Qu.: 88.00   3rd Qu.:178.0
##  Max.   :1.0000   Max.   :8.000   Max.   :106.00   Max.   :196.0
```

```
##  Body.mass.index Absenteeism.time.in.hours
##  Min.   :19.00   Min.   :  0.000
##  1st Qu.:22.00   1st Qu.:  2.000
##  Median :25.00   Median :  3.000
##  Mean   :25.43   Mean   :  8.473
##  3rd Qu.:28.00   3rd Qu.:  8.000
##  Max.   :38.00   Max.   :120.000
```

Pre-processing

```
test <- test %>% mutate(
    Age = as.character(Age),
    Age = ifelse(Age == "R" , NA, Age),
    Age = ifelse(Age == "0" , NA, Age),
    Age = as.numeric(Age)
  )


  test$ID <- as.factor(test$ID)
  test$Reason.for.absence <- as.factor(test$Reason.for.absence)
  test$Month.of.absence <- as.factor(test$Month.of.absence)
  test$Day.of.the.week <- as.factor(test$Day.of.the.week)
  test$Seasons <- as.factor(test$Seasons)
  test$Disciplinary.failure <- as.factor(test$Disciplinary.failure)
  test$Education <- as.factor(test$Education)
  test$Son <- as.factor(test$Son)
  test$Social.drinker <- as.factor(test$Social.drinker)
  test$Social.smoker <- as.factor(test$Social.smoker)
  test$Pet <- as.factor(test$Pet)

#Selecting factor columns
get_cat <- function(data)
  {
  return(colnames(data[,sapply(data, is.factor)]))
  }
cat_cnames <- get_cat(test)

#Selecting numeric columns
get_num <- function(data)
  {
  return(colnames(data[,sapply(data, is.numeric)]))
  }
num_cnames <- get_num(test)

summary(test)
```

```
##        ID     Reason.for.absence Month.of.absence Day.of.the.week Seasons
##  22     :11   22     : 9         0: 3             2:14            1:19
##  34     : 9   19     : 8         4: 8             3:10            2: 1
##  36     : 6   27     : 8         5:32             4:20            3:54
```

```
##  28     : 5   0      : 7         6:19             5:14
##  12     : 4   10     : 7         7:12             6:16
##  14     : 4   28     : 7
##  (Other):35   (Other):28
##  Transportation.expense Distance.from.Residence.to.Work  Service.time
##  Min.   :118.0          Min.   :10.00                    Min.   : 1.00
##  1st Qu.:155.0          1st Qu.:13.00                    1st Qu.: 9.00
##  Median :207.0          Median :21.00                    Median :10.50
##  Mean   :205.8          Mean   :22.97                    Mean   :11.24
##  3rd Qu.:235.0          3rd Qu.:26.00                    3rd Qu.:14.00
##  Max.   :378.0          Max.   :52.00                    Max.   :24.00
##
##       Age        Work.load.Average.day   Hit.target
Disciplinary.failure
##  Min.   :28.00   237,656:32            Min.   :91.00   0:70
##  1st Qu.:30.25   246,288: 8            1st Qu.:93.00   1: 4
##  Median :36.50   264,604:12            Median :96.00
##  Mean   :36.82   271,219: 3            Mean   :96.23
##  3rd Qu.:40.00   275,089:19            3rd Qu.:99.00
##  Max.   :58.00                         Max.   :99.00
##
##  Education Son    Social.drinker Social.smoker Pet        Weight
##  1:45      0:26   0:43           0:66          0:43   Min.   : 56.00
##  2: 9      1:28   1:31           1: 8          1:11   1st Qu.: 68.00
##  3:17      2:19                                2:10   Median : 75.00
##  4: 3      3: 1                                4: 4   Mean   : 77.38
##                                                5: 1   3rd Qu.: 88.00
##                                                8: 5   Max.   :106.00
##
##      Height      Body.mass.index Absenteeism.time.in.hours
##  Min.   :163.0   Min.   :19.00   Min.   :  0.000
##  1st Qu.:171.0   1st Qu.:22.00   1st Qu.:  2.000
##  Median :172.0   Median :25.00   Median :  3.000
##  Mean   :174.3   Mean   :25.43   Mean   :  8.473
##  3rd Qu.:178.0   3rd Qu.:28.00   3rd Qu.:  8.000
##  Max.   :196.0   Max.   :38.00   Max.   :120.000
str(test)

## 'data.frame':    74 obs. of  21 variables:
##  $ ID                              : Factor w/ 26 levels
"1","2","4","5",..: 18 22 20 18 12 22 26 18 1 23 ...
##  $ Reason.for.absence              : Factor w/ 17 levels
"0","5","6","7",..: 16 11 10 16 17 11 9 16 12 11 ...
##  $ Month.of.absence                : Factor w/ 5 levels "0","4","5","6",..:
2 2 2 2 2 2 2 2 3 3 ...
##  $ Day.of.the.week                 : Factor w/ 5 levels "2","3","4","5",..:
5 1 2 5 2 4 4 5 1 3 ...
##  $ Seasons                         : Factor w/ 3 levels "1","2","3": 3 3 3
3 3 3 3 3 3 3 ...
##  $ Transportation.expense          : int  179 225 235 179 155 225 118 179
```

```
235 225 ...
##  $ Distance.from.Residence.to.Work: int  26 26 16 26 12 26 13 26 11 15 ...
##  $ Service.time                   : int  9 9 8 9 14 9 18 9 14 15 ...
##  $ Age                            : num  30 28 32 30 34 28 50 30 37 41 ...
##  $ Work.load.Average.day          : Factor w/ 5 levels
"237,656","246,288",..: 2 2 2 2 2 2 2 2 1 1 ...
##  $ Hit.target                     : int  91 91 91 91 91 91 91 91 99 99 ...
##  $ Disciplinary.failure           : Factor w/ 2 levels "0","1": 1 1 1 1 1
1 1 1 1 1 ...
##  $ Education                      : Factor w/ 4 levels "1","2","3","4": 3
1 3 3 1 1 1 3 3 4 ...
##  $ Son                            : Factor w/ 4 levels "0","1","2","3": 1
2 1 1 3 2 2 1 2 3 ...
##  $ Social.drinker                 : Factor w/ 2 levels "0","1": 1 1 1 1 2
1 2 1 1 2 ...
##  $ Social.smoker                  : Factor w/ 2 levels "0","1": 1 1 1 1 1
1 1 1 1 1 ...
##  $ Pet                            : Factor w/ 6 levels "0","1","2","4",..:
1 3 1 1 1 3 1 1 2 3 ...
##  $ Weight                         : int  56 69 75 56 95 69 98 56 88 94 ...
##  $ Height                         : int  171 169 178 171 196 169 178 171
172 182 ...
##  $ Body.mass.index                : int  19 24 25 19 25 24 31 19 29 28 ...
##  $ Absenteeism.time.in.hours      : int  2 8 3 2 4 8 2 2 8 3 ...
```

Missing Value

```
sapply(test, function(x) sum(is.na(x)))
```

```
##                               ID              Reason.for.absence
##                                0                               0
##                Month.of.absence                 Day.of.the.week
##                                0                               0
##                         Seasons          Transportation.expense
##                                0                               0
## Distance.from.Residence.to.Work                    Service.time
##                                0                               0
##                             Age           Work.load.Average.day
##                                0                               0
##                      Hit.target            Disciplinary.failure
##                                0                               0
##                       Education                             Son
##                                0                               0
##                  Social.drinker                   Social.smoker
##                                0                               0
##                             Pet                          Weight
##                                0                               0
##                          Height                 Body.mass.index
##                                0                               0
##       Absenteeism.time.in.hours
##                                0
```

```r
#ID related Variables
Dependent_ID <-
c("ID","Transportation.expense","Service.time","Age","Height",

"Distance.from.Residence.to.Work","Education","Son","Weight",
                    "Social.smoker","Social.drinker","Pet","Body.mass.index")
Dependent_ID_data <- test[,Dependent_ID]
Dependent_ID_data <- aggregate(. ~ ID, data = Dependent_ID_data,
                               FUN = function(e) c(x = mean(e)))
for (i in Dependent_ID)
  {
  for (j in (1:nrow(test)))
    {
    ID <- test[j,"ID"]
    if(is.na(test[j,i]))
      {
      test[j,i] <- Dependent_ID_data[ID,i]
      }
    }
  }


sapply(test, function(x) sum(is.na(x)))
```

```
##                               ID             Reason.for.absence
##                                0                              0
##                 Month.of.absence                Day.of.the.week
##                                0                              0
##                          Seasons          Transportation.expense
##                                0                              0
## Distance.from.Residence.to.Work                    Service.time
##                                0                              0
##                              Age           Work.load.Average.day
##                                0                              0
##                       Hit.target            Disciplinary.failure
##                                0                              0
##                        Education                            Son
##                                0                              0
##                   Social.drinker                   Social.smoker
##                                0                              0
##                              Pet                         Weight
##                                0                              0
##                           Height                 Body.mass.index
##                                0                              0
##       Absenteeism.time.in.hours
##                                0
```

```r
#Other Variables
test = knnImputation(test, k = 7)
```

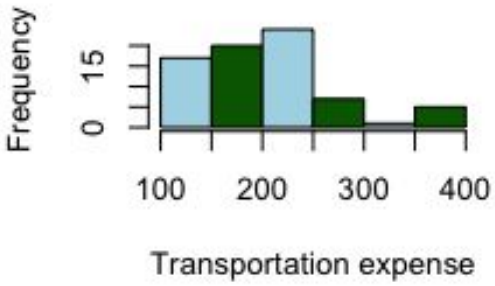
```
## Warning in knnImputation(test, k = 7): No case has missing values.
Stopping as
sapply(test, function(x) sum(is.na(x)))
```

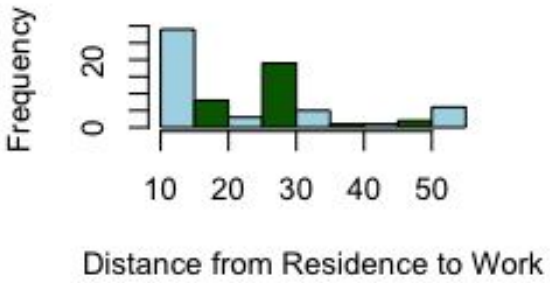
```
##                                ID              Reason.for.absence
##                                 0                               0
##                 Month.of.absence                 Day.of.the.week
##                                 0                               0
##                          Seasons          Transportation.expense
##                                 0                               0
## Distance.from.Residence.to.Work                    Service.time
##                                 0                               0
##                              Age           Work.load.Average.day
##                                 0                               0
##                      Hit.target            Disciplinary.failure
##                                 0                               0
##                       Education                             Son
##                                 0                               0
##                  Social.drinker                  Social.smoker
##                                 0                               0
##                             Pet                          Weight
##                                 0                               0
##                          Height                 Body.mass.index
##                                 0                               0
##       Absenteeism.time.in.hours
##                                 0
```

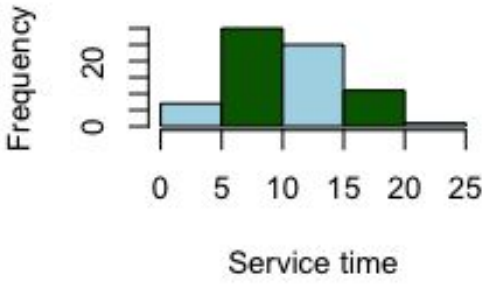
Outliers

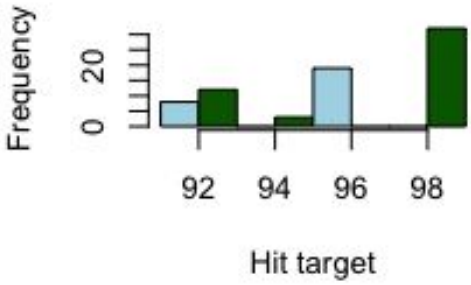
Histograms

```
par(mfrow=c(2,2))
hist(test$Transportation.expense, xlab="Transportation expense", main=" ",
col=(c("lightblue","darkgreen")))
hist(test$Distance.from.Residence.to.Work, xlab="Distance from Residence to
Work", main=" ", col=(c("lightblue","darkgreen")))
hist(test$Service.time, xlab="Service time", main=" ",
col=(c("lightblue","darkgreen")))
hist(test$Hit.target, xlab="Hit target", main=" ",
col=(c("lightblue","darkgreen")))
```

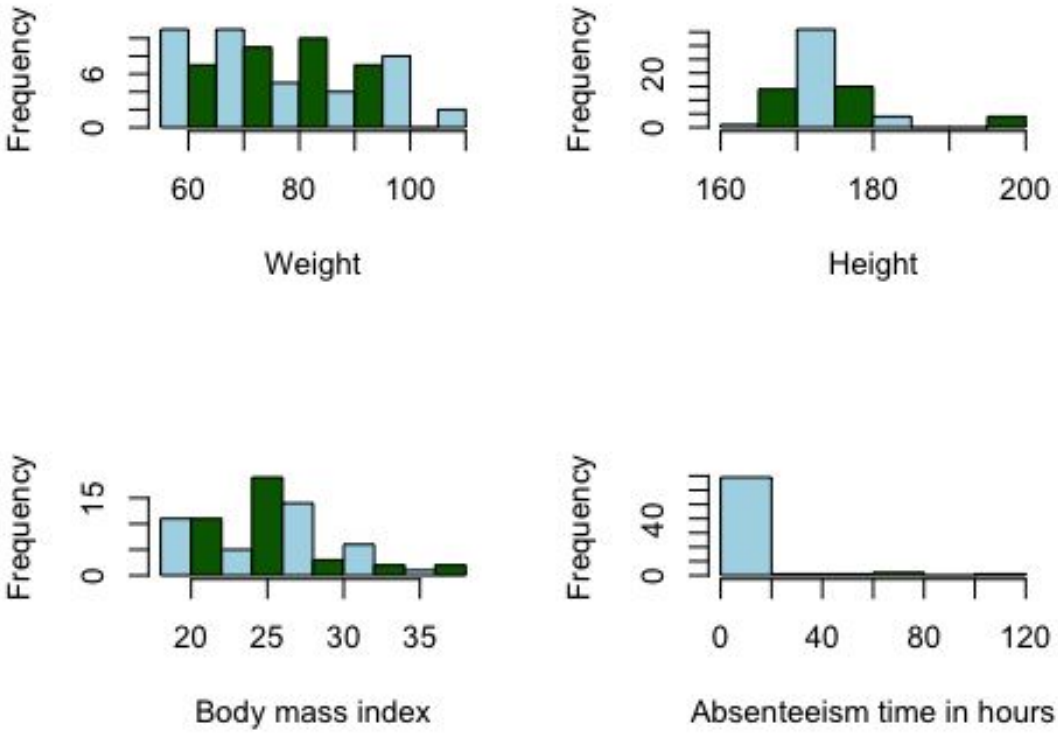






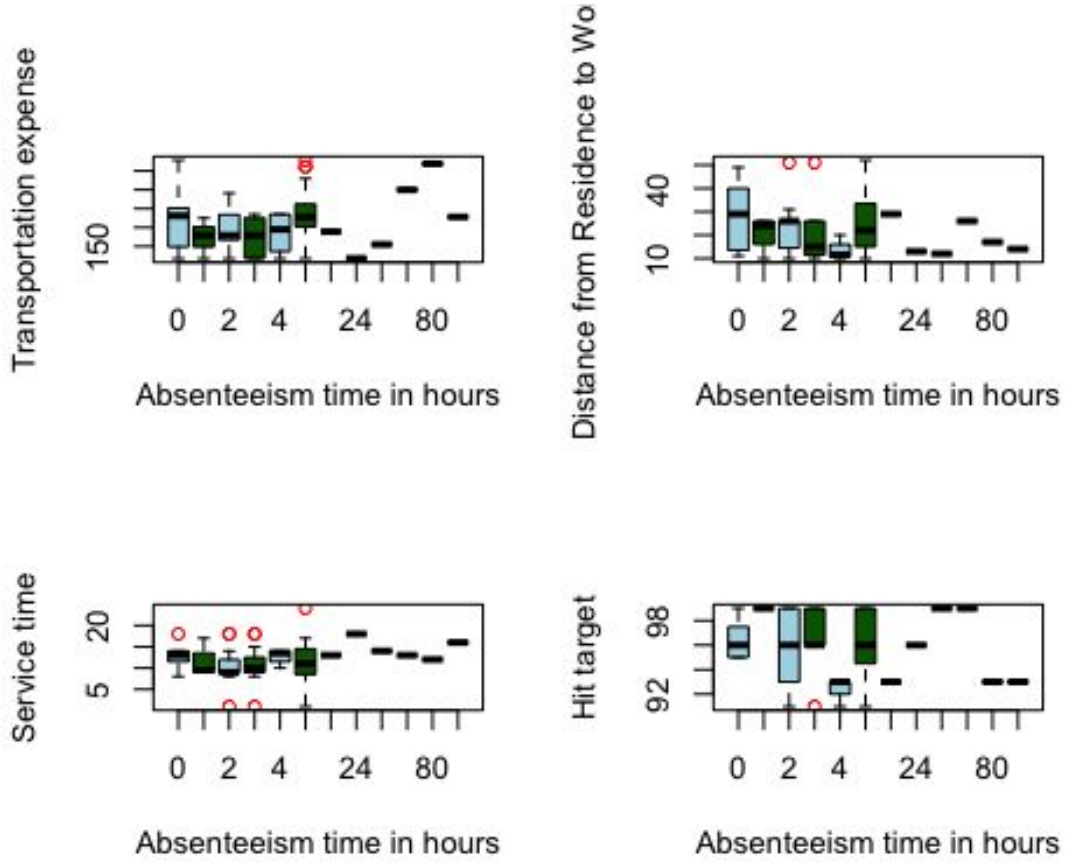



```r
hist(test$Weight, xlab="Weight", main=" ", col=(c("lightblue","darkgreen")))
hist(test$Height, xlab="Height", main=" ", col=(c("lightblue","darkgreen")))
hist(test$Body.mass.index, xlab="Body mass index", main=" ",
col=(c("lightblue","darkgreen")))
hist(test$Absenteeism.time.in.hours, xlab="Absenteeism time in hours", main="
", col=(c("lightblue","darkgreen")))
```

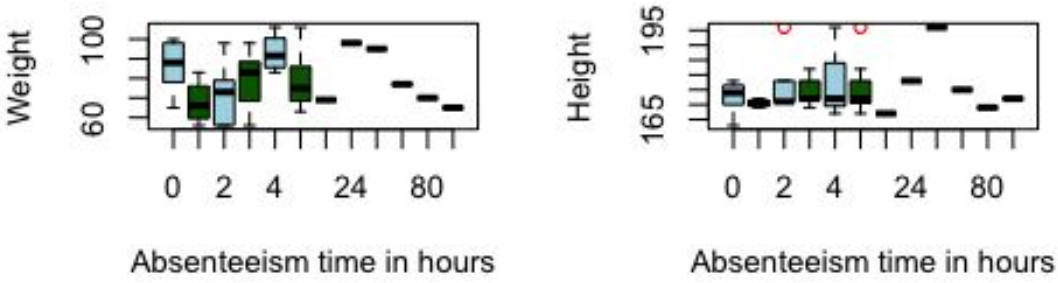


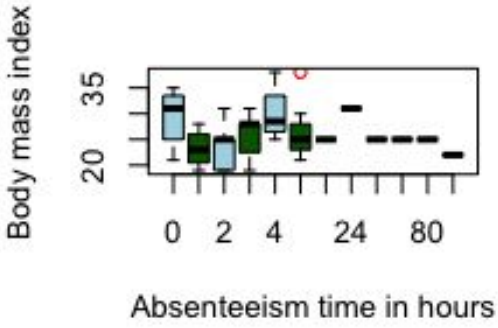
Boxplots

```
par(mfrow=c(2,2))
boxplot(test$Transportation.expense ~ test$Absenteeism.time.in.hours, data =
test, xlab = "Absenteeism time in hours", ylab="Transportation expense",
main=" ", col=(c("lightblue","darkgreen")), outcol="red")
boxplot(test$Distance.from.Residence.to.Work ~
test$Absenteeism.time.in.hours, data = test, xlab = "Absenteeism time in
hours", ylab="Distance from Residence to Work", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(test$Service.time ~ test$Absenteeism.time.in.hours, data = test, xlab
= "Absenteeism time in hours", ylab="Service time", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(test$Hit.target ~ test$Absenteeism.time.in.hours, data = test, xlab =
"Absenteeism time in hours", ylab="Hit target", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
```




```
boxplot(test$Weight ~ test$Absenteeism.time.in.hours, data = test, xlab =
"Absenteeism time in hours", ylab="Weight", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(test$Height ~ test$Absenteeism.time.in.hours, data = test, xlab =
"Absenteeism time in hours", ylab="Height", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
boxplot(test$Body.mass.index ~ test$Absenteeism.time.in.hours, data = test,
xlab = "Absenteeism time in hours", ylab="Body mass index", main=" ",
col=(c("lightblue","darkgreen")), outcol="red")
```






```
#Replacing all outliers with NA

num_cnames <- num_cnames[num_cnames != "Absenteeism.time.in.hours"]
for (i in Dependent_ID)
  {
  num_cnames <- num_cnames[num_cnames != i]
}

for(i in num_cnames)
  {
  val = test[,i][test[,i] %in% boxplot.stats(test[,i])$out]
  test[,i][test[,i] %in% val] = NA
}

sapply(test, function(x) sum(is.na(x)))
```

```
##                              ID             Reason.for.absence
##                               0                              0
##                Month.of.absence                Day.of.the.week
##                               0                              0
##                         Seasons          Transportation.expense
##                               0                              0
## Distance.from.Residence.to.Work                   Service.time
```

```
##                                0                                0
##                              Age          Work.load.Average.day
##                                0                                0
##                       Hit.target           Disciplinary.failure
##                                0                                0
##                        Education                              Son
##                                0                                0
##                   Social.drinker                   Social.smoker
##                                0                                0
##                              Pet                           Weight
##                                0                                0
##                           Height                 Body.mass.index
##                                0                                0
##       Absenteeism.time.in.hours
##                                0
```

```
# Impute NA

test <- knnImputation(test, k = 7)
```

```
## Warning in knnImputation(test, k = 7): No case has missing values.
Stopping as
sapply(test, function(x) sum(is.na(x)))
```

```
##                              ID             Reason.for.absence
##                               0                              0
##                Month.of.absence                Day.of.the.week
##                               0                              0
##                         Seasons          Transportation.expense
##                               0                              0
## Distance.from.Residence.to.Work                   Service.time
##                               0                              0
##                             Age          Work.load.Average.day
##                               0                              0
##                      Hit.target           Disciplinary.failure
##                               0                              0
##                       Education                            Son
##                               0                              0
##                  Social.drinker                   Social.smoker
##                               0                              0
##                             Pet                           Weight
##                               0                              0
##                          Height                 Body.mass.index
##                               0                              0
##       Absenteeism.time.in.hours
##                               0
```

```
test$Absenteeism.time.in.hours <- cut(
```

```
  test$Absenteeism.time.in.hours,
  breaks = c(0,0.1,6, Inf),
  labels = c("Group 1", "Group 2", "Group 3" ),
  right  = FALSE
)



train$Absenteeism.time.in.hours <- cut(
  train$Absenteeism.time.in.hours,
  breaks = c(0,0.1,6, Inf),
  labels = c("Group 1", "Group 2", "Group 3" ),
  right  = FALSE
)
library(BBmisc)
train <- normalize(train)
test <- normalize(test)
```

Task 1: Model Creation & Evaluation

Kmeans analysis

Start the k-Means analysis using the variable "nc" for the number of clusters

```
library(NbClust)
set.seed(12345)
nc <- NbClust(train[,-20], min.nc=2, max.nc = 15, method = "kmeans")
print(table(nc$Best.n[1,]))
##
##  0  1  2  3  4  5  8  9 10 13 14 15
```

```
##  2  1  5  2  2  1  1  4  1  1  4  2
barplot(table(nc$Best.n[1,]), xlab = "Number of Clusters", ylab = "Number of Criteria", main =
"Number of Clusters Chosen by 26 Criteria")


#Conduct the k-Means analysis using the best number of clusters
set.seed(1234)
n= 2# As per the above bar graph and wssplot
fit.km <- kmeans(train[,-20], n, nstart=25)
print(fit.km$size)
## [1] 299 367
print(fit.km$centers)
##   Reason.for.absence Month.of.absence Day.of.the.week             Seasons
## 1          0.1240294       -0.06738425    -0.11307165 -0.04188201
## 2         -0.1010484        0.05489889     0.09212105  0.03412185
##   Transportation.expense Distance.from.Residence.to.Work Service.time
## 1             -0.5467714                      0.09875146 0.6128307
## 2              0.4454622                     -0.08045418  -0.4992817
##        Age Work.load.Average.day  Hit.target Disciplinary.failure  Education
## 1  0.4397901          -0.04666825  0.03202358                -0.01717600 -0.3987424
## 2 -0.3583031          0.03802127 -0.02609006        0.01399352  0.3248610
##        Son Social.drinker Social.smoker        Pet     Weight           Height
## 1 -0.2452322      0.3346780          -0.2590008 -0.3743103  0.8952156  0.1855672
## 2  0.1997941          -0.2726668      0.2110115  0.3049558 -0.7293446 -0.1511841
##   Body.mass.index
## 1       0.8603477
```

```
## 2     -0.7009372

#print(aggregate(test[-20], by=list(cluster=fit.km$cluster), mean))

#Use a confusion or truth table to evaluate how well the k-Means analysis performed

ct.km <- table(train$Absenteeism.time.in.hours, fit.km$cluster)

#print(ct.km)


Accuracy <- sum(diag(ct.km))/sum(ct.km)*100


Accuracy

## [1] 32.88288
```

Knn

```
set.seed(12345)

library(class)

knnmodel <- knn(train[,-20], test[,-20], cl=train$Absenteeism.time.in.hours,
k = 25)

table(knnmodel)

## knnmodel

## Group 1 Group 2 Group 3

##    4     51      19

library(gmodels)

CrossTable(test$Absenteeism.time.in.hours, knnmodel, prop.chisq=F, prop.c=F,
prop.r=F, dnn=c("Actual", "Predicted (KNN)"))

##

##

##    Cell Contents

## |-------------------------|
```

```
## |                   N |
## |        N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  74
##
##
##              | Predicted (KNN)
##    Actual |   Group 1 |   Group 2 |   Group 3 | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##    Group 1 |         4 |         0 |         3 |         7 |
##              |       0.054 |     0.000 |     0.041 |           |
## -------------|-----------|-----------|-----------|-----------|
##    Group 2 |         0 |         35 |        3 |         38 |
##              |       0.000 |     0.473 |     0.041 |           |
## -------------|-----------|-----------|-----------|-----------|
##    Group 3 |         0 |         16 |        13 |        29 |
##              |       0.000 |     0.216 |     0.176 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |       4 |         51 |        19 |        74 |
## -------------|-----------|-----------|-----------|-----------|
##
##
p <- table(knnmodel, test$Absenteeism.time.in.hours)
Accuracy <- sum(diag(p))/sum(p)*100
```

```
Accuracy
## [1] 70.27027
prec.knn = 52/68
prec.knn
## [1] 0.7647059
```

Decision trees

```
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
## Loading required package: tibble
## Loading required package: bitops
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
##
## Attaching package: 'rattle'
## The following object is masked from 'package:randomForest':
##
##     importance
Dec_model = rpart(Absenteeism.time.in.hours~ .,data = train)

library(gmodels)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(ggplot2)

fancyRpartPlot(Dec_model)

#Now use the predict() function to see how well the model works

pred <- predict(Dec_model, test, type="class")

CrossTable(test$Absenteeism.time.in.hours, pred, prop.chisq=F, prop.c=F,
prop.r=F, dnn=c("Actual", "Predicted (RT)"))

##

##

##    Cell Contents

## |-------------------------|

## |                   N |

## |        N / Table Total |

## |-------------------------|

##

##

## Total Observations in Table:  74

##

##

##              | Predicted (RT)

##    Actual |   Group 1 |   Group 2 |   Group 3 | Row Total |

## -------------|-----------|-----------|-----------|-----------|

##    Group 1 |         7 |         0 |         0 |         7 |

##              |       0.095 |     0.000 |     0.000 |           |

## -------------|-----------|-----------|-----------|-----------|

##    Group 2 |         0 |         24 |        14 |        38 |
```

```
##               |       0.000 |     0.324 |     0.189 |           |
## -------------|-----------|-----------|-----------|-----------|
##    Group 3 |         0 |         5 |         24 |        29 |
##               |       0.000 |     0.068 |     0.324 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |       7 |         29 |        38 |        74 |
## -------------|-----------|-----------|-----------|-----------|
##
## Accuracy
## [1] 74.32432
auc.dtr = auc(roc.multi)
auc.dtr
## Multi-class area under the curve: 0.9099
rs <- roc.multi[['rocs']]
plot.roc(rs[[1]])
sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))
##                      [,1]        [,2]
## percent            FALSE         FALSE
## sensitivities      Numeric,4   Numeric,3
## specificities      Numeric,4   Numeric,3
## thresholds         Numeric,4   Numeric,3
## direction          "<"         "<"
## cases              Numeric,29  Numeric,29
## controls           Numeric,7   Numeric,38
## fun.sesp           ?             ?
## call               Expression  Expression
```

```
## original.predictor Numeric,74  Numeric,74

## original.response  factor,74   factor,74

## predictor          Numeric,36  Numeric,67

## response           factor,36   factor,67

## levels             Character,2 Character,2

ggplot(data=test, aes(x=Absenteeism.time.in.hours, y=74, fill=factor(pred)))
+ geom_bar(stat="identity")+

      scale_fill_manual(values = c("light yellow", "light pink","light
blue"),

                  labels = c("Group 1", "Group 2", "Group 3"),

                  name = "Predicted ") +   xlab("Actual") +

  ylab("Predicted") + ggtitle("Decision Trees")
```

Random forest Model

```
 test <- rbind(train[1, ] , test)

test <- test[-1,]


library(e1071)

##

## Attaching package: 'e1071'

## The following object is masked from 'package:raster':

##

##    interpolate

library(randomForest)

library(Metrics)

random_model <- randomForest(Absenteeism.time.in.hours ~ . , data= train)

random_model
```

```
##
## Call:
##  randomForest(formula = Absenteeism.time.in.hours ~ ., data = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##          OOB estimate of  error rate: 16.67%
## Confusion matrix:
##         Group 1 Group 2 Group 3 class.error
## Group 1      36      0     0   0.0000000
## Group 2       0      328   59   0.1524548
## Group 3       0      52    191   0.2139918
absent_pred <- predict(random_model, test)
p <- table(absent_pred, test$Absenteeism.time.in.hours)
Accuracy <- sum(diag(p))/sum(p)*100
Accuracy
## [1] 75.67568
hist(random_model$importance)
print(postResample(absent_pred,test$Absenteeism.time.in.hours))
##  Accuracy      Kappa
## 0.7567568 0.5792798
library(pROC)
pred.rf = as.numeric(predict(random_model, test, type = 'response'))
roc.multi = multiclass.roc(test$Absenteeism.time.in.hours, pred.rf)
## Setting direction: controls < cases
```

```
## Setting direction: controls < cases

## Setting direction: controls < cases

auc.rf = auc(roc.multi)

auc.rf

## Multi-class area under the curve: 0.7843

rs <- roc.multi[['rocs']]

plot.roc(rs[[1]])

sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))

##                       [,1]        [,2]

## percent             FALSE    FALSE

## sensitivities       Numeric,4   Numeric,3

## specificities       Numeric,4   Numeric,3

## thresholds          Numeric,4   Numeric,3

## direction           "<"         "<"

## cases               Numeric,29  Numeric,29

## controls            Numeric,7   Numeric,38

## fun.sesp            ?             ?

## call                Expression  Expression

## original.predictor Numeric,74  Numeric,74

## original.response  factor,74   factor,74

## predictor           Numeric,36  Numeric,67

## response            factor,36   factor,67

## levels              Character,2 Character,2

plot(random_model)

ggplot(data=test, aes(x=Absenteeism.time.in.hours, y=74,
fill=factor(absent_pred))) + geom_bar(stat="identity")+
```

```
      scale_fill_manual(values = c("light yellow", "light pink","light
blue"),

                  labels = c("Group 1", "Group 2", "Group 3"),

                  name = "Predicted ") +   xlab("Actual") +

  ylab("Predicted") + ggtitle("Random Forest ")
```

Random Forest with feature Selection

```
features_rm <-
c('Reason.for.absence','Work.load.Average.day','Transportation.expense','Day.
of.the.week','Month.of.absence','Age','Service.time','Absenteeism.time.in.hou
rs')



trainf <- train[features_rm]

testf <- test[features_rm]



random_model2 <- randomForest(Absenteeism.time.in.hours ~ . , data= trainf)







absent_pred2 <- predict(random_model2, testf)

p2 <- table(absent_pred2, testf$Absenteeism.time.in.hours)

Accuracy2 <- sum(diag(p2))/sum(p2)*100

Accuracy2

## [1] 78.37838

#importance(random_model2)



plot(random_model2,main="Random Forest with Feature selection")

pred.rfs = as.numeric(predict(random_model2, testf, type = 'response'))
```

```
roc.multi = multiclass.roc(testf$Absenteeism.time.in.hours, pred.rfs)

## Setting direction: controls < cases

## Setting direction: controls < cases

## Setting direction: controls < cases

auc.rfs = auc(roc.multi)

auc.rfs

## Multi-class area under the curve: 0.9285

CrossTable(testf$Absenteeism.time.in.hours, absent_pred2, prop.chisq=F,
prop.c=F, prop.r=F, dnn=c("Actual", "Predicted (KNN)"))

##

##

##    Cell Contents

## |-------------------------|

## |                       N |

## |        N / Table Total |

## |-------------------------|

##

##

## Total Observations in Table:  74

##

##

##                | Predicted (KNN)

##    Actual |   Group 1 |   Group 2 |   Group 3 | Row Total |

## -------------|-----------|-----------|-----------|-----------|

##    Group 1 |         7 |         0 |         0 |         7 |

##                |     0.095 |     0.000 |     0.000 |           |

## -------------|-----------|-----------|-----------|-----------|
```

```
##    Group 2 |         0 |         23 |        15 |        38 |
##                |     0.000 |     0.311 |     0.203 |           |
## -------------|-----------|-----------|-----------|-----------|
##    Group 3 |         0 |         1 |         28 |        29 |
##                |     0.000 |     0.014 |      0.378 |          |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |         7 |       24 |        43 |        74 |
## -------------|-----------|-----------|-----------|-----------|
##
##
rs <- roc.multi[['rocs']]
plot.roc(rs[[1]])
sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))
##                      [,1]        [,2]
## percent             FALSE         FALSE
## sensitivities       Numeric,4   Numeric,3
## specificities       Numeric,4   Numeric,3
## thresholds          Numeric,4   Numeric,3
## direction           "<"         "<"
## cases               Numeric,29  Numeric,29
## controls            Numeric,7   Numeric,38
## fun.sesp            ?             ?
## call             Expression  Expression
## original.predictor Numeric,74  Numeric,74
## original.response  factor,74   factor,74
## predictor           Numeric,36  Numeric,67
```

```
## response           factor,36   factor,67

## levels             Character,2 Character,2

ggplot(data=testf, aes(x=Absenteeism.time.in.hours, y=74,
fill=factor(absent_pred2))) + geom_bar(stat="identity")+

      scale_fill_manual(values = c("light yellow", "light pink","light
blue"),

                  labels = c("Group 1", "Group 2", "Group 3"),

                  name = "Predicted ") +   xlab("Actual") +

  ylab("Predicted") + ggtitle("Random Forest with Feature Selection")

#Naive Bayes

library(naivebayes)

## naivebayes 0.9.7 loaded

naive_model <- naive_bayes(Absenteeism.time.in.hours ~ . , data= train)

## Warning: naive_bayes(): Feature Reason.for.absence - zero probabilities
are

## present. Consider Laplace smoothing.

## Warning: naive_bayes(): Feature Disciplinary.failure - zero probabilities
are

## present. Consider Laplace smoothing.

## Warning: naive_bayes(): Feature Education - zero probabilities are
present.

## Consider Laplace smoothing.

## Warning: naive_bayes(): Feature Pet - zero probabilities are present.
Consider

## Laplace smoothing.

naive_pred <- predict(naive_model, test)

## Warning: predict.naive_bayes(): more features in the newdata are provided
as

## there are probability tables in the object. Calculation is performed based
on
```

```
## features to be found in the tables.

p <- table(naive_pred, test$Absenteeism.time.in.hours)



CrossTable(test$Absenteeism.time.in.hours, naive_pred, prop.chisq = FALSE,
prop.c = FALSE, prop.r = FALSE, dnn = c('Actual  ', 'Predicted '))

##

##

##    Cell Contents

## |-------------------------|

## |                       N |

## |        N / Table Total |

## |-------------------------|

##

##

## Total Observations in Table:  74

##

##

##                | Predicted

##    Actual   |   Group 1 |   Group 2 |   Group 3 | Row Total |

## -------------|-----------|-----------|-----------|-----------|

##    Group 1 |         4 |         3 |         0 |         7 |

##                |     0.054 |     0.041 |     0.000 |           |

## -------------|-----------|-----------|-----------|-----------|

##    Group 2 |         0 |         25 |        13 |        38 |

##                |     0.000 |     0.338 |     0.176 |           |

## -------------|-----------|-----------|-----------|-----------|

##    Group 3 |         0 |         3 |         26 |        29 |
```

```
##                 |     0.000 |     0.041 |     0.351 |           |
## ----------------|-----------|-----------|-----------|-----------|
## Column Total |         4 |       31 |        39 |        74 |
## ----------------|-----------|-----------|-----------|-----------|
##
##
Accuracy <- sum(diag(p))/sum(p)*100


Accuracy
## [1] 74.32432
pred.nb = as.numeric(predict(naive_model, test))
## Warning: predict.naive_bayes(): more features in the newdata are provided
as
## there are probability tables in the object. Calculation is performed based
on
## features to be found in the tables.
roc.multi = multiclass.roc(test$Absenteeism.time.in.hours, pred.nb)
## Setting direction: controls < cases
## Setting direction: controls < cases
## Setting direction: controls < cases
auc.nb = auc(roc.multi)
auc.nb
## Multi-class area under the curve: 0.8714
rs <- roc.multi[['rocs']]
plot.roc(rs[[1]])
sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))
##                      [,1]       [,2]
```

```
## percent             FALSE         FALSE

## sensitivities      Numeric,4   Numeric,3

## specificities      Numeric,4   Numeric,3

## thresholds         Numeric,4   Numeric,3

## direction          "<"         "<"

## cases              Numeric,29  Numeric,29

## controls           Numeric,7   Numeric,38

## fun.sesp           ?           ?

## call               Expression  Expression

## original.predictor Numeric,74  Numeric,74

## original.response  factor,74   factor,74

## predictor          Numeric,36  Numeric,67

## response           factor,36   factor,67

## levels             Character,2 Character,2

 ggplot(data=test, aes(x=Absenteeism.time.in.hours, y=74,
fill=factor(naive_pred))) + geom_bar(stat="identity")+

      scale_fill_manual(values = c("light yellow", "light pink","light
blue"),

                  labels = c("Group 1", "Group 2", "Group 3"),

                  name = "Predicted ") +   xlab("Actual") +

  ylab("Predicted") + ggtitle("Naive Bayes ")

 naive_model

##

## ================================= Naive Bayes
==================================

##

##  Call:
```

```
## naive_bayes.formula(formula = Absenteeism.time.in.hours ~ .,

##    data = train)

##

##
-------------------------------------------------------------------------
----

##

## Laplace smoothing: 0

##

##
-------------------------------------------------------------------------
----

##

##  A priori probabilities:

##

##    Group 1     Group 2    Group 3

## 0.05405405 0.58108108 0.36486486

##

##
-------------------------------------------------------------------------
----

##

##  Tables:

##

##
-------------------------------------------------------------------------
----

##  ::: Reason.for.absence (Categorical)

##
-------------------------------------------------------------------------
----
```

```
##
## Reason.for.absence     Group 1   Group 2     Group 3
##                  0  1.000000000 0.000000000 0.000000000
##                  1  0.000000000 0.005167959 0.057613169
##                  2  0.000000000 0.000000000 0.004115226
##                  3  0.000000000 0.000000000 0.004115226
##                  4  0.000000000 0.002583979 0.004115226
##                  5  0.000000000 0.000000000 0.008230453
##                  6  0.000000000 0.000000000 0.020576132
##                  7  0.000000000 0.018087855 0.024691358
##                  8  0.000000000 0.002583979 0.012345679
##                  9  0.000000000 0.000000000 0.016460905
##                 10 0.000000000 0.007751938 0.061728395
##                 11 0.000000000 0.025839793 0.057613169
##                 12 0.000000000 0.010335917 0.016460905
##                13 0.000000000 0.036175711 0.152263374
##                 14 0.000000000 0.023255814 0.028806584
##                 15 0.000000000 0.000000000 0.008230453
##                 16 0.000000000 0.002583979 0.000000000
##                 17 0.000000000 0.000000000 0.004115226
##                 18 0.000000000 0.010335917 0.069958848
##                 19 0.000000000 0.010335917 0.115226337
##                 21 0.000000000 0.005167959 0.016460905
##                 22 0.000000000 0.007751938 0.106995885
##                  23 0.000000000 0.341085271 0.053497942
##                 24 0.000000000 0.000000000 0.012345679
```

```
##                25 0.000000000 0.064599483 0.016460905

##                26 0.000000000 0.018087855 0.094650206

##                27 0.000000000 0.157622739 0.000000000

##                28 0.000000000 0.250645995 0.032921811

##

##
-------------------------------------------------------------------------
----

##  ::: Month.of.absence (Gaussian)

##
-------------------------------------------------------------------------
----

##

## Month.of.absence     Group 1     Group 2     Group 3

##             mean  0.47505926 -0.05629000  0.01926789

##          sd    0.75734966  1.04062008  0.94800394

##

##
-------------------------------------------------------------------------
----

##  ::: Day.of.the.week (Categorical)

##
-------------------------------------------------------------------------
----

##

## Day.of.the.week   Group 1   Group 2   Group 3

##                2 0.1111111 0.1912145 0.2839506

##                3 0.3055556 0.2015504 0.2263374

##                4 0.2500000 0.1912145 0.2181070

##                5 0.1944444 0.1912145 0.1234568
```

```
##                6 0.1388889 0.2248062 0.1481481

##

##
-------------------------------------------------------------------------
----

##  ::: Seasons (Categorical)

##
-------------------------------------------------------------------------
----

##

## Seasons  Group 1     Group 2     Group 3

##    1 0.13888889 0.21963824 0.25102881

##    2 0.02777778 0.33074935 0.25514403

##    3 0.25000000 0.16537468 0.23045267

##    4 0.58333333 0.28423773 0.26337449

##

##
-------------------------------------------------------------------------
----

##  ::: Transportation.expense (Gaussian)

##
-------------------------------------------------------------------------
----

##

## Transportation.expense    Group 1      Group 2     Group 3

##                   mean  0.4384380 -0.2311083  0.3031076

##                   sd 1.1587585  0.8468031  1.0971374

##

##
-------------------------------------------------------------------------
----
```

```
##

## # ... and 14 more tables

##

##
--------------------------------------------------------------------------
----
```

Naive Bayes with classifier

```
library(corrplot)

## corrplot 0.84 loaded

library(caret)



#Randomize the data

absent_rand <- train_nu[order(runif(nrow(train_nu))),]

#Scale the data

absentDataScaled <- scale(absent_rand, center=TRUE, scale = TRUE)

#Compute the correlation matrix (note that this does not include the class
variable)

m <- cor(absentDataScaled)



highlycor <- findCorrelation(m,.5)

highlycor

## [1] 10  4

train

## # A tibble: 666 x 20

##    Reason.for.abse… Month.of.absence Day.of.the.week Seasons
Transportation.…

##    <fct>                       <dbl> <fct>            <fct>
<dbl>
```

```
##  1 26                              0.156 3                 1
0.994

##  2 0                            0.156 3              1                 -1.58

##  3 23                              0.156 4                 1
-0.664

##  4 7                               0.156 5                 1
0.843

##  5 23                           0.156 5              1                 0.994

##  6 23                              0.156 6                 1
-0.664

##  7 22                              0.156 6                 1
2.08

##  8 23                              0.156 6                 1
0.557

##  9 19                              0.156 2                 1
-1.03

## 10 22                              0.156 2                 1
0.180

## # … with 656 more rows, and 15 more variables:

## #   Distance.from.Residence.to.Work <dbl>, Service.time <dbl>, Age <dbl>,

## #   Work.load.Average.day <dbl>, Hit.target <dbl>, Disciplinary.failure
<fct>,

## #   Education <fct>, Son <fct>, Social.drinker <fct>, Social.smoker <fct>,

## #   Pet <fct>, Weight <dbl>, Height <dbl>, Body.mass.index <dbl>,

## #   Absenteeism.time.in.hours <fct>

filteredData <- train[, -c(7,19)]

filteredtest <- test[, -c(7,19)]

nb_model <- naive_bayes(Absenteeism.time.in.hours ~ ., data=filteredData)

## Warning: naive_bayes(): Feature Reason.for.absence - zero probabilities
are

## present. Consider Laplace smoothing.
```

```
## Warning: naive_bayes(): Feature Disciplinary.failure - zero probabilities
are

## present. Consider Laplace smoothing.

## Warning: naive_bayes(): Feature Education - zero probabilities are
present.

## Consider Laplace smoothing.

## Warning: naive_bayes(): Feature Pet - zero probabilities are present.
Consider

## Laplace smoothing.

nb_model

##

## ================================== Naive Bayes
==================================

##

##  Call:

## naive_bayes.formula(formula = Absenteeism.time.in.hours ~ .,

##    data = filteredData)

##

##
-------------------------------------------------------------------------
----

##

## Laplace smoothing: 0

##

##
-------------------------------------------------------------------------
----

##

##  A priori probabilities:

##
```

```
##    Group 1     Group 2    Group 3

## 0.05405405 0.58108108 0.36486486

##

##
-------------------------------------------------------------------------
----

##

##  Tables:

##

##
-------------------------------------------------------------------------
----

##  ::: Reason.for.absence (Categorical)

##
-------------------------------------------------------------------------
----

##

## Reason.for.absence     Group 1   Group 2     Group 3

##                  0  1.000000000 0.000000000 0.000000000

##                  1  0.000000000 0.005167959 0.057613169

##                  2  0.000000000 0.000000000 0.004115226

##                  3  0.000000000 0.000000000 0.004115226

##                  4  0.000000000 0.002583979 0.004115226

##                  5  0.000000000 0.000000000 0.008230453

##                    6  0.000000000 0.000000000 0.020576132

##                  7  0.000000000 0.018087855 0.024691358

##                  8  0.000000000 0.002583979 0.012345679

##                  9  0.000000000 0.000000000 0.016460905

##                 10 0.000000000 0.007751938 0.061728395
```

```
##                11 0.000000000 0.025839793 0.057613169
##                12 0.000000000 0.010335917 0.016460905
##                13 0.000000000 0.036175711 0.152263374
##                14 0.000000000 0.023255814 0.028806584
##                15 0.000000000 0.000000000 0.008230453
##                16 0.000000000 0.002583979 0.000000000
##                17 0.000000000 0.000000000 0.004115226
##                18 0.000000000 0.010335917 0.069958848
##                19 0.000000000 0.010335917 0.115226337
##                21 0.000000000 0.005167959 0.016460905
##                22 0.000000000 0.007751938 0.106995885
##                23 0.000000000 0.341085271 0.053497942
##                24 0.000000000 0.000000000 0.012345679
##                25 0.000000000 0.064599483 0.016460905
##                26 0.000000000 0.018087855 0.094650206
##                27 0.000000000 0.157622739 0.000000000
##                28 0.000000000 0.250645995 0.032921811
##
##
--------------------------------------------------------------------------
----
##  ::: Month.of.absence (Gaussian)
##
--------------------------------------------------------------------------
----
##
## Month.of.absence     Group 1     Group 2     Group 3
##             mean  0.47505926 -0.05629000  0.01926789
```

```
##          sd    0.75734966  1.04062008  0.94800394

##

##
------------------------------------------------------------------------
----

##  ::: Day.of.the.week (Categorical)

##
------------------------------------------------------------------------
----

##

## Day.of.the.week   Group 1   Group 2   Group 3

##               2 0.1111111 0.1912145 0.2839506

##               3 0.3055556 0.2015504 0.2263374

##               4 0.2500000 0.1912145 0.2181070

##               5 0.1944444 0.1912145 0.1234568

##               6 0.1388889 0.2248062 0.1481481

##

##
------------------------------------------------------------------------
----

##  ::: Seasons (Categorical)

##
------------------------------------------------------------------------
----

##

## Seasons  Group 1     Group 2     Group 3

##    1 0.13888889 0.21963824 0.25102881

##    2 0.02777778 0.33074935 0.25514403

##    3 0.25000000 0.16537468 0.23045267

##    4 0.58333333 0.28423773 0.26337449
```

```
##

##
-------------------------------------------------------------------------
----

##  ::: Transportation.expense (Gaussian)

##
-------------------------------------------------------------------------
----

##

## Transportation.expense    Group 1      Group 2     Group 3

##                    mean  0.4384380 -0.2311083  0.3031076

##                    sd 1.1587585  0.8468031  1.0971374

##

##
-------------------------------------------------------------------------
----

##

## # ... and 12 more tables

##

##
-------------------------------------------------------------------------
----

filteredTestPred <- predict(nb_model, newdata = filteredtest)

## Warning: predict.naive_bayes(): more features in the newdata are provided
as

## there are probability tables in the object. Calculation is performed based
on

## features to be found in the tables.

p <- table(filteredTestPred, filteredtest$Absenteeism.time.in.hours)

Accuracy <- sum(diag(p))/sum(p)*100
```

```
Accuracy

## [1] 74.32432

pred.nbc = as.numeric(predict(nb_model, filteredtest))

## Warning: predict.naive_bayes(): more features in the newdata are provided
as

## there are probability tables in the object. Calculation is performed based
on

## features to be found in the tables.

roc.multi = multiclass.roc(filteredtest$Absenteeism.time.in.hours, pred.nbc)

## Setting direction: controls < cases

## Setting direction: controls < cases

## Setting direction: controls < cases

auc.nbc = auc(roc.multi)

auc.nbc

## Multi-class area under the curve: 0.8714

rs <- roc.multi[['rocs']]

plot.roc(rs[[1]])

sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))

##                       [,1]        [,2]

## percent             FALSE         FALSE

## sensitivities       Numeric,4   Numeric,3

## specificities       Numeric,4   Numeric,3

## thresholds          Numeric,4   Numeric,3

## direction           "<"         "<"

## cases               Numeric,29  Numeric,29

## controls            Numeric,7   Numeric,38

## fun.sesp            ?             ?
```

```
## call               Expression  Expression

## original.predictor Numeric,74  Numeric,74

## original.response  factor,74   factor,74

## predictor          Numeric,36  Numeric,67

## response           factor,36   factor,67

## levels             Character,2 Character,2

# BMI and Service time




ggplot(data=filteredtest, aes(x=Absenteeism.time.in.hours, y=74,
fill=factor(filteredTestPred))) + geom_bar(stat="identity")+

      scale_fill_manual(values = c("light yellow", "light pink","light
blue"),

                  labels = c("Group 1", "Group 2", "Group 3"),

                  name = "Predicted ") +   xlab("Actual") +

  ylab("Predicted") + ggtitle("Naive Bayes with Classifier")
```

SVM with vanilladot

```
library(kernlab)

##

## Attaching package: 'kernlab'

## The following object is masked from 'package:psych':

##

##    alpha

## The following objects are masked from 'package:raster':

##

##    buffer, rotated
```

```
## The following object is masked from 'package:ggplot2':

##

##    alpha

svm_classifier <- ksvm(Absenteeism.time.in.hours ~ ., data=train,kernel
="vanilladot")

##  Setting default kernel parameters

svm_classifier

## Support Vector Machine object of class "ksvm"

##

## SV type: C-svc  (classification)

##  parameter : cost C = 1

##

## Linear (vanilla) kernel function.

##

## Number of Support Vectors : 330

##

## Objective Function Value : -0.9739 -0.9803 -194.8695

## Training error : 0.135135

pred_svm <- predict(svm_classifier, test)


p<-table(pred_svm, test$Absenteeism.time.in.hours)

Accuracy <- sum(diag(p))/sum(p)*100


Accuracy

## [1] 72.97297

pred.svm = as.numeric(predict(svm_classifier, test))

roc.multi = multiclass.roc(test$Absenteeism.time.in.hours, pred.svm)
```

```
## Setting direction: controls < cases

## Setting direction: controls < cases

## Setting direction: controls < cases

auc.svm = auc(roc.multi)

auc.svm

## Multi-class area under the curve: 0.9028

rs <- roc.multi[['rocs']]

plot.roc(rs[[1]])

sapply(2:length(rs),function(i) lines.roc(rs[[i]],col=i))

##                       [,1]        [,2]

## percent             FALSE         FALSE

## sensitivities       Numeric,4   Numeric,3

## specificities       Numeric,4   Numeric,3

## thresholds          Numeric,4   Numeric,3

## direction           "<"         "<"

## cases               Numeric,29  Numeric,29

## controls            Numeric,7   Numeric,38

## fun.sesp            ?             ?

## call                Expression  Expression

## original.predictor Numeric,74  Numeric,74

## original.response  factor,74   factor,74

## predictor           Numeric,36  Numeric,67

## response            factor,36   factor,67

## levels              Character,2 Character,2

ggplot(data=test, aes(x=Absenteeism.time.in.hours, y=74,
fill=factor(pred_svm))) + geom_bar(stat="identity")+
```

```
      scale_fill_manual(values = c("light yellow", "light pink","light
blue"),
                  labels = c("Group 1", "Group 2", "Group 3"),
                  name = "Predicted ") +   xlab("Actual") +
  ylab("Predicted") + ggtitle("SVM with vanilladot ")
```

Task 2: Model Creation & Evaluation

Assumptions

```
train_nu <- train[,-c(1:4,11:16)]
###Assumptions for linear regression

modeltime = lm(Absenteeism.time.in.hours~.,train)
summary(modeltime)
##
## Call:
## lm(formula = Absenteeism.time.in.hours ~ ., data = train)
##
## Residuals:
##    Min   1Q  Median        3Q    Max
## -34.456  -3.525  -0.427   1.830  97.533
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.585e+01  1.014e+02   0.354  0.72372
```

```
## Reason.for.absence1              1.038e+01  3.600e+00   2.882  0.00410 **
## Reason.for.absence2              2.292e+01  1.184e+01   1.936  0.05333 .
## Reason.for.absence3              8.620e+00  1.179e+01   0.731  0.46484
## Reason.for.absence4              8.625e+00  8.657e+00   0.996  0.31952
## Reason.for.absence5              1.054e+01  8.549e+00   1.233  0.21813
## Reason.for.absence6              7.565e+00  5.575e+00   1.357  0.17526
## Reason.for.absence7              1.084e+01  3.831e+00   2.829  0.00482 **
## Reason.for.absence8              8.052e+00  6.233e+00   1.292  0.19692
## Reason.for.absence9              4.321e+01  6.153e+00   7.023 5.95e-12 ***
## Reason.for.absence10             1.052e+01  3.468e+00   3.033  0.00252 **
## Reason.for.absence11             1.303e+01  3.204e+00   4.065 5.44e-05 ***
## Reason.for.absence12               2.445e+01  4.685e+00   5.218 2.49e-07
***
## Reason.for.absence13             1.473e+01  2.638e+00   5.585 3.55e-08 ***
## Reason.for.absence14             9.937e+00  3.523e+00   2.820  0.00496 **
## Reason.for.absence15             7.441e+00  8.411e+00   0.885  0.37666
## Reason.for.absence16             1.266e-01  1.200e+01   0.011  0.99159
## Reason.for.absence17             8.141e+00  1.194e+01   0.682  0.49557
## Reason.for.absence18             9.825e+00  3.384e+00   2.904  0.00382 **
## Reason.for.absence19             1.888e+01  2.967e+00   6.363 3.95e-10 ***
## Reason.for.absence21             6.747e+00  5.261e+00   1.282  0.20020
## Reason.for.absence22             7.313e+00  3.169e+00   2.308  0.02136 *
## Reason.for.absence23             3.030e+00  2.300e+00   1.317  0.18832
## Reason.for.absence24             5.823e+00  7.037e+00   0.827  0.40828
## Reason.for.absence25             4.077e+00  3.162e+00   1.290  0.19766
## Reason.for.absence26             6.955e+00  2.947e+00   2.360  0.01860 *
## Reason.for.absence27             4.924e+00  2.931e+00   1.680  0.09353 .
```

```
## Reason.for.absence28              3.121e+00  2.444e+00   1.277  0.20207
## Month.of.absence2                 6.648e-01  2.323e+00   0.286  0.77483
## Month.of.absence3                 2.505e+00  2.422e+00   1.034  0.30153
## Month.of.absence4                 3.481e+00  3.622e+00   0.961  0.33699
## Month.of.absence5                -8.967e-01  3.706e+00  -0.242  0.80891
## Month.of.absence6                 6.285e+00  3.631e+00   1.731  0.08394 .
## Month.of.absence7                 7.882e+00  4.570e+00   1.725  0.08504 .
## Month.of.absence8                 6.774e+00  4.696e+00   1.443  0.14965
## Month.of.absence9                 6.722e+00  4.460e+00   1.507  0.13230
## Month.of.absence10                5.098e+00  4.622e+00   1.103  0.27043
## Month.of.absence11                5.386e+00  4.321e+00   1.246  0.21311
## Month.of.absence12                5.996e+00  3.868e+00   1.550  0.12161
## Day.of.the.week3                 -7.225e-01  1.400e+00  -0.516  0.60607
## Day.of.the.week4                 -6.105e-01  1.411e+00  -0.433  0.66544
## Day.of.the.week5                 -2.860e+00  1.503e+00  -1.903  0.05746 .
## Day.of.the.week6                 -2.118e+00  1.470e+00  -1.441  0.15022
## Seasons2                          5.753e+00  3.718e+00   1.547  0.12227
## Seasons3                          5.876e+00  3.577e+00   1.643  0.10095
## Seasons4                          3.554e+00  2.944e+00   1.207  0.22789
## Transportation.expense           -6.612e-03  1.586e-02  -0.417  0.67687
## Distance.from.Residence.to.Work   9.821e-02  9.807e-02   1.001  0.31700
## Service.time                     -1.512e-01  3.865e-01  -0.391  0.69582
## Age                               1.761e-01  2.277e-01   0.773  0.43957
## Work.load.Average.day            -1.698e-05  1.444e-05  -1.176  0.23994
## Hit.target                        1.829e-01  2.246e-01   0.814  0.41575
## Disciplinary.failure1                    NA         NA      NA       NA
```

```
## Education2                      -5.473e-01  3.777e+00  -0.145  0.88484
## Education3                      -1.943e+00  2.961e+00  -0.656  0.51204
## Education4                      -2.209e+00  1.252e+01  -0.176  0.86001
## Son1                            4.732e+00  2.836e+00   1.669  0.09571 .
## Son2                            6.438e+00  2.679e+00   2.403  0.01657 *
## Son3                            5.849e+00  5.197e+00   1.125  0.26090
## Son4                            -1.694e-01  3.031e+00  -0.056  0.95545
## Social.drinker1                 -6.090e-01  2.832e+00  -0.215  0.82980
## Social.smoker1                  -4.846e+00  2.699e+00  -1.796  0.07306 .
## Pet1                            -2.750e+00  3.148e+00  -0.874  0.38269
## Pet2                            -4.234e+00  4.713e+00  -0.898  0.36940
## Pet4                            -5.543e+00  6.385e+00  -0.868  0.38564
## Pet5                            3.724e+00  7.261e+00   0.513  0.60819
## Pet8                            -2.316e+01  9.064e+00  -2.555  0.01088 *
## Weight                          6.193e-01  6.777e-01   0.914  0.36114
## Height                          -3.188e-01  6.015e-01  -0.530  0.59637
## Body.mass.index                 -2.109e+00  1.930e+00  -1.093  0.27489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.23 on 597 degrees of freedom
## Multiple R-squared:  0.2948, Adjusted R-squared:  0.2145
## F-statistic:  3.67 on 68 and 597 DF,  p-value: < 2.2e-16
#Additivity
correl = cor(train_nu)
symnum(correl)
```

```
##                                 T D S Ag W. H. Wg Hg B A.
## Transportation.expense          1
## Distance.from.Residence.to.Work   1
## Service.time                    .   1
## Age                             , 1
## Work.load.Average.day                 1
## Hit.target                                  1
## Weight                          . .         1
## Height                          .           .  1
## Body.mass.index                 . .         *     1
## Absenteeism.time.in.hours                               1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
#High Correlation between Weight and BMI so removing BMI from analysis,Age
and Service time also have high correlation but it is less than 0.7
#linearity
standardized = rstudent(modeltime)
fitted = scale(modeltime$fitted.values)
qqnorm(standardized)
abline(0,1)
# yes , it met the assumption of linearity  as the dots are close to line


#Normality
hist(standardized)
library(moments)
##
## Attaching package: 'moments'
```

```
## The following objects are masked from 'package:e1071':
##
##    kurtosis, moment, skewness
skewness(train_nu)
##          Transportation.expense Distance.from.Residence.to.Work
##                     0.379713854                      0.237504505
##                    Service.time                              Age
##                     0.021930349                      0.638157586
##           Work.load.Average.day                    Hit.target
##                     0.853811816                     -0.438380776
##                          Weight                           Height
##                     0.008970067                      2.689673868
##                 Body.mass.index       Absenteeism.time.in.hours
##                     0.292766864                      5.889442141
kurtosis(train_nu)
##          Transportation.expense Distance.from.Residence.to.Work
##                       2.667529                       1.688401
##                    Service.time                              Age
##                       3.674035                       3.372462
##           Work.load.Average.day                    Hit.target
##                       3.316162                       2.652522
##                          Weight                           Height
##                       2.095362                       10.997373
##                 Body.mass.index       Absenteeism.time.in.hours
##                       2.640967                       44.895648
#Kurtosis ,Height , Weight , Disciplinaryfailure , pet , Social Smoker,
Work.load.Average.day, education
```

```
shapiro.test(modeltime$residuals)

##

##  Shapiro-Wilk normality test

##

## data:  modeltime$residuals

## W = 0.58909, p-value < 2.2e-16

# Height and WorkloadAverage day have high values , so we will do boots strap
to check the results if it is ithin coinfidence interval or we will use
transformation



#does not met the assumption of normality as the p value is less 0.05



#Heterocadasticity



plot(fitted, standardized)

abline(0,0)

abline(v = 0)
```

Stepwise Selection before PCA

```
model <-
lm(Absenteeism.time.in.hours~.-Height-Body.mass.index-Disciplinary.failure,tr
ain)

summary(model)

model2.bw = step(model, direction = 'backward')

absent_pred <- predict(model2.bw, test[,-c(18,19,11)])

p <- table(absent_pred, test[,-c(18,19,11)]$Absenteeism.time.in.hours)

Accuracy <- sum(diag(p))/sum(p)*100

Accuracy
```

```
## [1] 1.351351

AIC(model2.bw)

## [1] 5148.543

print(postResample(pred = absent_pred, obs = test$Absenteeism.time.in.hours))

##        RMSE    Rsquared          MAE

## 17.28958321  0.09750446  7.20859550

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

lines(absent_pred,col="red")

title(main= "Linear Regression")
```

Random forest before pca

```
trainnew <- train[,-c(18,19,11)]

testnew <- test[,-c(18,19,11)]

x <-
c("Hit.target","Work.load.Average.day","Weight","Age","Day.of.the.week","Mont
h.of.absence","Reason.for.absence","Absenteeism.time.in.hours")



train2 <- train[x]

test2 <- test[x]

random_model2 <- randomForest(Absenteeism.time.in.hours ~., data= train2)

random_model2

##

## Call:

##  randomForest(formula = Absenteeism.time.in.hours ~ ., data = train2)

##                Type of random forest: regression

##                      Number of trees: 500

## No. of variables tried at each split: 2

##
```

```
##           Mean of squared residuals: 142.2494

##                     % Var explained: 11.2

absent_pred2 <- predict(random_model2, test2)

p <- table(absent_pred2, test$Absenteeism.time.in.hours)

Accuracy <- sum(diag(p))/sum(p)*100

Accuracy

## [1] 1.351351

print(postResample(pred = absent_pred2, obs =
test$Absenteeism.time.in.hours))

##      RMSE  Rsquared        MAE

## 17.061488  0.166972  6.661527

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

lines(absent_pred2,col="red")

title(main= "Random forest with feature selection")
```

PCA

```
library(fastDummies)


trainnew <- train[,-c(18,19,11)]

testnew <- test[,-c(18,19,11)]


total <- rbind(trainnew,testnew)

dummynew <- fastDummies::dummy_cols(total,remove_selected_columns=TRUE)


len <- nrow(train)

train <- dummynew[1:666,]

test <- dummynew[667:740,]
```

```
train <- normalize(train)

test <- normalize(test)


pcatrain <- train[,-8]

pcatest <- test[,-8]

#principal component analysis

prin_comp <- prcomp(pcatrain, center = TRUE)

prin_comp$rotation

plot(cumsum(prop_varex), xlab = "Principal Component",

            ylab = "Cumulative Proportion of Variance Explained",

            type = "b")

train.data <- data.frame(Absenteeism.time.in.hours =
train$Absenteeism.time.in.hours, prin_comp$x)


train.data <- train.data[,1:60]


test.data <- predict(prin_comp, newdata = test)

test.data <- as.data.frame(test.data)




test.data <- test.data[,1:60]
```

Random Forest

```
#Train the model using training data

rf_model = randomForest(Absenteeism.time.in.hours~., data = train.data,
ntrees = 10000)

rf_model
```

```
##

## Call:

##  randomForest(formula = Absenteeism.time.in.hours ~ ., data = train.data,
      ntrees = 10000)

##                Type of random forest: regression

##                      Number of trees: 500

## No. of variables tried at each split: 19

##

##           Mean of squared residuals: 1.005977

##                     % Var explained: -0.75

#Predict the test cases

rf_predictions = predict(rf_model,test.data)




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

print(postResample(pred = rf_predictions, obs =
test$Absenteeism.time.in.hours))

##       RMSE   Rsquared      MAE

## 0.95156479 0.08334156 0.42430598

#Plot a graph for actual vs predicted values

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

lines(rf_predictions,col="red")
```

Linear regression

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



model2.bw = step(lr_model, direction = 'backward')
```

```
absent_pred <- predict(model2.bw, test.data)

p <- table(absent_pred, test$Absenteeism.time.in.hours)

Accuracy <- sum(diag(p))/sum(p)*100

Accuracy

## [1] 0

AIC(model2.bw)

## [1] 1767.864

print(postResample(pred = absent_pred, obs = test$Absenteeism.time.in.hours))

##    RMSE  Rsquared       MAE

## 0.9030284 0.1734792 0.4407038

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

lines(absent_pred,col="red")

title(main= "Linear Regression")
```