**INDEX**

|  |  |
| --- | --- |
| **SL.NO** | **TITLE** |
| **1** | PROJECT OBJECTIVE |
| **2** | DATASET SELECTION |
| **3** | DATA PREPROCESSING |
| **4** | EXPLORATORY DATA ANALYSIS |
| **5** | DATA VISUALIZATION |
| **6**  **7**  **8**  **9** | FEATURE SELECTION  MODEL DEVELOPMENT  MODEL COMPARISION  CONCLUSION |

# **PROJECT OBJECTIVE:**

The objective of our project was to predict student’s performance in a particular subject given different attributes.

In the chosen dataset the attributes used are as follows:

* school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)
* sex - student's sex (binary: "F" - female or "M" - male)
* age - student's age (numeric: from 15 to 22)
* address - student's home address type (binary: "U" - urban or "R" - rural)
* famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)
* Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)
* Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
* Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
* Mjob - mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at\_home" or "other")
* Fjob - father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at\_home" or "other")
* reason - reason to choose this school (nominal: close to "home", school "reputation", "course" preference or "other")
* guardian - student's guardian (nominal: "mother", "father" or "other")
* traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
* studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
* failures - number of past class failures (numeric: n if 1<=n<3, else 4)
* schoolsup - extra educational support (binary: yes or no)
* famsup - family educational support (binary: yes or no)
* paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
* activities - extra-curricular activities (binary: yes or no)
* nursery - attended nursery school (binary: yes or no)
* higher - wants to take higher education (binary: yes or no)
* internet - Internet access at home (binary: yes or no)
* romantic - with a romantic relationship (binary: yes or no)
* famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
* freetime - free time after school (numeric: from 1 - very low to 5 - very high)
* goout - going out with friends (numeric: from 1 - very low to 5 - very high)
* Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
* Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
* health - current health status (numeric: from 1 - very bad to 5 - very good)
* absences - number of school absences (numeric: from 0 to 93.

These grades are related with the course subject, Math or Portuguese:

* G1 - first period grade (numeric: from 0 to 20)
* G2 - second period grade (numeric: from 0 to 20)
* G3 - final grade (numeric: from 0 to 20, output target)

# 

# **DATASET SELECTION:**

The data selected is called ‘STUDENT PERFORMANCE’

Which was collected by Paulo Cortez and Alice Silva.

This data approaches student achievement in secondary education of two Portuguese schools in two subjects – maths and Portuguese language. The above-mentioned attributes were considered to play a major role in the grades obtained by the students in the two subjects. We chose this dataset for the variety of the attributes and to see how a small change in a student’s environment would affect his or her grade.

**DATA PREPROCESSING AND EXPLORATORY DATA ANALYSIS (EDA):**

Exploratory Data Analysis (EDA) is the process of examining and understanding a dataset to uncover patterns, relationships, and insights. It involves descriptive statistics, data visualization, and data cleaning techniques to gain a deeper understanding of the data's structure, distribution, and characteristics. EDA helps to identify outliers, missing values, and potential data issues, as well as formulating hypotheses and guiding further analysis. It is an important initial step in data analysis that enables researchers and analysts to make informed decisions and derive meaningful insights from the data.

The below steps were followed to do exploratory data analysis on the data set:

STEP 1: Loading the required packages.

We have mostly used the built in packages. The package used

is the ‘tidyverse’ package. The following command was used to load the package:

library(tidyverse)

STEP 2: Importing the dataset.

The dataset contains two different data frames – one for Mathematics subject and the other for Portuguese.

Both the datasets were imported using the following command:

d1=read.table("student-mat.csv",sep=";",header=TRUE)

d2=read.table("student-por.csv",sep=";",header=TRUE)

d1 here refers to the mathematics data frame while d2 refers

to the Portuguese data frame. The dataset in the file were separated by ‘;’, hence the ‘sep = “;”’ function is used. The first row of the dataset is considered the header line.

STEP 3: Initial data inspection.

In this step we do a surface level inspection of the data which includes things like inspecting different attributes, differentiating the attributes as quantitative or categorical etc. To do this the following commands were used:

head(d1)

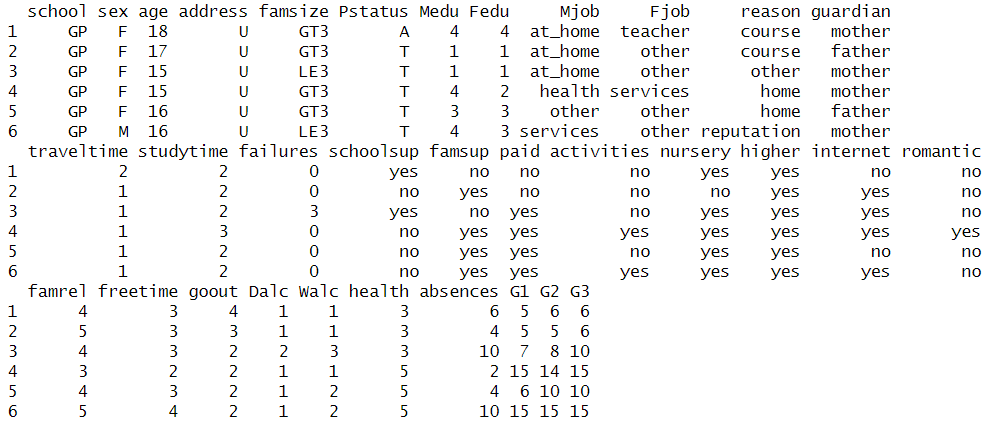
head(d2)

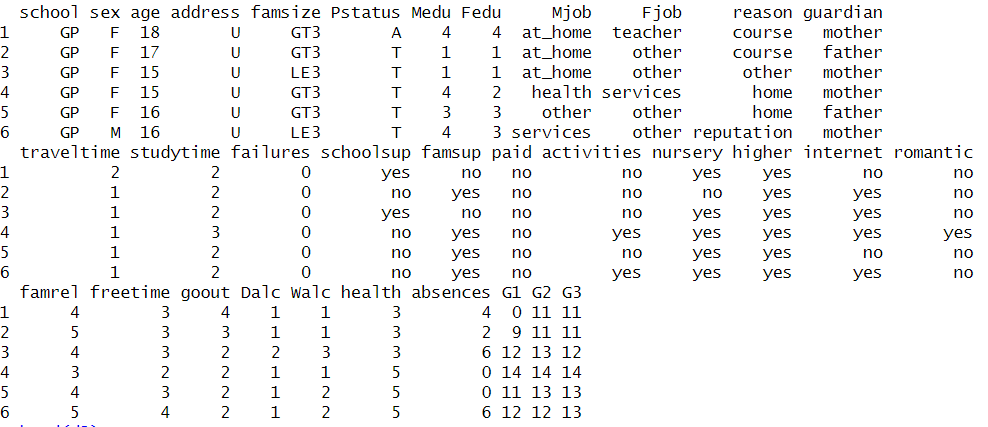
summary(d1)

summary(d2)

str(d1)

str(d2)





STEP 4: Handling missing data:

To check if there are any missing values in the dataset the following commands were used. This function returns the number of missing data values in the dataset.

missing\_value1 <- sum(is.na(d1))

missing\_value2 <- sum(is.na(d2))

As we goth 0 for both the commands, it was concluded that there were no missing values in the dataset we have used.

STEP 5: Summarizing the data:

Summarizing the data includes calculating the summary like mean, median, mode etc. or counting the frequency of categorical variables.

A screenshot of a computer screen

Description automatically generated

**DATA VISUALIZATION**

In data visualization, we represent the data in visual forms to make it easier to understand and analyze.

*BAR PLOTS~*

Bar plots are used to represent the frequency of categorical data while histograms are used to represent frequency of continuous data.

The following are some of the observations made:

A graph of a graph

Description automatically generated

The above graph one box shows the number of students who are in a romantic relationship and the other box shows those who are not.

A graph of a graph

Description automatically generated

This graph depicts the number of students who joined the school because of reasons like:

* Courses offered by the school.
* Distance of the school from home.
* Reputation of the school, etc.

We can see that majority of the students have opted the school because of the courses offered, i.e., many students have joined in on their own interest in the course.

A graph with a bar graph

Description automatically generated

The above plot is a histogram for the health of the students.

Here 1 represents students having some chronic medical condition while 5 represents students who are healthy.

We can observe that majority of the students are healthy which is a great indication of the localities healthcare system.

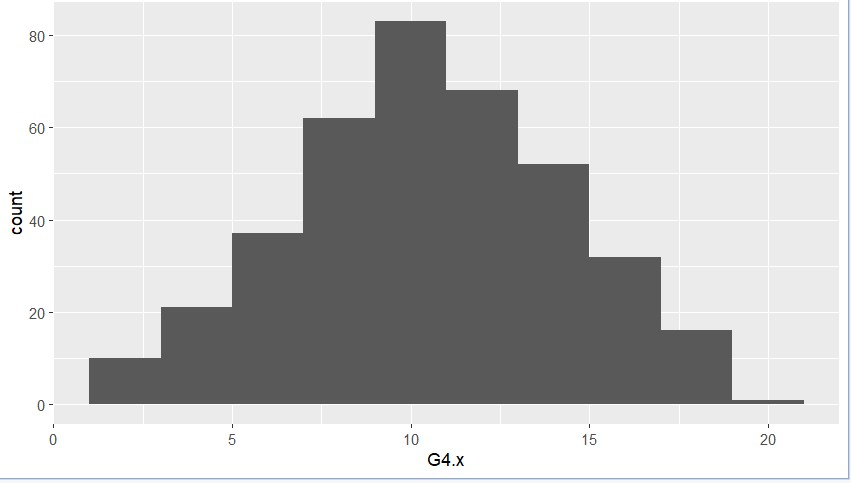
A graph of a graph

Description automatically generated

This graph represents the count of average absentees in a class. We notice that majority of the class is usually present for most of the classes as 0 i.e., 0 absent students is clearly the highest among others.

*HISTOGRAM~*

A histogram helps in representing the frequency of numerical data using rectangles.



A graph of a graph

Description automatically generated

The above plots show the distribution of values in G4.x and G4.y. We can see that in G4.x the graph is not skewed and the peak is above 80 whereas for G4.y the graph is skewed towards the right direction and the peak is around 100. This shows that students are scoring way better in Portuguese as compared to maths.

*DENSITY PLOT~*

Density plot is a smoothened histogram. It removes the kinks and smoothens the histogram. It basically show the distribution of the data.

A graph with a line

Description automatically generated A graph of a graph

Description automatically generated

*BOX PLOT~*

It displays 5 number summary of a set of data. It includes minimum 1st quartile, median, 3rd quartile, and maximum. In a box plot we draw the box from the 1st to the 3rd quartile. It basically gives the quick summary of the data and shows if there are any outliers.

A graph with a line

Description automatically generated A graph with a line

Description automatically generated

In these plots we can clearly observe there are no outliers.

*QQPLOT~*

QQplot is a type of probabiity plot that provides a graphical method to assess the probability of a distribution.

A graph showing a line

Description automatically generated A graph showing a line

Description automatically generatedWe can observe that the probability is following a linear pattern.

**FEATURE SELECTION AND ENGINEERING:**

Feature selections is the way of selecting the most relevant set of attributes from the given dataset. Not all the attributes affect the dependant variable the same way. We can do this by checking the correlation between each attribute and the dependant variable. In order to achieve this we used the rpart algorithm that recursively divides the dataset into subsets to find out which attributes are the most relevant. Here, from our analysis we conclude that weekend alcohol consumption is the most relevant feature followed by studytime and freetime.

STEP 1: Loading the packages:

The following packages wee used for feature selection-

library(caret)

library(dplyr)

library(randomForest)

library(ggcorrplot)

STEP 2: Performing initial feature exploration:

In this step we gain insight into the features and calculate summary statistics and check the correlation between variables. From this we can draw the correlation matrix which gives us a clear picture of which variables are relevant and which are not.

The correlation matrix is shown as follows-

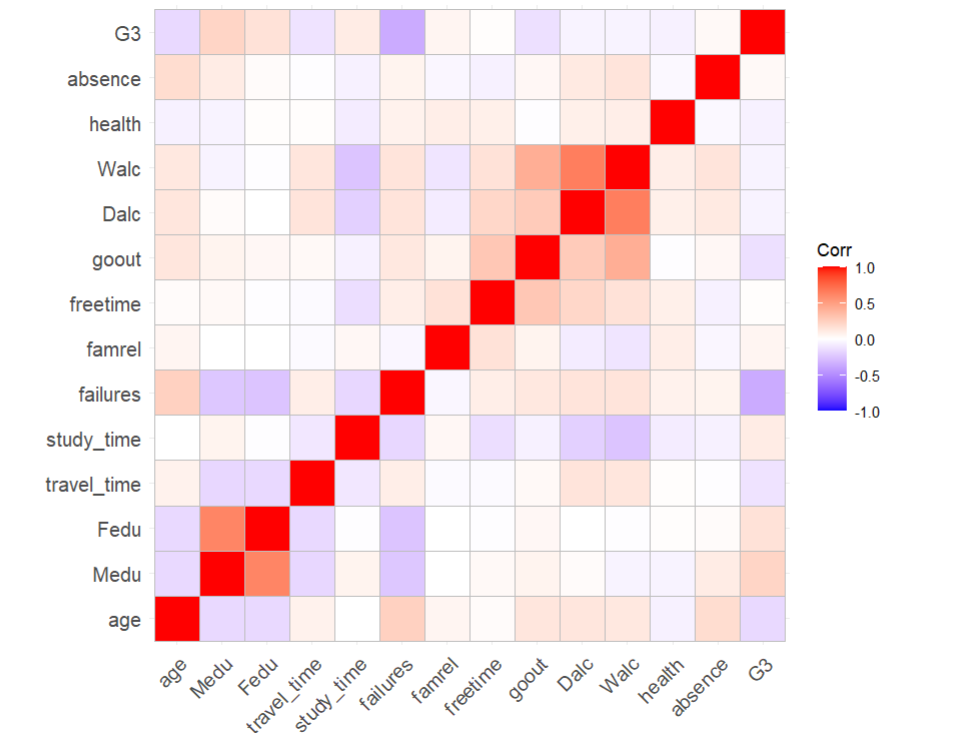
A computer screen shot of a code

Description automatically generated

A close-up of a screen

Description automatically generated

We can visualize this in the following graph:

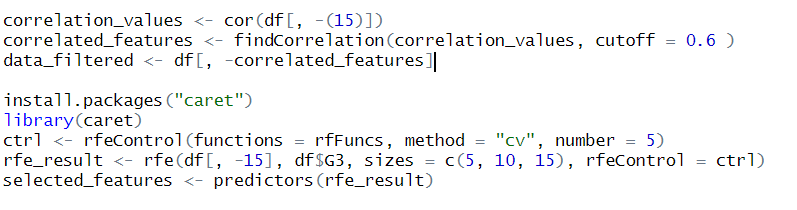


As we can see in the above plot, the attributes which have a total white hue in the cell corresponding to G3 is not relevant as they give a correlation of 0. The attributes which are reddish have positive correlation with G3 while bluish tint is the indicator of negative correlation with G3.

STEP 3: Filter based methods:

Filter-Based Methods: Filter-based methods evaluate the features independently of the machine learning algorithm. They consider statistical measures or domain knowledge to rank features and select the most relevant ones.

The following commands were used:



This removed all the irrelevant data with lesser degree of correlation with the dependent variable G3.

The relevant features selcted were:

* "G3"
* "absence"
* "failures"
* "Medu"
* "goout"
* "Walc"
* "Dalc"
* "Fedu"
* "age"
* "travel\_time"

STEP 4: Using rpart algorithm:

We got these relevant attributes-

A screenshot of a computer

Description automatically generated

**MODEL DEVELOPMENT**

*Linear Regression~*

Linear regression is the linear approach for modelling. It represents the relationship between dependant and independent variables on a straight line known as intercept. Our dependent variable is the marks in Portuguese and maths (G4.y and G4.x respectively) and our independent variables are Walc.x and Walc.y . We have also done the analysis for freetime and studytime.

A graph with red lines

Description automatically generated

summary(lm.r1)

Call:

lm(formula = G4.x ~ Walc.x, data = trainingset)

Residuals:

Min 1Q Median 3Q Max

-9.4919 -2.1585 -0.1585 2.5081 8.5081

Coefficients:

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

(Intercept) 11.0949 0.4650 23.859 <2e-16 \*\*\*

Walc.x -0.2697 0.1800 -1.499 0.135

---

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

Residual standard error: 3.746 on 263 degrees of freedom

Multiple R-squared: 0.008467, Adjusted R-squared: 0.004696

F-statistic: 2.246 on 1 and 263 DF, p-value: 0.1352

A graph with lines and numbers

Description automatically generated

summary(lm.r2)

Call:

lm(formula = G4.y ~ Walc.y, data = trainingset)

Residuals:

Min 1Q Median 3Q Max

-7.2619 -1.7425 -0.0758 1.6451 6.4978

Coefficients:

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

(Intercept) 13.2076 0.3167 41.701 < 2e-16 \*\*\*

Walc.y -0.4264 0.1227 -3.474 0.000599 \*\*\*

---

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

Residual standard error: 2.542 on 263 degrees of freedom

Multiple R-squared: 0.04388, Adjusted R-squared: 0.04025

F-statistic: 12.07 on 1 and 263 DF, p-value: 0.0005989

In the above graph, we have plotted our predicted values for both G4.x and G4.y respectively using the lm function and the attribute weekend alcohol consumption for both the subjects.

*Support vector machine~*

SVM is a linear as well as classification model. In this model, the data is divided into different groups based on similarities. The model considers the extreme points that helps in create the hyperplane which divides the plane into two categories of samples.

A line graph with numbers and dots

Description automatically generated

summary(svm.model)

Call:

svm(formula = Type ~ ., data = my.data, type = "C-classification", kernel = "linear",

scale = FALSE)

Parameters:

SVM-Type: C-classification

SVM-Kernel: linear

cost: 1

Number of Support Vectors: 18

( 9 9 )

Number of Classes: 2

Levels:

-1 1

> predict(svm.model)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 -1 1 1 -1 1 1 -1 -1 1 1 1 -1 -1 1 -1 1 1 1 1

Levels: -1 1

The above result is for Walc.x

A line of dots with numbers

Description automatically generated

summary(svm.model1)

Call:

svm(formula = Type ~ ., data = my.data, type = "C-classification", kernel = "linear",

scale = FALSE)

Parameters:

SVM-Type: C-classification

SVM-Kernel: linear

cost: 1

Number of Support Vectors: 18

( 9 9 )

Number of Classes: 2

Levels:

-1 1

> predict(svm.model1)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 -1 1 1 -1 1 -1 -1 -1 1 1 1 -1 1 1 1 1 1 1 1

Levels: -1 1

The above result if for Walc.y

**MODEL COMPARISON-**

Model comparison is the process of comparing two models based on how well it fits the data. We have used a regression model comparison model known as anova to compare the fit of two models. The anova function will take model object as arguments and return an anova testing whether the most complex algorithm is significantly better at capturing the data as compared to the simpler model. Comparison is done based on p-values.

> #linear walc.x vs svm walc.x

> anova(lm.r1,svm.model,test='F')

Analysis of Variance Table

Response: G4.x

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

Walc.x 1 31.5 31.505 2.2457 0.1352

Residuals 263 3689.6 14.029

> #linear walc.y vs svm walc.y

For the above G4.x the significant p-value is more than 0.05 which means the results are significant and the more complex mode i.e. SVM gives better results as compared linear regression.

> anova(lm.r2,svm.model1,test='F')

Analysis of Variance Table

Response: G4.y

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

Walc.y 1 78.0 77.999 12.07 0.0005989 \*\*\*

Residuals 263 1699.5 6.462

---

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

For the above G4.y the significant p-value is less than 0.05 which means the results are not significant and the less complex mode i.e. linear regression gives better results as compared to SVM.

**CONCLUSION:**

As we can see that we can only predict the performance of the students up to some extent as the correlation between the attribute and grade is less i.e., there is no proper correlation between them. We can infer from this that Student’s performance not only depends on some physical attributes but also depends on other factors like each student's individual capability and their mindset and other immeasurable factors. But some of the factors like daily alcohol consumption, study time or chronic ailments of the students play some major role in their results.

**SOURCE CODE:**

#data pre-processing

d1=read.table("student-mat.csv",sep=";",header=TRUE)

d2=read.table("student-por.csv",sep=";",header=TRUE)

d3=merge(d1,d2,by=c("school","sex","age","address","famsize","Pstatus","Medu","Fedu","Mjob","Fjob","reason","nursery","internet"))

print(nrow(d3)) # 382 students

#x=math records

#Y=port records

range(d3$Medu)

print(paste("No outliers in Medu(Mother's education). Values in the range",min(d3$Medu),"and",max(d3$Medu)))

range(d3$Fedu)

print(paste("No outliers in Fedu(Father's education). Values in the range",min(d3$Fedu),"and",max(d3$Fedu)))

range(d3$traveltime.x)

print(paste("No outliers in traveltime(home to school travel time) in x dataset. Values in the range",min(d3$traveltime.x),"and",max(d3$traveltime.x)))

range(d3$traveltime.y)

print(paste("No outliers in traveltime(home to school travel time) in y dataset. Values in the range",min(d3$traveltime.y),"and",max(d3$traveltime.y)))

range(d3$studytime.x)

print(paste("No outliers in studytime(weekly study time) in x dataset. Values in the range",min(d3$studytime.x),"and",max(d3$studytime.x)))

range(d3$studytime.y)

print(paste("No outliers in studytime(weekly study time) in y dataset. Values in the range",min(d3$studytime.y),"and",max(d3$studytime.y)))

range(d3$failures.x)

print(paste("No outliers in failures(number of past class failures) in x dataset. Values in the range",min(d3$failures.x),"and",max(d3$failures.x)))

range(d3$famrel.x)

print(paste("No outliers in famrel(quality of family relationships) in x dataset. Values in the range",min(d3$famrel.x),"and",max(d3$famrel.x)))

range(d3$freetime.x)

print(paste("No outliers in freetime(free time after school) in x dataset. Values in the range",min(d3$freetime.x),"and",max(d3$freetime.x)))

range(d3$goout.x)

print(paste("No outliers in goout(going out with friends) in x dataset. Values in the range",min(d3$goout.x),"and",max(d3$goout.x)))

range(d3$G1.x)

print(paste("No outliers in G1(first period grade) in x dataset. Values in the range",min(d3$G1.x),"and",max(d3$G1.x)))

range(d3$G1.y)

print(paste("No outliers in G1(first period grade) in y dataset. Values in the range",min(d3$G1.y),"and",max(d3$G1.y)))

range(d3$G2.x)

print(paste("No outliers in G2(second period grade) in y dataset. Values in the range",min(d3$G2.x),"and",max(d3$G2.x)))

c1 <- range(d3$Dalc.x)

print(paste("no outliers in workday alcohol consumption in the X dataset.values are within range",c1[1]," to ",c1[2]))

c2<- range(d3$Dalc.y)

print(paste("no outliers in workday alcohol consumption in the Y dataset.values are within range",c2[1]," to ",c2[2]))

c3<- range(d3$Walc.x)

print(paste("no outliers in weekend alcohol consumption in the X dataset.values are within range",c3[1]," to ",c3[2]))

c4<- range(d3$Walc.y)

print(paste("no outliers in weekend alcohol consumption in the Y dataset.values are within range",c4[1]," to ",c4[2]))

c5 <-range(d3$health.x)

print(paste("no outliers in health status in the X dataset.values are within range",c5[1]," to ",c5[2]))

c6<-range(d3$health.y)

print(paste("no outliers in health status in the Y dataset.values are within range",c6[1]," to ",c6[2]))

c7 <-range(d3$absences.x)

print(paste("no outliers in absence records in the X dataset.values are within range",c7[1]," to ",c7[2]))

c8 <-range(d3$absences.y)

print(paste("no outliers in absence records in the Y dataset.values are within range",c8[1]," to ",c8[2]))

c9<-range(d3$G3.x)

print(paste("no outliers in Final grades in the X dataset.values are within range",c9[1]," to ",c9[2]))

c10<-range(d3$G3.y)

print(paste("no outliers in Final grades in the Y dataset.values are within range",c10[1]," to ",c10[2]))

c11<-range(d3$failures.y)

print(paste("no outliers in Failures in the Y dataset.values are within range",c11[1]," to ",c11[2]))

c12<-range(d3$famrel.y)

print(paste("no outliers in quality of family relation in the Y dataset.values are within range",c12[1]," to ",c12[2]))

c13<-range(d3$freetime.y)

print(paste("no outliers in quality of freetime relation in the Y dataset.values are within range",c13[1]," to ",c13[2]))

c14<-range(d3$goout.y)

print(paste("no outliers in go out relation in the Y dataset.values are within range",c14[1]," to ",c14[2]))

c15<-range(d3$G2.y)

print(paste("no outliers in second period grade in the Y dataset.values are within range",c15[1]," to ",c15[2]))

#calculating average scores of students in math subject

for(i in 1:382){

d3$G4.y[i] <- (d3$G1.y[i]+d3$G2.y[i]+d3$G3.y[i])/3.0

d3$G4.x[i] <- (d3$G1.x[i]+d3$G2.x[i]+d3$G3.x[i])/3.0

}

#eda

summary(d3)

data(d3)

mean(d3$G4.y)

mean(d3$G4.x)

median(d3$G4.x)

median(d3$G4.y)

mode(d3$G4.y)

mode(d3$G4.x)

str(d3)

var(d3$G4.x, na.rm=TRUE)

sd(d3$G4.x, na.rm = TRUE)

cv <- sd(d3$G4.x) / mean(d3$G4.x) \* 100

cv

quantile(d3$G4.x)

range(d3$G4.x)

IQR(d3$G4.x)

sort(unique(d3$Walc.x))

var(d3$G4.y, na.rm=TRUE)

sd(d3$G4.y, na.rm = TRUE)

cv <- sd(d3$G4.y) / mean(d3$G4.y) \* 100

cv

quantile(d3$G4.y)

range(d3$G4.y)

IQR(d3$G4.y)

sort(unique(d3$studytime.y))

install.packages("ggplot2")

library(ggplot2)

# graphs

# bar plot

ggplot(d3, aes(x = G4.x)) +geom\_bar()

ggplot(d3, aes(x = G4.y)) +geom\_bar()

ggplot(d3, aes(x = G1.x)) +geom\_bar()

ggplot(d3, aes(x = G1.y)) +geom\_bar()

ggplot(d3, aes(x = G2.x)) +geom\_bar()

ggplot(d3, aes(x = G2.y)) +geom\_bar()

ggplot(d3, aes(x = G3.x)) +geom\_bar()

ggplot(d3, aes(x = G3.y)) +geom\_bar()

# histogram

ggplot(d3, aes(x = G4.x)) +geom\_histogram(bins = nclass.Sturges(d3$G4.x))

ggplot(d3, aes(x = G4.y)) +geom\_histogram(bins = nclass.Sturges(d3$G4.y))

# density curve

ggplot(d3, aes(x = G4.x)) + geom\_density()

ggplot(d3, aes(x = G4.y)) + geom\_density()

ggplot(d3, aes(x = G3.x)) + geom\_density()

ggplot(d3, aes(x = G3.y)) + geom\_density()

ggplot(d3, aes(x = G2.x)) + geom\_density()

ggplot(d3, aes(x = G2.y)) + geom\_density()

ggplot(d3, aes(x = G1.x)) + geom\_density()

ggplot(d3, aes(x = G1.y)) + geom\_density()

# box plot

ggplot(d3, (aes(x = Walc.x, y =G4.x))) +geom\_boxplot()

ggplot(d3, (aes(x = Walc.y, y =G4.y))) +geom\_boxplot()

ggplot(d3, (aes(x = freetime.x, y =G4.x))) +geom\_boxplot()

ggplot(d3, (aes(x = freetime.y, y =G4.y))) +geom\_boxplot()

ggplot(d3, (aes(x = studytime.x, y =G4.x))) +geom\_boxplot()

ggplot(d3, (aes(x = studytime.y, y =G4.y))) +geom\_boxplot()

# QQ Plot for Clay

ggplot(d3, aes(sample = G4.x)) +geom\_qq() +geom\_qq\_line()

ggplot(d3, aes(sample = G4.y)) +geom\_qq() +geom\_qq\_line()

#feature selection using r part algorithm

install.packages("Boruta")

install.packages("mlbench")

install.packages("caret")

install.packages("randomForest")

library(Boruta)

library(mlbench)

library(caret)

library(randomForest)

# Perform r part model search

install.packages("caTools")

set.seed(2)

library(caTools)

spli<- sample.split(d3,SplitRatio=0.7)

tr <-sample.split(spli,)

botrain <- subset(d3,spli="TRUE")

tdata <-train(sex ~ .,data=botrain,method="rpart")

rpartimp <- varImp(tdata)

rpartimp

#model development

split= sample.split(d3, SplitRatio = 0.7)

trainingset = subset(d3, split == "TRUE")

testset = subset(d3, split == "FALSE")

# Fitting Simple Linear Regression to the Training set for walc

lm.r1= lm(formula = G4.x~Walc.x,data = trainingset)

coef(lm.r1)

summary(lm.r1)

# Predicting the Test set results

ypred = predict(lm.r1, newdata = testset)

plot(ypred,col="red",type="l")

summary(ypred)

lm.r2= lm(formula = G4.y~Walc.y,data = trainingset)

coef(lm.r2)

summary(lm.r2)

# Predicting the Test set results

ypred = predict(lm.r2, newdata = testset)

plot(ypred,col="black",type="l")

summary(ypred)

#linear regression on study time

lm.r3= lm(formula = G4.x~studytime.x,data = trainingset)

coef(lm.r3)

summary(lm.r3)

# Predicting the Test set results

ypred = predict(lm.r3, newdata = testset)

plot(ypred,col="blue",type="l")

summary(ypred)

lm.r4= lm(formula = G4.y~studytime.y,data = trainingset)

coef(lm.r4)

summary(lm.r4)

# Predicting the Test set results

ypred = predict(lm.r4, newdata = testset)

plot(ypred,col="darkgreen",type="l")

summary(ypred)

#free time

lm.r5= lm(formula = G4.x~freetime.x,data = trainingset)

coef(lm.r5)

summary(lm.r5)

# Predicting the Test set results

ypred = predict(lm.r5,testset)

plot(ypred,col="darkred",type="l")

summary(ypred)

lm.r6= lm(formula = G4.y~freetime.y,data = trainingset)

coef(lm.r6)

summary(lm.r6)

# Predicting the Test set results

ypred = predict(lm.r6, newdata = testset)

plot(ypred,col="darkblue",type="l")

summary(ypred)

#SVM of Walc.x

install.packages("tidyverse")

install.packages("e1071")

#create a is sample indicator

sam <- c(rep(-1,10),rep(+1,10))

length(sam) <- length(d3$Walc.x)

library(e1071)

my.data <- data.frame(Walc.x= d3['Walc.x'],

G4.x= d3['G4.x'],

Type=as.factor(sam))

my.data

#plot the data

plot(my.data[,-3],col=(3)/2,pch=19); abline(h=0,v=0,lty=3)

#perform svm by calling the svm method and passing the parameters

svm.model <- svm(Type~.,

data=my.data,

type='C-classification',

kernel='linear',

scale=FALSE)

#display summary of svm

summary(svm.model)

#show the support vectors

points(my.data[svm.model$index, c(1,2)] , col="orange", cex=2)

#get parameters of the hyperplane

w <- t(svm.model$coefs)%\*% svm.model$SV

b <- svm.model$rho

#in this 2D case the hyperplane is the line w[1,1]\*x1 + w[1,2]\*2+b=0

abline(a=-b/w[1,2], b=-w[1,1]/w[1,2], col='blue', lty=3)

#predict the results

predict(svm.model)

#SVM of Walc.y

sam <- c(rep(-1,10),rep(+1,10))

length(sam) <- length(d3$Walc.y)

library(e1071)

my.data <- data.frame(

Walc.y= d3['Walc.y'],

G4.y= d3['G4.y'],

Type=as.factor(sam))

my.data

#plot the data

plot(my.data[,-3],col=(3)/2,pch=19); abline(h=0,v=0,lty=3)

#perform svm by calling the svm method and passing the parameters

svm.model1 <- svm(Type~.,

data=my.data,

type='C-classification',

kernel='linear',

scale=FALSE)

#display summary of svm

summary(svm.model1)

#show the support vectors

points(my.data[svm.model1$index, c(1,2)] , col="orange", cex=2)

#get parameters of the hyperplane

w <- t(svm.model1$coefs)%\*% svm.model1$SV

b <- svm.model1$rho

#in this 2D case the hyperplane is the line w[1,1]\*x1 + w[1,2]\*2+b=0

abline(a=-b/w[1,2], b=-w[1,1]/w[1,2], col='blue', lty=3)

#predict the results

predict(svm.model1)

#SVM of freetime.x

sam <- c(rep(-1,10),rep(+1,10))

length(sam) <- length(d3$freetime.x)

library(e1071)

my.data <- data.frame(

freetime.x= d3['freetime.x'],

G4.x= d3['G4.x'],

Type=as.factor(sam))

my.data

#plot the data

plot(my.data[,-3],col=(3)/2,pch=19); abline(h=0,v=0,lty=3)

#perform svm by calling the svm method and passing the parameters

svm.model2 <- svm(Type~.,

data=my.data,

type='C-classification',

kernel='linear',

scale=FALSE)

#display summary of svm

summary(svm.model2)

#show the support vectors

points(my.data[svm.model2$index, c(1,2)] , col="orange", cex=2)

#get parameters of the hyperplane

w <- t(svm.model2$coefs)%\*% svm.model2$SV

b <- svm.model2$rho

#in this 2D case the hyperplane is the line w[1,1]\*x1 + w[1,2]\*2+b=0

abline(a=-b/w[1,2], b=-w[1,1]/w[1,2], col='blue', lty=3)

#predict the results

predict(svm.model2)

#SVM od freertime.y

sam <- c(rep(-1,10),rep(+1,10))

length(sam) <- length(d3$freetime.y)

library(e1071)

my.data <- data.frame(

freetime.y= d3['freetime.y'],

G4.y= d3['G4.y'],

Type=as.factor(sam))

my.data

#plot the data

plot(my.data[,-3],col=(3)/2,pch=19); abline(h=0,v=0,lty=3)

#perform svm by calling the svm method and passing the parameters

svm.model2 <- svm(Type~.,

data=my.data,

type='C-classification',

kernel='linear',

scale=FALSE)

#display summary of svm

summary(svm.model2)

#show the support vectors

points(my.data[svm.model2$index, c(1,2)] , col="orange", cex=2)

#get parameters of the hyperplane

w <- t(svm.model2$coefs)%\*% svm.model2$SV

b <- svm.model2$rho

#in this 2D case the hyperplane is the line w[1,1]\*x1 + w[1,2]\*2+b=0

abline(a=-b/w[1,2], b=-w[1,1]/w[1,2], col='blue', lty=3)

#predict the results

predict(svm.model2)

#Polynomial regression of Walc.x

plot(d3$G4.x~d3$Walc.x)

lines(lowess(d3$G4.x~d3$Walc.x))

y <- lm(G4.x ~ Walc.x + I(Walc.x^2), data=d3)

summary(y)

lines(d3$Walc.x, predict(y))

lines(d3$Walc.x, predict(y),col=2)

#Polynomial regression of Walc.y

plot(d3$G4.y~d3$Walc.y)

lines(lowess(d3$G4.y~d3$Walc.y))

z <- lm(G4.y ~ Walc.y + I(Walc.y^2), data=d3)

summary(z)

lines(d3$Walc.y, predict(z))

lines(d3$Walc.y, predict(z),col=2)

#Polynomial regression of freetime.x

plot(d3$G4.x~d3$freetime.x)

lines(lowess(d3$G4.x~d3$freetime.x))

h <- lm(G4.x ~ freetime.x + I(freetime.x^2), data=d3)

summary(h)

lines(d3$freetime.x, predict(h))

lines(d3$freetime.x, predict(h),col=2)

#Polynomial regression of freetime.y

plot(d3$G4.y~d3$freetime.y)

lines(lowess(d3$G4.y~d3$freetime.y))

k <- lm(G4.y ~ freetime.y + I(freetime.y^2), data=d3)

summary(k)

lines(d3$G4.y, predict(k))

lines(d3$G4.y, predict(k),col=2)

#MODEL COMPARISON

#linear walc.x vs svm walc.x

anova(lm.r1,svm.model,test='F')

#linear walc.y vs svm walc.y

anova(lm.r2,svm.model1,test='F')

#linear freetime.x vs svm freetime.x

anova(lm.r5,svm.model2,test='F')

#linear freetime.y vs svm freetime.y

anova(lm.r6,svm.model3,test='F')

#linear freetime.x vs svm freetime.x

anova(lm.r5,svm.model2,test='F')