NAIVE BAYES to predict the GRADE (LOW (<6) / Medium (6-12) / High (>12))

Bogdan Tanasa

THE SECTIONS in the RMARKDOWN DOCUMENT :

1. INTRODUCTION

We are using the data from **UCI** : !(https://archive.ics.uci.edu/ml/datasets/Student+Performance)

We are reading a file about **STUDENTS**, and we aim to predict whether they have passed or not the exams **(PASS/no_PASS)**;

In contrast to the previous version where we have used the KNN-based approach, in the document below :

- **we are not showing the BAR PLOTS during DATA EXPLORATION step (we have done it already when we have presented the results after using KNN-approach)**
- **we are using the NAIVE BAYES algorithm instead of KNN**

The attributes in the **INPUT FILE** are the following :

- 1 school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)
- 2 sex - student's sex (binary: "F" - female or "M" - male)
- 3 age - student's age (numeric: from 15 to 22)
- 4 address - student's home address type (binary: "U" - urban or "R" - rural)
- 5 famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)
- 6 Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)
- 7 Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- 8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- 9 Mjob - mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- 10 Fjob - father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- 11 reason - reason to choose this school (nominal: close to "home", school "reputation", "course" preference or "other")
- 12 guardian - student's guardian (nominal: "mother", "father" or "other")
- 13 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)
- 14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
- 15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)
- 16 schoolsup - extra educational support (binary: yes or no)
- 17 famsup - family educational support (binary: yes or no)
- 18 paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- 19 activities - extra-curricular activities (binary: yes or no)
- 20 nursery - attended nursery school (binary: yes or no)
- 21 higher - wants to take higher education (binary: yes or no)
- 22 internet - Internet access at home (binary: yes or no)

- 23 romantic - with a romantic relationship (binary: yes or no)
- 24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
- 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)
- 26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)
- 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 29 health - current health status (numeric: from 1 - very bad to 5 - very good)
- 30 absences - number of school absences (numeric: from 0 to 93)

2. DATA EXPLORATION

```
library(ggplot2)
library(reshape2)
library(readxl)
library(dplyr)
library(tibble)
library(class)
library(gmodels)
library(caret)
library(e1071)
library(GGally)
library(klaR)

#########################################################
#########################################################
#########################################################
#########################################################

FILE1="student.mat.txt"
# FILE2="student.por.txt"
# FILE3="student.mat.and.por.txt"

#########################################################
#########################################################
#########################################################
#########################################################

student <- read.delim(FILE1, sep="\t", header=T, stringsAsFactors=F)

#########################################################
#########################################################

summary(student)
```

```
##     school              sex                 age         address
##  Length:395         Length:395         Min.   :15.0   Length:395
##  Class :character   Class :character   1st Qu.:16.0   Class :character
##  Mode  :character   Mode  :character   Median :17.0   Mode  :character
```

```
##                                     Mean   :16.7
##                                     3rd Qu.:18.0
##                                     Max.   :22.0
##    famsize            Pstatus             Medu            Fedu
##  Length:395         Length:395         Min.   :0.000   Min.   :0.000
##  Class :character   Class :character   1st Qu.:2.000   1st Qu.:2.000
##  Mode  :character   Mode  :character   Median :3.000   Median :2.000
##                                        Mean   :2.749   Mean   :2.522
##                                        3rd Qu.:4.000   3rd Qu.:3.000
##                                        Max.   :4.000   Max.   :4.000
##      Mjob               Fjob              reason            guardian
##  Length:395         Length:395         Length:395         Length:395
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    traveltime      studytime        failures       schoolsup
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   Length:395
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   Class :character
##  Median :1.000   Median :2.000   Median :0.0000   Mode  :character
##  Mean   :1.448   Mean   :2.035   Mean   :0.3342
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000
##     famsup              paid            activities          nursery
##  Length:395         Length:395         Length:395         Length:395
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    higher            internet           romantic            famrel
##  Length:395         Length:395         Length:395         Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:4.000
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
##                                                           Mean   :3.944
##                                                           3rd Qu.:5.000
##                                                           Max.   :5.000
##    freetime         goout            Dalc            Walc
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :3.000   Median :3.000   Median :1.000   Median :2.000
##  Mean   :3.235   Mean   :3.109   Mean   :1.481   Mean   :2.291
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##      health         absences            G1              G2
##  Min.   :1.000   Min.   : 0.000   Min.   : 3.00   Min.   : 0.00
##  1st Qu.:3.000   1st Qu.: 0.000   1st Qu.: 8.00   1st Qu.: 9.00
##  Median :4.000   Median : 4.000   Median :11.00   Median :11.00
##  Mean   :3.554   Mean   : 5.709   Mean   :10.91   Mean   :10.71
##  3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:13.00   3rd Qu.:13.00
##  Max.   :5.000   Max.   :75.000   Max.   :19.00   Max.   :19.00
##        G3
##  Min.   : 0.00
```

```
##  1st Qu.: 8.00
##  Median :11.00
##  Mean   :10.42
##  3rd Qu.:14.00
##  Max.   :20.00
```

```
str(student)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : chr  "at_home" "at_home" "at_home" "health" ...
##  $ Fjob      : chr  "teacher" "other" "other" "services" ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : chr  "yes" "no" "yes" "no" ...
##  $ famsup    : chr  "no" "yes" "no" "yes" ...
##  $ paid      : chr  "no" "no" "yes" "yes" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : chr  "yes" "no" "yes" "yes" ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```
class(student)
```

```
## [1] "data.frame"
```

Here we are starting to display the data for visual exploration.

```
##############################################################################################################
##############################################################################################################
# 1 school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)

# unique(student$school)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=school, fill=school))
```

```r
# ggsave("display.1.school.png")

student$school = as.factor(student$school)

##############################################################################################################
##############################################################################################################
# 2 sex - student's sex (binary: "F" - female or "M" - male)

# unique(student$sex)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=sex , fill=sex))

# ggsave("display.2.sex.png")

student$sex = as.factor(student$sex)

##############################################################################################################
##############################################################################################################
# 3 age - student's age (numeric: from 15 to 22)

# unique(student$age)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=age , fill=age))

# ggsave("display.3.age.png")

# AGE is already on the numerical scale !!
student$age = as.integer(student$age)

##############################################################################################################
##############################################################################################################
# 4 address - student's home address type (binary: "U" - urban or "R" - rural)

# unique(student$address) ## [1] "U" "R"

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=address, fill=address))

# ggsave("display.4.address.png")

student$address = as.factor(student$address)

##############################################################################################################
##############################################################################################################
# 5 famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)

# unique(student$famsize)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=famsize, fill=famsize))
```

```
# ggsave("display.5.famsize.png")

student$famsize = as.factor(student$famsize)

##############################################################################################################
##############################################################################################################
# 6 Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)

# unique(student$Pstatus)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=Pstatus, fill=Pstatus))

# ggsave("display.6.Pstatus.png")

student$Pstatus = as.factor(student$Pstatus)

##############################################################################################################
##############################################################################################################
# 7 Medu - mother's education (numeric: 0 - none,  1 - primary education (4th grade), 2 â€" 5th to 9th

# unique(student$Medu)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Medu, fill=Medu))

# ggsave("display.7.Medu.png")

# we may wanna use the numerical values in various regression models
student$Medu = as.integer(student$Medu)

##############################################################################################################
##############################################################################################################
# 8 Fedu - father's education (numeric: 0 - none,  1 - primary education (4th grade), 2 - 5th to 9th gr

unique(student$Fedu)
```

```
## [1] 4 1 2 3 0
```

```
# ggplot(data = student) +
#       geom_bar(mapping = aes(x=Fedu, fill=Fedu))

# ggsave("display.8.Fedu.png")

# we may wanna use the numerical values in various regression models
student$Fedu = as.integer(student$Fedu)

##############################################################################################################
##############################################################################################################
# 9 Mjob - mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrati

# unique(student$Mjob)

# ggplot(data = student) +
```

```
#       geom_bar(mapping = aes(x=Mjob, fill=Mjob))

# ggsave("display.9.Mjob.png")

student$Mjob = as.factor(student$Mjob)

##############################################################################################################
##############################################################################################################
# 10 Fjob - father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrat

# unique(student$Fjob)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=Fjob, fill=Fjob))

# ggsave("display.10.Fjob.png")

student$Fjob = as.factor(student$Fjob)

##############################################################################################################
##############################################################################################################
# 11 reason - reason to choose this school (nominal: close to "home", school "reputation", "course" pre

# unique(student$reason)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=reason, fill=reason))

# ggsave("display.11.reason.png")

student$reason = as.factor(student$reason)

##############################################################################################################
##############################################################################################################
# 12 guardian - student's guardian (nominal: "mother", "father" or "other")

# unique(student$guardian)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=guardian, fill=guardian))

# ggsave("display.12.guardian.png")

student$guardian = as.factor(student$guardian)

##############################################################################################################
##############################################################################################################
# 13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to

# unique(student$traveltime)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=traveltime, fill=traveltime))
```

```r
# ggsave("display.13.traveltime.png")

# we may wanna use the NUMERICAL VALUES :
student$traveltime = as.integer(student$traveltime)

#########################################################################################################
#########################################################################################################
# 14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 -

# unique(student$studytime)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=studytime, fill=studytime))

# ggsave("display.14.studytime.png")

# we may wanna use the NUMERICAL VALUES :
student$studytime = as.integer(student$studytime)

#########################################################################################################
#########################################################################################################
# 15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)

# unique(student$failures)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=failures, fill=failures))

# ggsave("display.15.failures.png")

# we may wanna use the NUMERICAL VALUES :
student$failures = as.integer(student$failures)

#########################################################################################################
#########################################################################################################
# 16 schoolsup - extra educational support (binary: yes or no)

# unique(student$schoolsup)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=schoolsup, fill=schoolsup))

# ggsave("display.16.schoolsup.png")

student$schoolsup = as.factor(student$schoolsup)

#########################################################################################################
#########################################################################################################
# 17 famsup - family educational support (binary: yes or no)

# unique(student$famsup)

# ggplot(data = student) +
```

```
#        geom_bar(mapping = aes(x=famsup, fill=famsup))

# ggsave("display.17.famsup.png")

student$famsup = as.factor(student$famsup)

##############################################################################################################
##############################################################################################################
# 18 paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)

# unique(student$paid)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=paid, fill=paid))

# ggsave("display.18.paid.png")

student$paid = as.factor(student$paid)

##############################################################################################################
##############################################################################################################
# 19 activities - extra-curricular activities (binary: yes or no)

# unique(student$activities)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=activities, fill=activities))

# ggsave("display.19.activities.png")

student$activities = as.factor(student$activities)

##############################################################################################################
##############################################################################################################
# 20 nursery - attended nursery school (binary: yes or no)

# unique(student$nursery)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=nursery, fill=nursery))

# ggsave("display.20.nursery.png")

student$nursery = as.factor(student$nursery)

##############################################################################################################
##############################################################################################################
# 21 higher - wants to take higher education (binary: yes or no)

# unique(student$higher)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=higher, fill=higher))
```

```r
# ggsave("display.21.higher.png")

student$higher = as.factor(student$higher)

##############################################################################################################
##############################################################################################################
# 22 internet - Internet access at home (binary: yes or no)

# unique(student$internet)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=internet, fill=internet))

# ggsave("display.22.internet.png")

student$internet = as.factor(student$internet)

##############################################################################################################
##############################################################################################################
# 23 romantic - with a romantic relationship (binary: yes or no)

# unique(student$romantic)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=romantic, fill=romantic))

# ggsave("display.23.romantic.png")

student$romantic = as.factor(student$romantic)

##############################################################################################################
##############################################################################################################
# 24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)

# unique(student$famrel)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=famrel, fill=famrel))

# ggsave("display.24.famrel.png")

# i believe that we can keep these as numerical :
student$famrel = as.integer(student$famrel)

##############################################################################################################
##############################################################################################################
# 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)

# unique(student$freetime)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=freetime, fill=freetime))
```

```r
# ggsave("display.25.freetime.png")

# i believe that we can keep these as numerical :
student$freetime = as.integer(student$freetime)

##########################################################################################################
##########################################################################################################
# 26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)

# unique(student$goout)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=goout, fill=goout))

# ggsave("display.26.goout.png")

# i believe that we can keep these as numerical :
student$goout = as.integer(student$goout)

##########################################################################################################
##########################################################################################################
# 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)

# unique(student$Dalc)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=Dalc, fill=Dalc))

# ggsave("display.27.Dalc.png")

# i believe that we can keep these as numerical :
student$Dalc = as.integer(student$Dalc)

##########################################################################################################
##########################################################################################################
# 28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)

# unique(student$Walc)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=Walc, fill=Walc))

# ggsave("display.28.Walc.png")

# i believe that we can keep these as numerical :
student$Walc = as.integer(student$Walc)

##########################################################################################################
##########################################################################################################
# 29 health - current health status (numeric: from 1 - very bad to 5 - very good)

# unique(student$health)
```

```r
# ggplot(data = student) +
#        geom_bar(mapping = aes(x=health, fill=health))

# ggsave("display.29.health.png")

# i believe that we can keep these as numerical :
student$health = as.integer(student$health)

##########################################################################################################
##########################################################################################################
# 30 absences - number of school absences (numeric: from 0 to 93)

# unique(student$absences)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=absences, fill=absences))

# ggsave("display.30.absences.png")

# i believe that we can keep these as numerical :
student$absences = as.integer(student$absences)

##########################################################################################################
##########################################################################################################
# $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...

# unique(student$G1)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=G1, fill=G1))

# ggsave("display.0.G1.png")

# i believe that we can keep these as numerical, although we may not need it :
student$G1 = as.factor(student$G1)

##########################################################################################################
##########################################################################################################
# $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...

# unique(student$G2)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=G2, fill=G2))

# ggsave("display.0.G2.png")

# i believe that we can keep these as numerical, although we may not need it :
student$G2 = as.factor(student$G2)

##########################################################################################################
##########################################################################################################
# $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```r
# unique(student$G3)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=G3, fill=G3))

# ggsave("display.0.G3.png")

# i believe that we can covert it into RANGES of VALUES :
student$G3 = as.factor(student$G3)

##############################################################################################################
##############################################################################################################
##############################################################################################################
##############################################################################################################
##############################################################################################################
##############################################################################################################

summary(student)
```

```
##  school   sex          age       address famsize   Pstatus      Medu
##  GP:349   F:208   Min.   :15.0   R: 88   GT3:281   A: 41   Min.   :0.000
##  MS: 46   M:187   1st Qu.:16.0   U:307   LE3:114   T:354   1st Qu.:2.000
##                   Median :17.0                            Median :3.000
##                   Mean   :16.7                            Mean   :2.749
##                   3rd Qu.:18.0                            3rd Qu.:4.000
##                   Max.   :22.0                            Max.   :4.000
##
##       Fedu             Mjob           Fjob            reason      guardian
##  Min.   :0.000   at_home : 59   at_home : 20   course    :145   father: 90
##  1st Qu.:2.000   health  : 34   health  : 18   home      :109   mother:273
##  Median :2.000   other   :141   other   :217   other     : 36   other : 32
##  Mean   :2.522   services:103   services:111   reputation:105
##  3rd Qu.:3.000   teacher : 58   teacher : 29
##  Max.   :4.000
##
##    traveltime      studytime        failures      schoolsup famsup     paid
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   no :344   no :153   no :214
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   yes: 51   yes:242   yes:181
##  Median :1.000   Median :2.000   Median :0.0000
##  Mean   :1.448   Mean   :2.035   Mean   :0.3342
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000
##
##  activities nursery   higher    internet  romantic      famrel
##  no :194    no : 81   no : 20   no : 66   no :263   Min.   :1.000
##  yes:201    yes:314   yes:375   yes:329   yes:132   1st Qu.:4.000
##                                                     Median :4.000
##                                                     Mean   :3.944
##                                                     3rd Qu.:5.000
##                                                     Max.   :5.000
##
##     freetime         goout            Dalc            Walc
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
```

```
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :3.000   Median :3.000   Median :1.000   Median :2.000
##  Mean   :3.235   Mean   :3.109   Mean   :1.481   Mean   :2.291
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##      health         absences            G1            G2            G3
##  Min.   :1.000   Min.   : 0.000   10     : 51   9      : 50   10     : 56
##  1st Qu.:3.000   1st Qu.: 0.000   8      : 41   10     : 46   11     : 47
##  Median :4.000   Median : 4.000   11     : 39   12     : 41   0      : 38
##  Mean   :3.554   Mean   : 5.709   7      : 37   13     : 37   15     : 33
##  3rd Qu.:5.000   3rd Qu.: 8.000   12     : 35   11     : 35   8      : 32
##  Max.   :5.000   Max.   :75.000   13     : 33   15     : 34   12     : 31
##                                   (Other):159   (Other):152   (Other):158
```

```
str(student)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 1 1 1 2 2 1 2 2 ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 2 2 2 2 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 2 1 1 2 2 1 2 1 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : Factor w/ 5 levels "at_home","health",..: 1 1 1 2 3 4 3 3 4 3 ...
##  $ Fjob      : Factor w/ 5 levels "at_home","health",..: 5 3 3 4 3 3 3 5 3 3 ...
##  $ reason    : Factor w/ 4 levels "course","home",..: 1 1 3 2 2 4 2 2 2 2 ...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 2 1 2 2 1 2 2 2 2 2 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : Factor w/ 2 levels "no","yes": 2 1 2 1 1 1 1 2 1 1 ...
##  $ famsup    : Factor w/ 2 levels "no","yes": 1 2 1 2 2 2 1 2 2 2 ...
##  $ paid      : Factor w/ 2 levels "no","yes": 1 1 2 2 2 2 1 1 2 2 ...
##  $ activities: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 1 2 ...
##  $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
##  $ higher    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ internet  : Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 2 ...
##  $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : Factor w/ 17 levels "3","4","5","6",..: 3 3 5 13 4 13 10 4 14 12 ...
##  $ G2        : Factor w/ 17 levels "0","4","5","6",..: 4 3 6 12 8 13 10 3 16 13 ...
##  $ G3        : Factor w/ 18 levels "0","4","5","6",..: 4 4 8 13 8 13 9 4 17 13 ...
```

```
class(student)
```

```
## [1] "data.frame"
```

3. DATA FILTERING

```
## the OUTPUT VARIABLES is G3
## when SELECTING the FEATURES : we may remove G1 and G2

student1 <- subset(student, select = -c(G1, G2))

student2 <- subset(student1,
                   select = -c(school, sex, address, famsize, Pstatus,
                   Mjob, Fjob, reason, guardian, schoolsup, famsup, paid, activities, nursery,
                   higher, internet, romantic))

str(student2)
```

```
## 'data.frame':    395 obs. of  14 variables:
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G3        : Factor w/ 18 levels "0","4","5","6",..: 4 4 8 13 8 13 9 4 17 13 ...
```

```
student2$G3 = as.factor(student2$G3)

table(student2$G3)
```

```
##
##  0  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 38  1  7 15  9 32 28 56 47 31 31 27 33 16  6 12  5  1
```

```
### for simplicity, to work with a copy of STUDENT3
### although we may keep as well ALL the FEATURES

student3 = subset(student2,
                  select= c(age, traveltime, studytime, failures, absences, G3))

table(student3$G3)
```

```
##
##  0  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 38  1  7 15  9 32 28 56 47 31 31 27 33 16  6 12  5  1
```

```
### shall we keep as well ALL the FEATURES

### student3 = student1
```

4. DATA TRANSFORMATION

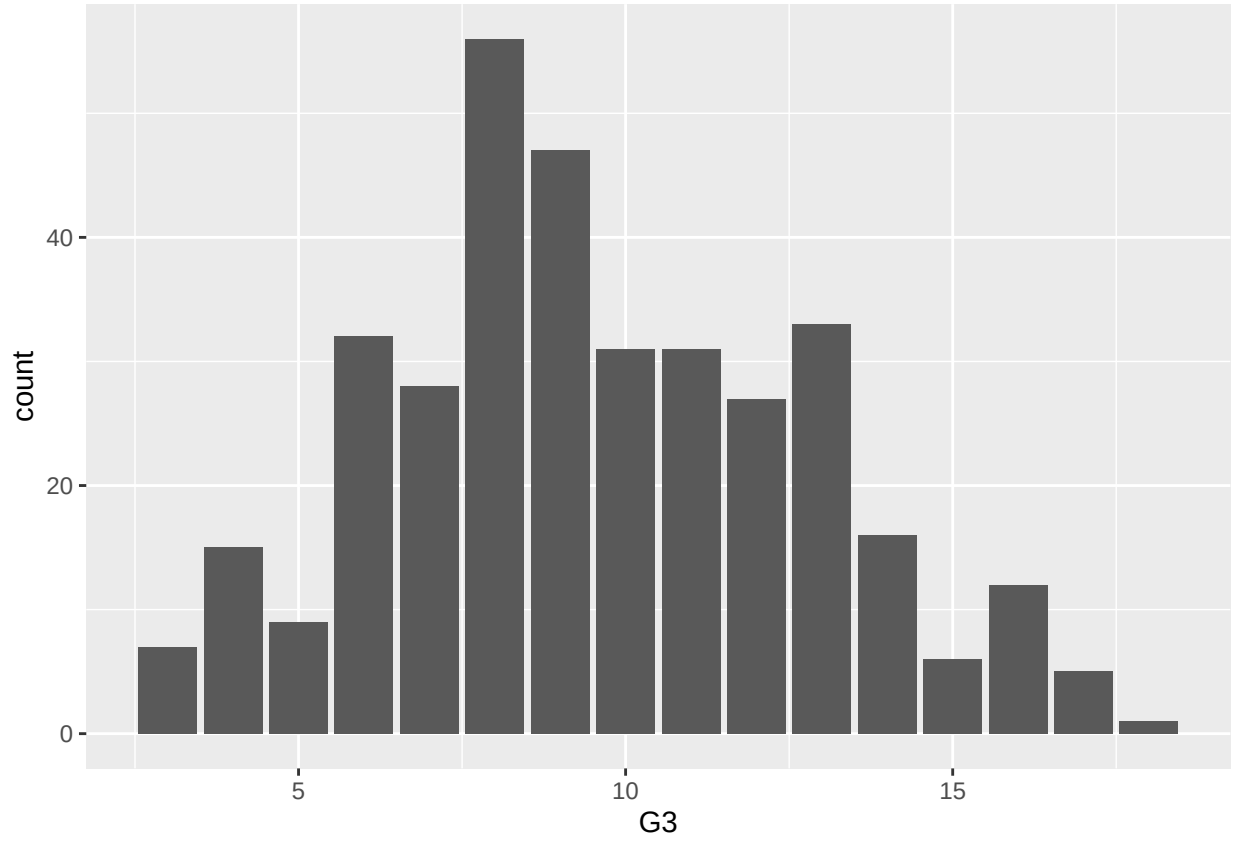
```
## in order to REMOVE the RECORDS where the GRADE 3 is > 2 :
## a new piece of R code (in contrast to the previous version)

dim(student3)
```

```
## [1] 395   6
```

```
student3$G3 = as.integer(student3$G3)

student4 = student3[student3$G3 > 2, ]

dim(student4)
```

```
## [1] 356   6
```

```
ggplot(data = student4) +
       geom_bar(mapping = aes(x=G3, fill=G3))
```




```
ggsave("display.0.G3.after.filtering.grade3.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
student3 = student4
```

```
## TRANSFORMING G3 into RANGES of LOW, MEDIUM, HIGH :
## LOW : 2 - 6
## MEDIUM : 6 - 12
## HIGH : > 12

student3$G3 = as.integer(student3$G3)

student3$RESULT[student3$G3 <= 6] = "Low"
student3$RESULT[student3$G3 > 6 & student3$G3 < 12 ] = "Medium"
student3$RESULT[student3$G3 >=12 ] = "High"

student3 <- subset(student3, select = -c(G3))

student3$RESULT = as.factor(student3$RESULT)
```

5. TRAINING AND TEST SETS

```
## CHOOSING the TRAINING and TESTING SETS

indxTrain <- createDataPartition(student3$RESULT,
                                 p = .75,
                                 list = FALSE)

training <- student3[indxTrain,]
head(training)
```

```
##   age traveltime studytime failures absences RESULT
## 1  18          2         2        0        6    Low
## 2  17          1         2        0        4    Low
## 3  15          1         2        3       10 Medium
## 4  15          1         3        0        2   High
## 6  16          1         2        0       10   High
## 7  16          1         2        0        0 Medium
```

```
testing <- student3[-indxTrain,]
head(testing)
```

```
##    age traveltime studytime failures absences RESULT
## 5   16          1         2        0        4 Medium
## 8   17          2         2        0        6    Low
## 13  15          1         1        0        2   High
## 20  16          1         1        0        4 Medium
## 29  16          1         2        0        4 Medium
## 31  15          1         2        0        0 Medium
```

```
dim(student3)
```

```
## [1] 356   6
```

```
dim(training)
```

```
## [1] 268   6
```

```r
dim(testing)
```

```
## [1] 88  6
```

6. PRE-PROCESSING THE DATA

```r
### PRE-PROCESSING the DATA

trainX        <- training[, names(training) != "RESULT"]

# for NB we may not need to CENTER and SCALE the data :)
# preProcValues <- preProcess(x = trainX, method = c("center", "scale"))
# preProcValues

names(trainX)
```

```
## [1] "age"        "traveltime" "studytime"  "failures"   "absences"
```

```r
dim(trainX)
```

```
## [1] 268   5
```

```r
names(training)
```

```
## [1] "age"        "traveltime" "studytime"  "failures"   "absences"
## [6] "RESULT"
```

```r
### THE BALANCE of the DATA in TRAINING and TESTING SETS

prop.table(table(training$RESULT)) * 100
```

```
##
##     High      Low   Medium
## 27.98507 17.91045 54.10448
```

```r
prop.table(table(testing$RESULT)) * 100
```

```
##
##     High      Low   Medium
## 28.40909 17.04545 54.54545
```

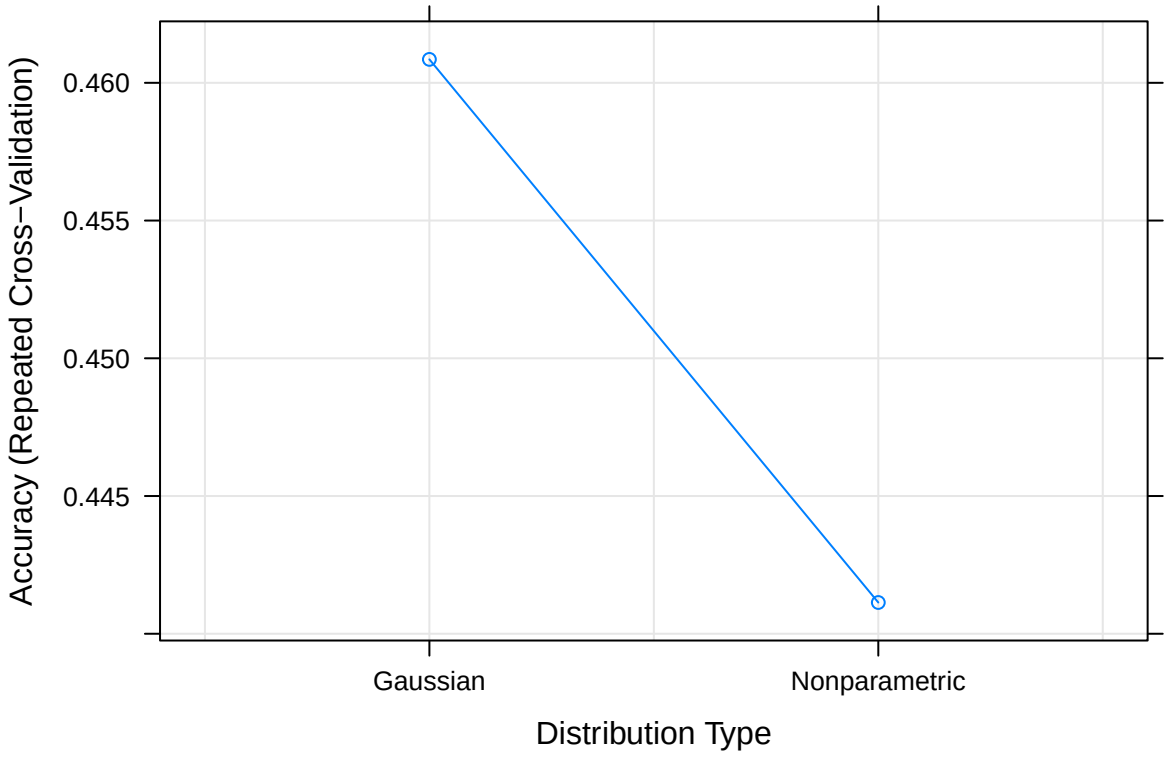
7. PERFORMING THE TRAINING

```r
### PERFORMING the TRAINING

set.seed(400)
ctrl <- trainControl(method="repeatedcv", repeats = 10)

nbFit = train( RESULT~ .,
                 data = training,
                 method = "nb",
                 trControl = ctrl)
```

```
## The output of nbFit fit

nbFit
```

```
## Naive Bayes
##
## 268 samples
##   5 predictor
##   3 classes: 'High', 'Low', 'Medium'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 242, 242, 240, 241, 241, 241, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.4608511  0.1693887
##    TRUE      0.4411350  0.1255610
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE and adjust
##  = 1.
```

```
plot(nbFit)
```




```
png("the.results.nb.FIT.png")
plot(nbFit)
dev.off()
```

```
## pdf
##   2
```

8. MAKING THE PREDICTIONS

```
### Making the PREDICTIONS :

nbPredict <- predict(nbFit, newdata = testing)
```

9. THE CONFUSION MATRIX (caret package)

```
### COMPUTING the CONFUSION MATRIX :

confusionMatrix(nbPredict, testing$RESULT)
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction High Low Medium
##     High     20   6     34
##     Low       0   3      2
##     Medium    5   6     12
##
## Overall Statistics
##
##                Accuracy : 0.3977
##                  95% CI : (0.2949, 0.5077)
##     No Information Rate : 0.5455
##     P-Value [Acc > NIR] : 0.9981
##
##                   Kappa : 0.0792
##
##  Mcnemar's Test P-Value : 1.704e-06
##
## Statistics by Class:
##
##                      Class: High Class: Low Class: Medium
## Sensitivity               0.8000    0.20000        0.2500
## Specificity               0.3651    0.97260        0.7250
## Pos Pred Value            0.3333    0.60000        0.5217
## Neg Pred Value            0.8214    0.85542        0.4462
## Prevalence                0.2841    0.17045        0.5455
## Detection Rate            0.2273    0.03409        0.1364
## Detection Prevalence      0.6818    0.05682        0.2614
## Balanced Accuracy         0.5825    0.58630        0.4875
```

```
mean(nbPredict == testing$RESULT)
```

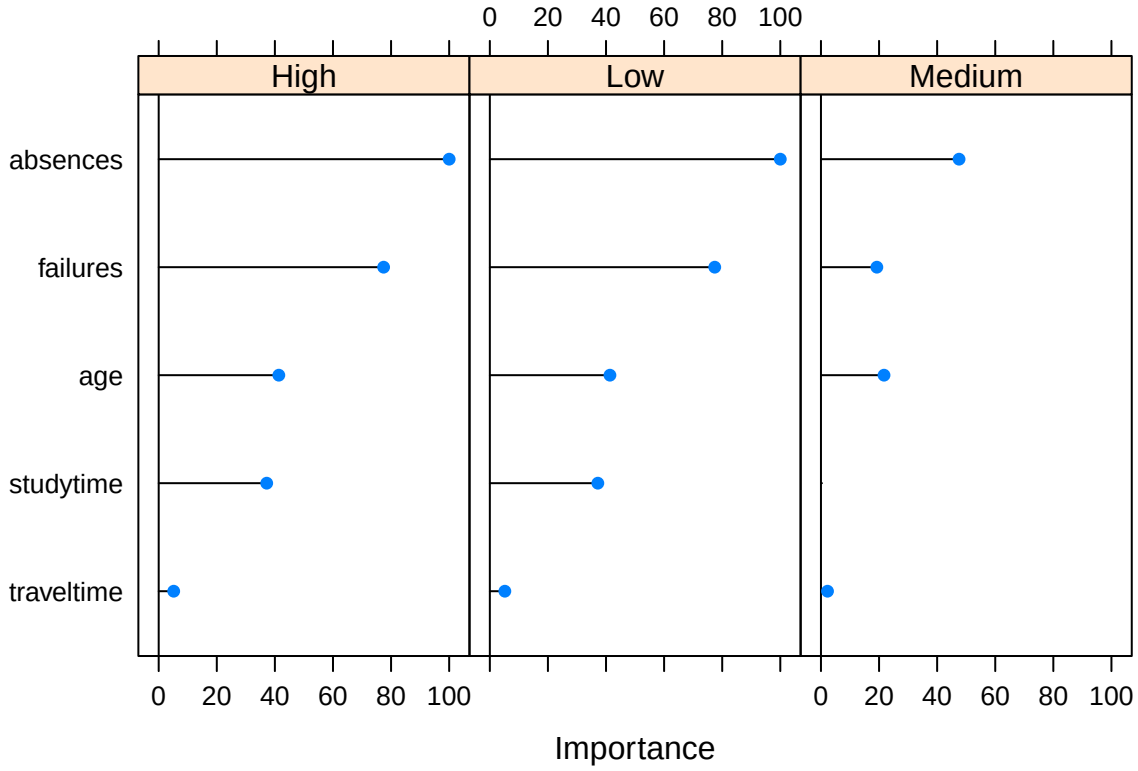
```
## [1] 0.3977273
```

```
dim(student3)
```

```
## [1] 356   6
```

```
# We implement the NB model also in other packages ("klaR", "e1071").
# here only another version of the R code

# library(e1071)

# x = training[, -6]
# y = training$RESULT

# model = train(x, y, 'nb', trControl=trainControl(method='cv',number=10))

# predict(model$finalModel,x)
# head(predict(model$finalModel,x)$class)

# table(predict(model$finalModel,x)$class,y)

# Predict <- predict(model, newdata = testing )
```

```
# We draw a plot that shows how each predictor variable is independently
# responsible for predicting the outcome.

# to display Variable Performance
# X <- varImp(model)
# plot(X)

# the confusion matrix to see accuracy value and other parameter values
# confusionMatrix(Predict, testing$RESULT)

X <- varImp(nbFit)
plot(X)
```

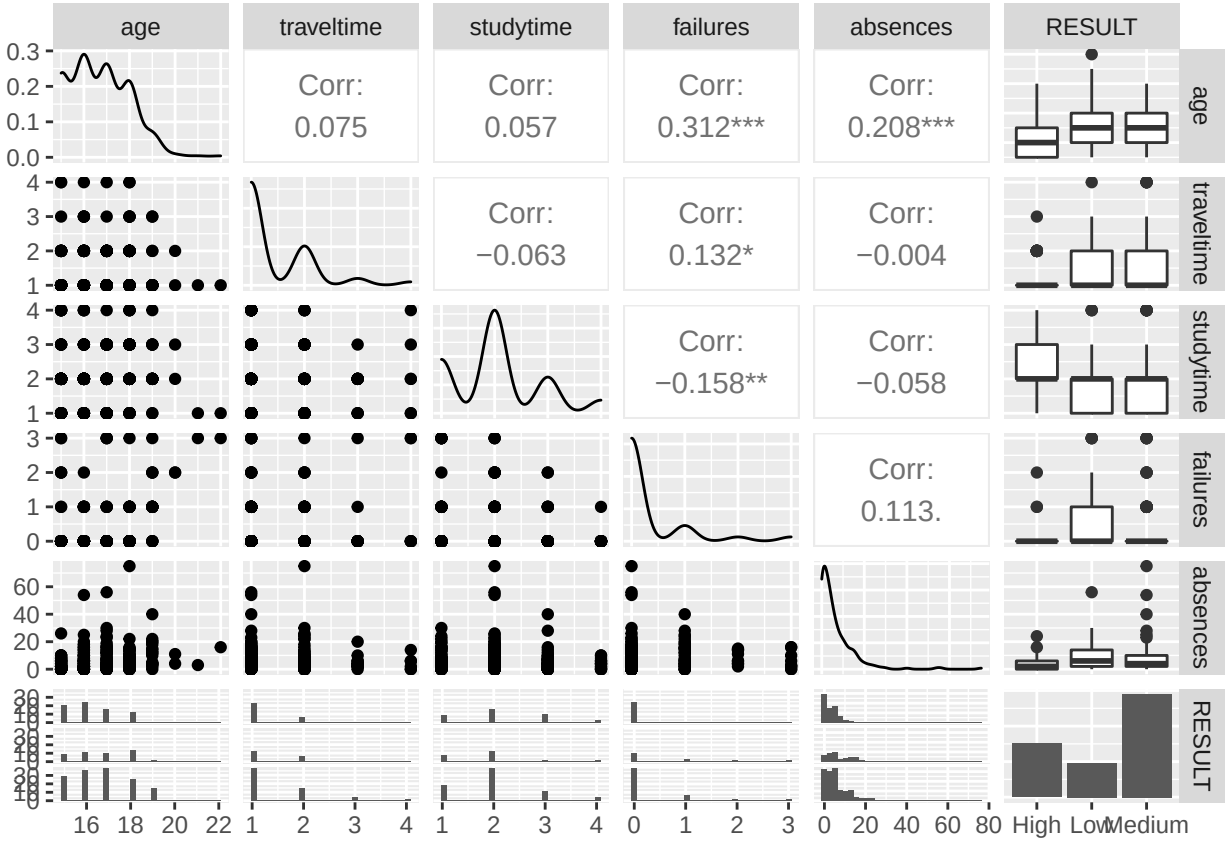


10. THE RESULTS (klaR package)

```
### looking at the CORRELATIONS between the FEATURES
library(GGally)

ggpairs(training)
```

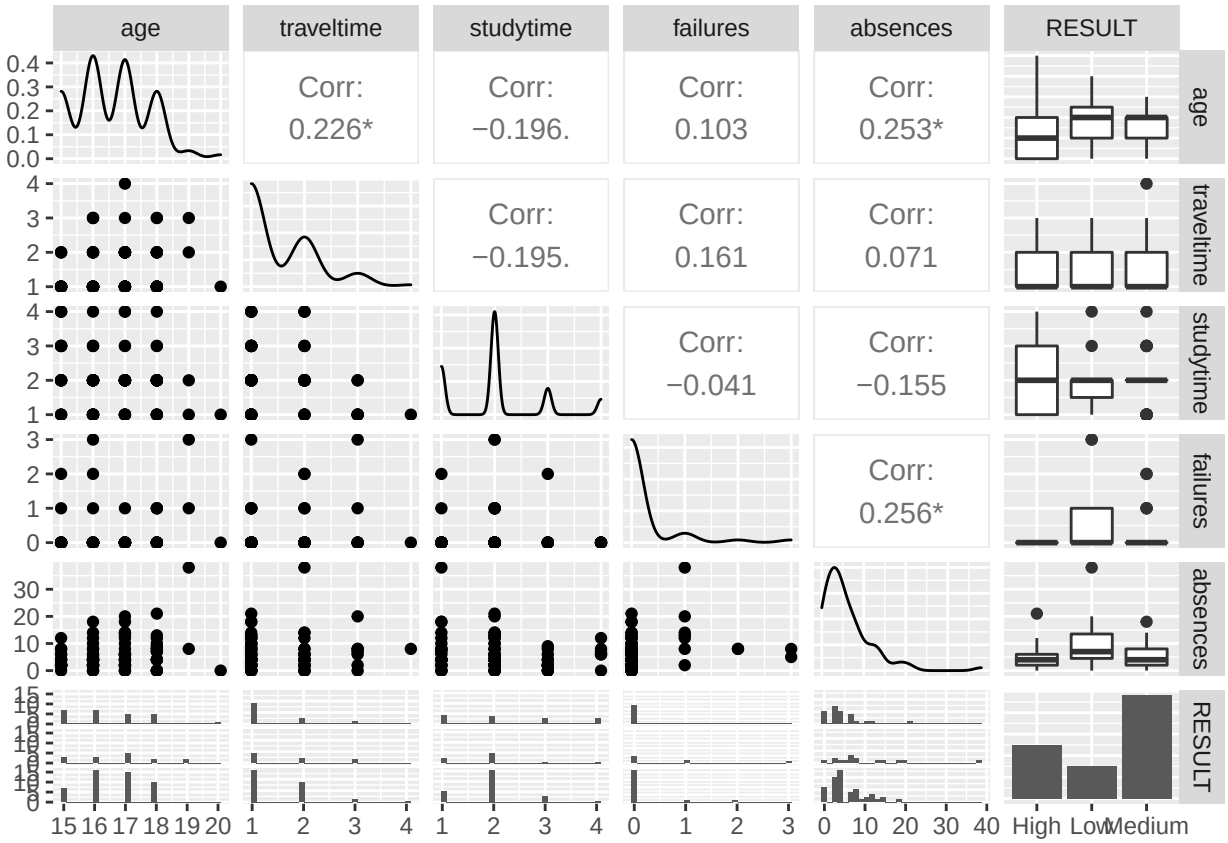
```
##  plot: [1,1] [>-----------------------------------------------]  3% est: 0s  plot: [1,2] [==>-----
```

```
##  plot: [6,2] [=========================================>------] 89% est: 0s `stat_bin()` using `bi
##  plot: [6,3] [==========================================>----] 92% est: 0s `stat_bin()` using `bi
##  plot: [6,4] [===========================================>---] 94% est: 0s `stat_bin()` using `bi
##  plot: [6,5] [============================================>-] 97% est: 0s `stat_bin()` using `bi
##  plot: [6,6] [=============================================]100% est: 0s
```




```
ggpairs(testing)
```

```
##  plot: [1,1] [>-------------------------------------------------]   3% est: 0s  plot: [1,2] [==>-----
##  plot: [6,2] [=========================================>------] 89% est: 0s `stat_bin()` using `bi
##  plot: [6,3] [==========================================>----] 92% est: 0s `stat_bin()` using `bi
##  plot: [6,4] [===========================================>---] 94% est: 0s `stat_bin()` using `bi
##  plot: [6,5] [============================================>-] 97% est: 0s `stat_bin()` using `bi
##  plot: [6,6] [=============================================]100% est: 0s
```




```
### using KLAR PACKAGE
library(klaR)

model = NaiveBayes( RESULT~ ., data = training)

predictions <- model %>% predict(testing)

# The ACCURACY
mean(predictions$class == testing$RESULT)
```

```
## [1] 0.3977273
```

As we can see, shall we set up the ML approach with NB, the accuracies of our models are almost equal and not too great.