“Student Performance - Executive Report”

Tools for Decision Making- Project 02

Babak Barghi, Cyra Stamm, Daniel Zöttl

December 20, 2020

Contents

The analysis is carried out in *R 4.0.2*[1] and the package *tidyverse* [2], *kableExtra* [3], *RColorBrewer* [4], *gridExtra* [5], *caret* [6], *rpart* [7], *rpart.plot* [8] and *caTools* [9] are used.

```
library("tidyverse")
library("kableExtra")
library("RColorBrewer")
library("gridExtra")
library("caret")
library("rpart")
library ("rpart.plot")
library("caTools")
```

1 Introduction

The objective of this analysis is to predict the students academic results based on the information given on their circumstances and based on that suggest a set of strategies or actions for the schools. The given data approach student achievement in secondary education of two schools in Portugal. The data attributes include student grades, demographic, social and school related features. Two data sets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por).

```
#importing the data sets
mat <- read.csv("student-mat.csv", header=TRUE, sep=";", dec=".")
por <- read.csv("student-por.csv", header=TRUE, sep=";", dec=".")
```

1.1 Data Overview

The data sets *por* and *mat* provide 649 and 395 observations respectively representing the different students of the schools with 33 different variables. The variables are going to be into consideration for this analysis to understand the influence of these predictors on the final grades of the students.

```
#number of observations
nrow(por)
```

```
## [1] 649
```

```
nrow(mat)
```

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

```
#number of variables
ncol(por)
```

```
## [1] 33
```

```
ncol(mat)
```

```
## [1] 33
```

To check whether the data sets are in need of cleaning, the NAs are checked.

```
#Check the NAs
sum(is.na(mat))
```

```
## [1] 0
```

```
sum(is.na(por))
```

```
## [1] 0
```

As we see there is no missing values in any data sets, thus before starting with the analysis we would take a closer look at data frames using *str* function.

```
#Close look
str(mat)
```

```
## '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 ...
```

By using the *str* function, we are able to view the different variables that influence the students grades. As we already taken from the exercise, the variables from 1-30 represent the different influencing factors on the students grades describing their circumstances. The variables 31, 32 and 33 represent the first period grade, second period grade and final grade respectively. As there are no NAs in the data set and the variables match the description in the task, we conclude that the data frames are already in a tidy format and does not require cleaning for the data analysis.

2 Methodology

In the following we are explaining the methodology used for the data analysis. To explore the data sets and get a first glance analysis, a box plot is used to illustrate the scores of G1, G2 and G3 which represent the grade of each exam during the semester for the students.

```
#mat
grades1 <- gather(mat,key = grade,value = score, G1,G2,G3)

p1 <- ggplot(grades1, aes(x=grade, y=score, fill=grade)) +
      geom_boxplot() +
      coord_flip() +
      stat_summary(fun = "mean", color="red", shape=15) +
      geom_jitter(alpha=0.3, width = 0.2) +
 labs(x = "Cluster Groups of Mathematics") +
  scale_fill_brewer(palette = "Dark2") +
 geom_hline(yintercept=10, linetype="dashed", color = "blue") +
  theme_bw()

#por
grades2 <- gather(por,key = grade,value = score, G1,G2,G3)

p2 <- ggplot(grades2, aes(x=grade, y=score, fill=grade)) +
      geom_boxplot() +
      coord_flip() +
      stat_summary(fun = "mean", color="red", shape=15) +
      geom_jitter(alpha=0.3, width = 0.2) +
 labs(x = "Cluster Groups of Portuguese") +
  scale_fill_brewer(palette = "Dark2") +
 geom_hline(yintercept=10, linetype="dashed", color = "blue") +
  theme_bw()


grid.arrange(p1,p2)
```

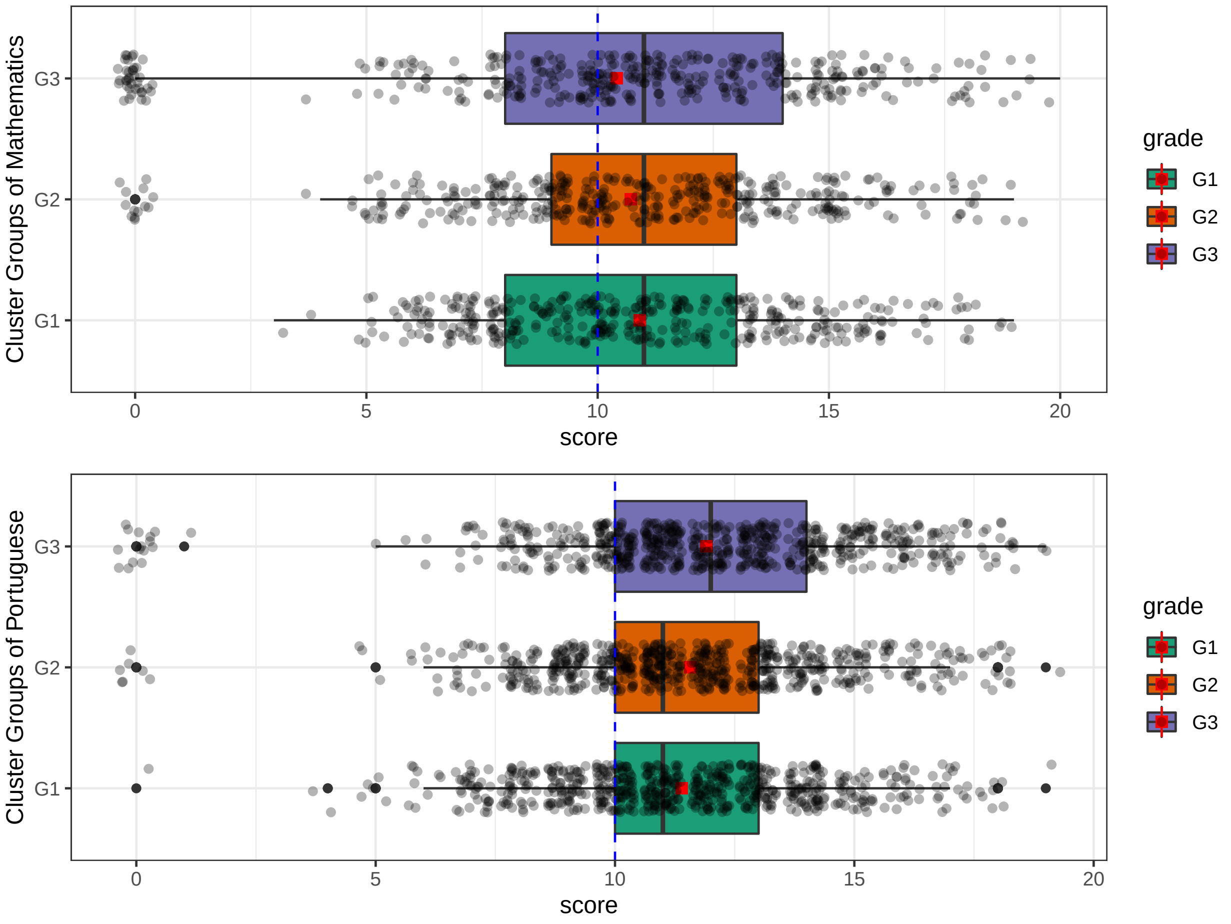

Figure 1: Frequency of Grades

From the figure above the distribution of the grades can be seen. It is clear that many students have a score below 10 which means that they failed their exam. In the following we are going to identify the factors with the highest influence on the failureof the exams.

2.1 Introduce the Analysis

The decision tree method is a powerful and popular predictive technique that is used for both *classification* and *regression*. There are many methodologies for constructing regression trees but in this report the classification and regression tree approach by *Rpart* and *caret* packages is used.

3 Results

Decision tree analysis will be used to predict the students failure or pass based on certain important variables as chosen by the algorithm due to the correlation and co linearity exhibited by the variables.

3.1 Modeling data sets

Classification methodology was used for these particular data frames and the the response variable **grade** is modeled as a binary variable.

```r
# Math
mat$final <- factor(ifelse(mat$G3 > 9, 1, 0), labels = c("Fail", "Pass"))
mat$G3 <- NULL

# Port
por$final <- factor(ifelse(por$G3 > 9, 1, 0), labels = c("Fail", "Pass"))
por$G3 <- NULL
```

After that we split the data sets as *train* and *test*.

```r
# Math - Training and test data
set.seed(42)
intrain = createDataPartition (y = mat [["final"]], p =0.75, list= FALSE )
train_M <- mat [intrain,]
test_M <- mat [-intrain,]
# Port - Training and test data
set.seed(42)
intrain = createDataPartition (y = por [["final"]], p =0.75, list= FALSE )
train_P <- por [intrain,]
test_P <- por [-intrain,]
```

To split the dataset we use a split ratio equal to 0.75, as our datasets does not have so many observations and we need an appropriate number of rows in the test set to validate our modeling.

```r
dim(train_M)
```

```
## [1] 297  33
```

```r
dim(test_M)
```

```
## [1] 98 33
```

```r
dim(train_P)
```

```
## [1] 487  33
```

```r
dim(test_P)
```

```
## [1] 162  33
```

The observations of each *train* and *test* data sets is clear.

3.2 Create the Regression Tree

To obtain the most accurate result the best *cp* value is the one that minimize the prediction error RMSE (root mean squared error). For these models we consider the *cp* as 0.01.

```
# Math tree
tree_M <- rpart(final ~ .,
              data = train_M,
              method = "class", cp=0.01)

imp_M <- varImp(tree_M)
rownames(imp_M)[order(imp_M$Overall, decreasing=TRUE)]
```

```
##  [1] "G2"         "G1"         "failures"   "absences"   "Fjob"
##  [6] "reason"     "goout"      "age"        "Medu"       "famrel"
## [11] "guardian"   "Walc"       "school"     "sex"        "address"
## [16] "famsize"    "Pstatus"    "Fedu"       "Mjob"       "traveltime"
## [21] "studytime"  "schoolsup"  "famsup"     "paid"       "activities"
## [26] "nursery"    "higher"     "internet"   "romantic"   "freetime"
## [31] "Dalc"       "health"
```

```
# Port tree
tree_P <- rpart(final ~ .,
              data = train_P,
              method = "class", cp=0.01)

imp_P <- varImp(tree_P)
rownames(imp_P)[order(imp_P$Overall, decreasing=TRUE)]
```

```
##  [1] "G2"         "G1"         "higher"     "failures"   "school"
##  [6] "Mjob"       "absences"   "age"        "Walc"       "famrel"
## [11] "Dalc"       "traveltime" "sex"        "address"    "famsize"
## [16] "Pstatus"    "Medu"       "Fedu"       "Fjob"       "reason"
## [21] "guardian"   "studytime"  "schoolsup"  "famsup"     "paid"
## [26] "activities" "nursery"    "internet"   "romantic"   "freetime"
## [31] "goout"      "health"
```

We create the tree using all the variables and rank the variables in terms of importance to figure out the variables used by the decision tree algorithm to predict the final outcome of pass or fail. According to the results, Grades in 2nd and 1st exam are key predictors followed by past class failures, absences and time spending factors. It is also interesting to note that parents job and family relationships have a great influence on the grades.

We use *printcp* function to understand the variables that actually were used in construction of tree.

```
# Math cp
printcp(tree_M)
```

```
##
## Classification tree:
## rpart(formula = final ~ ., data = train_M, method = "class",
##     cp = 0.01)
##
## Variables actually used in tree construction:
## [1] Fjob G2
##
## Root node error: 98/297 = 0.32997
```

```
##
## n= 297
##
##         CP nsplit rel error  xerror     xstd
## 1 0.744898      0   1.00000 1.00000 0.082687
## 2 0.017007      1   0.25510 0.25510 0.048826
## 3 0.010000      4   0.20408 0.28571 0.051387
```

```
# Por cp
printcp(tree_P)
```

```
##
## Classification tree:
## rpart(formula = final ~ ., data = train_P, method = "class",
##     cp = 0.01)
##
## Variables actually used in tree construction:
## [1] G1   G2   Mjob
##
## Root node error: 75/487 = 0.154
##
## n= 487
##
##         CP nsplit rel error  xerror     xstd
## 1 0.613333      0   1.00000 1.00000 0.106207
## 2 0.031111      1   0.38667 0.38667 0.069632
## 3 0.010000      4   0.29333 0.40000 0.070745
```

By understanding the models we saw the number of calculated cp - Complexity parameter can be reduced to control the tree growth by methods such as *Pruning* to make the model more accurate. However due to the size of the data sets and reliable outcome, we accept the results. The tree logic is as below where only “Parent’s job and Grades in 2nd and 1st Exam” are used as variables by the tree based on correlation and co linearity between some of the other variables.

```
# Math tree
prp(tree_M, extra=4,varlen = 4)
```

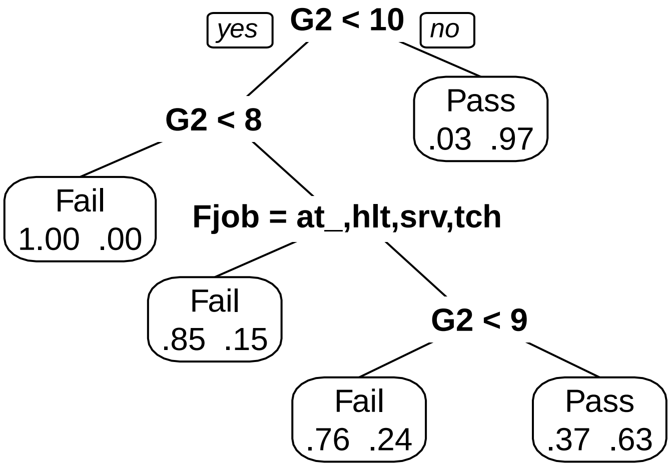

Figure 2: Decision Tree of Math

```
# Port tree
prp(tree_P, extra=4,varlen = 4)
```

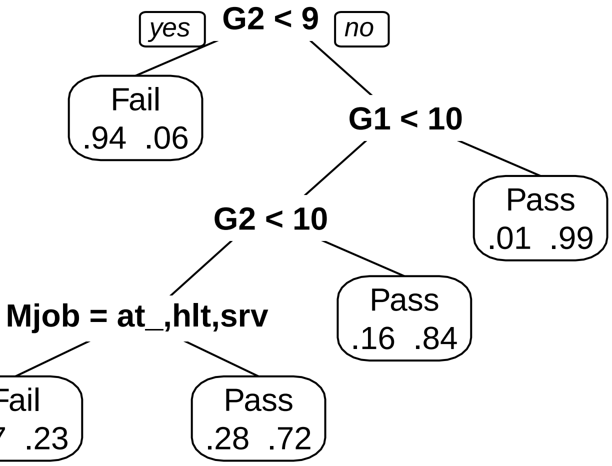

Figure 3: Decision Tree of Port

It is much more clear with visualizing the trees that the classification rate at the node, expressed as the number of correct classifications and the number of observations in the node.

3.3 Prediction

In the following part we do prediction and assessing classification model performance using *confusionMatrix* by the test data sets.

```r
# Math
confusionMatrix(predict(tree_M ,test_M, type = "class" ),test_M$final)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction Fail Pass
##       Fail   27    2
##       Pass    5   64
##                                           
##                Accuracy : 0.9286          
##                  95% CI : (0.8584, 0.9708)
##     No Information Rate : 0.6735          
##     P-Value [Acc > NIR] : 1.539e-09       
##                                           
##                   Kappa : 0.8336          
##                                           
##  Mcnemar's Test P-Value : 0.4497          
##                                           
##             Sensitivity : 0.8438          
##             Specificity : 0.9697          
##          Pos Pred Value : 0.9310          
##          Neg Pred Value : 0.9275          
##              Prevalence : 0.3265          
##          Detection Rate : 0.2755          
##    Detection Prevalence : 0.2959          
##       Balanced Accuracy : 0.9067          
##                                           
##        'Positive' Class : Fail            
## 
```

```r
# Port
confusionMatrix(predict(tree_P ,test_P, type = "class" ),test_P$final)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction Fail Pass
##       Fail   19    5
##       Pass    6  132
##                                           
##                Accuracy : 0.9321          
##                  95% CI : (0.8818, 0.9656)
##     No Information Rate : 0.8457          
##     P-Value [Acc > NIR] : 0.0006884       
##                                           
##                   Kappa : 0.7355          
##                                           
##  Mcnemar's Test P-Value : 1.0000000       
##                                           
##             Sensitivity : 0.7600          
##             Specificity : 0.9635          
##          Pos Pred Value : 0.7917          
##          Neg Pred Value : 0.9565
##              Prevalence : 0.1543
##          Detection Rate : 0.1173
##    Detection Prevalence : 0.1481
##       Balanced Accuracy : 0.8618
##
##        'Positive' Class : Fail
##
```

By the output of the confusion matrix of both data sets we see great accuracy for the models. For the **Mathematics** 92 percent and for **Portuguese** 93 percent of accuracy is obtained.

4 Conclusions

Education is a key factor affecting long term economic progress. For maximizing the performance of students, the factors influencing the final grade should be identified and evaluated to control them. As expected, the student evaluations have a high impact in the models. For instance, G2 and G1 are the most important features for passing or failing the final exam. Nevertheless, an analysis to knowledge provided by the best predictive models has shown that, in some cases, there are other relevant features, such as: school related (e.g. number of absences, reason to choose school, extra educational school support), demographic (e.g. student's age, parent's job and education) and social (e.g. going out with friends, alcohol consumption) variables. More research is also needed (e.g. sociological studies) in order to understand why and how some variables (e.g. reason to choose school, parent's job or alcohol consumption) affect student performance. For this, the analysis could be further deepened and additional influencing factors can be analyzed.

The models have described an obvious relationship between most recent test score, G2, but has also identified the father's job, Fjob, as being a useful indicator which may not have been revealed in a human expert analysis. Based on this first findings, the school can take initial action by offering extra tutoring to students who have received a poor grade in G2. In addition the influence of the fathers job can be checked further to find additional correlations to aid students based on their circumstances.

5 References

[1] R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[2] Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

[3] Hao Zhu (2020). kableExtra: Construct Complex Table with 'kable' and Pipe Syntax. R package version 1.2.1. https://CRAN.R-project.org/package=kableExtra

[4] Erich Neuwirth (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1-2. https://CRAN.R-project.org/package=RColorBrewer

[5] Baptiste Auguie (2017). gridExtra: Miscellaneous Functions for "Grid" Graphics. R package version 2.3. https://CRAN.R-project.org/package=gridExtra

[6] Max Kuhn (2020). caret: Classification and Regression Training. R package version 6.0-86. https://CRAN.R-project.org/package=caret

[7] Terry Therneau and Beth Atkinson (2019). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15. https://CRAN.R-project.org/package=rpart

[8] Stephen Milborrow (2020). rpart.plot: Plot ‘rpart’ Models: An Enhanced Version of ‘plot.rpart’. R package version 3.0.9. https://CRAN.R-project.org/package=rpart.plot

[9] Jarek Tuszynski (2020). caTools: Tools: Moving Window Statistics, GIF, Base64, ROC AUC, etc. R package version 1.18.0. https://CRAN.R-project.org/package=caTools