Student Performance Prediction. Capstone Data Science Project.

Belen Ibarra

2022

Contents

Chapter 1

Introduction

In this project we will try to predict if a student passed or failed the mathematics exam.

Many factors could influence student performance and in this dataset it is obtained a lot of information about the learning environment that could be affecting their learning process.

Machine learning techniques used in the student performance could help identify those more vulnerable to failure so that educational programs could be applied to them and this way improve the education system.

Chapter 2

About the Dataset

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features and it was collected by using school reports and questionnaires. .

Chapter 3

Attribute Information:

Attributes for student-mat.csv (Math course) datasets:

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 â€" 5th to 9th grade, 3 â€" secondary education or 4 â€" higher education)
8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€" 5th to 9th grade, 3 â€" secondary education or 4 â€" 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) 31(*). G1 - first period grade (numeric: from 0 to 20) 31(*).* G2 - second period grade (numeric: from 0 to 20) 32(*). G3 - final grade (numeric: from 0 to 20, output target)

- these grades are related with the course subject, Math.

Chapter 4

Model Evaluation - Data Analysis

In this case, we will use the accuracy metric to evaluate the performance of the model. The accuracy derives from the confusion matrix where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Accuracy is defined by this formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

4.1 Data Preparation

First, we install all needed libraries

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org"
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.or
if(!require(dslabs)) install.packages("dslabs", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(RColorBrewer)) install.packages("RColorBrewer", repos = "http://cran.us.r-projec
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(MLmetrics)) install.packages("MLmetrics", repos = "http://cran.us.r-project.org"
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-projec
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
if(!require(rpart.plot)) install.packages("rpart.plot", repos = "http://cran.us.r-project.or
if(!require(MLmetrics)) install.packages("MLmetrics", repos = "http://cran.us.r-project.org"
if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")
```

Now we load the required libraries

```
library(tidyverse)
library(dslabs)
library(caret)
library(ggplot2)
library(RColorBrewer)
library(corrplot)
library(data.table)
library(e1071)
library(randomForest)
library(rpart)
library(rpart.plot)
library(MLmetrics)
```

Now we load the dataset, it is important to notice I will only be using the dataset correspondent to the math results and not the portuguese test scores.

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

We add a new column to make our G3 scores binary, presenting 1 if the student passed(hence G3>9) or 0 if they did not pass (G3<9)

```
d1<-d1%>% mutate(pass=ifelse(d1$G3>9,1,0))
```

Now is time to encode the categorical features as factors and integers to numeric

```
d1$famsize<-factor(d1$famsize)
d1$address<-factor(d1$address)
d1$Pstatus<-factor(d1$Pstatus)
d1$Medu<-factor(d1$Medu)
d1$Fedu<-factor(d1$Fedu)
d1$Mjob<-factor(d1$Mjob)
d1$Fjob<-factor(d1$Fjob)
d1$schoolsup<-factor(d1$schoolsup)
d1$famsup<-factor(d1$famsup)
d1$nursery<-factor(d1$nursery)
d1$pass<-factor(d1$pass)
d1$age <- as.numeric(d1$age)
d1[c("traveltime","studytime","failures","famrel","freetime","goout","Dalc","Walc","health",
```

We will delete the G1 and G2 scores given that we are not going to take them into consideration in our machine learning approach.

```
d1$G1<- NULL
d1$G2<- NULL
```

We will use G3 to try and make a correlation plot and notice the most important variables

4.2 Exploratory Data Anlaysis

Now it's time for our first Data Exploration, we start by looking at the class and some more details of our dataset.

```
str(d1)
```

```
## 'data.frame':    395 obs. of  32 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : num  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      : Factor w/ 5 levels "0","1","2","3",..: 5 2 2 5 4 5 3 5 4 4 ...
##  $ Fedu      : Factor w/ 5 levels "0","1","2","3",..: 5 2 2 3 4 4 3 5 3 5 ...
##  $ 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    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: num  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : num  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : num  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      : chr  "no" "no" "yes" "yes" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : num  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : num  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : num  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : num  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : num  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : num  3 3 3 5 5 5 3 1 1 5 ...
```

```
##  $ absences  : num  6 4 10 2 4 10 0 6 0 0 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
##  $ pass      : Factor w/ 2 levels "0","1": 1 1 2 2 2 2 2 1 2 2 ...
```

For a first glimpse we will see the mean of students who passed

```
mean(d1$pass==1)
```

```
## [1] 0.6708861
```

We also observe our dataset contains 395 observations with 32 variables.

```
head(d1)
```

```
##   school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob     reason
## 1     GP   F  18       U     GT3       A    4    4  at_home  teacher     course
## 2     GP   F  17       U     GT3       T    1    1  at_home    other     course
## 3     GP   F  15       U     LE3       T    1    1  at_home    other      other
## 4     GP   F  15       U     GT3       T    4    2   health services       home
## 5     GP   F  16       U     GT3       T    3    3    other    other       home
## 6     GP   M  16       U     LE3       T    4    3 services    other reputation
##   guardian traveltime studytime failures schoolsup famsup paid activities
## 1   mother          2         2        0       yes     no   no         no
## 2   father          1         2        0        no    yes   no         no
## 3   mother          1         2        3       yes     no  yes         no
## 4   mother          1         3        0        no    yes  yes        yes
## 5   father          1         2        0        no    yes  yes         no
## 6   mother          1         2        0        no    yes  yes        yes
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1     yes    yes       no       no      4        3     4    1    1      3
## 2      no    yes      yes       no      5        3     3    1    1      3
## 3     yes    yes      yes       no      4        3     2    2    3      3
## 4     yes    yes      yes      yes      3        2     2    1    1      5
## 5     yes    yes       no       no      4        3     2    1    2      5
## 6     yes    yes      yes       no      5        4     2    1    2      5
##   absences G3 pass
## 1        6  6    0
## 2        4  6    0
## 3       10 10    1
## 4        2 15    1
## 5        4 10    1
## 6       10 15    1
```

```
ncol(d1)
```

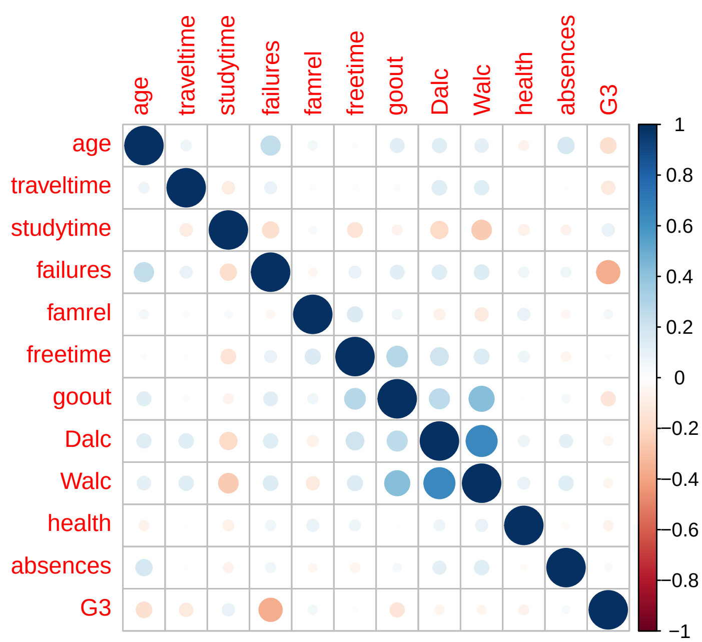
```
## [1] 32
```

```
nrow(d1)
```

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

We use corrplot to see which variables have a more relevant correlation with G3. This correlation graph is made to identify which variables are the most significant either positively or negatively when it comes to define whether a student had a high score or a low score. This is why we focus on the points related to G3, the total score in the math test. The color blue means the relation is positive and the color red means negative. All these variables are numeric. we see that these are age, failures,goout, and traveltime negatively and studytime and family relations positively.

```
corrplot(cor(d1[,unlist(lapply(d1,is.numeric))]))
```



We will not be using the G3 variable anymore, because we are only interested if the studen passed or not and not the particular score, so we transform it to a binary variable called passed to make the process easier.

```
d1$G3<- NULL
```

We will partition our data into train set and a validation set. For this purpose we will set the train set to be 75% of our dataset and the remaining 25% to be the validation set.

```
#Partitioning in validation and train

set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = d1$pass, times = 1, p = 0.75, list = FALSE)
train <- d1[test_index,]
validation <- d1[-test_index,]
nrow(validation)
```

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

```
nrow(train)
```

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

We have to divide the train database in train_set and test_set . train_set is used to create the models and test_set is used to prove how nice those models works, and the best among them, is used to test it with the validation database.

```
#Partitioning the train into train set and test set
trainingindex<- createDataPartition(train$pass,times=1,p=0.8,list = FALSE)
training_set<- train[trainingindex,]
test_set<- train[-trainingindex,]
nrow(training_set)
```

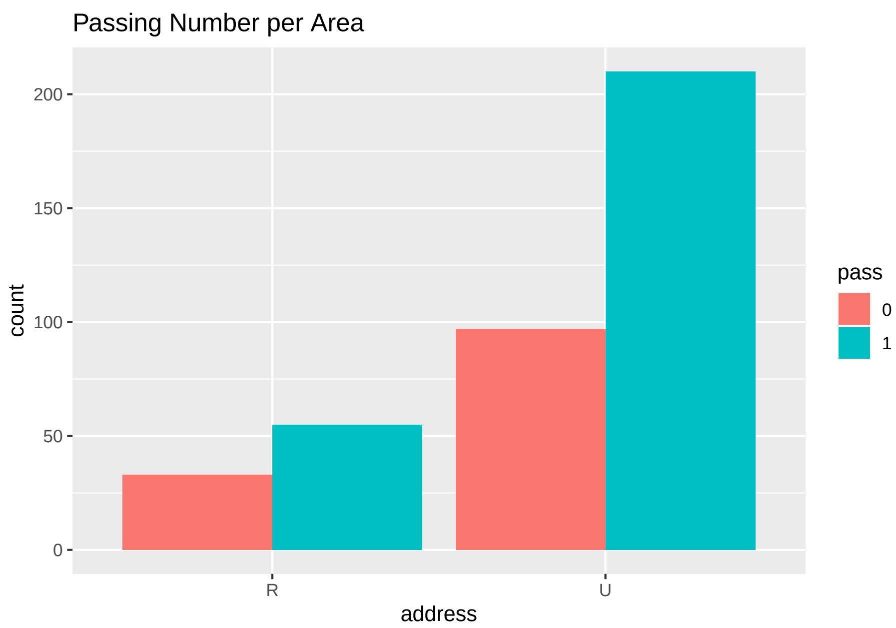
```
## [1] 239
```

```
nrow(test_set)
```

```
## [1] 58
```

To continue the analysis, we are now going to see the ammount of people that passed based on their arear, where R means Rural and U means Urban. We observe the means of the ones that passed and can see the urban area has a bigger ratio of students that passed.

```r
d1 %>% ggplot(aes(address,fill=pass))+geom_bar(position="dodge")+ggtitle("Passing Number per
```

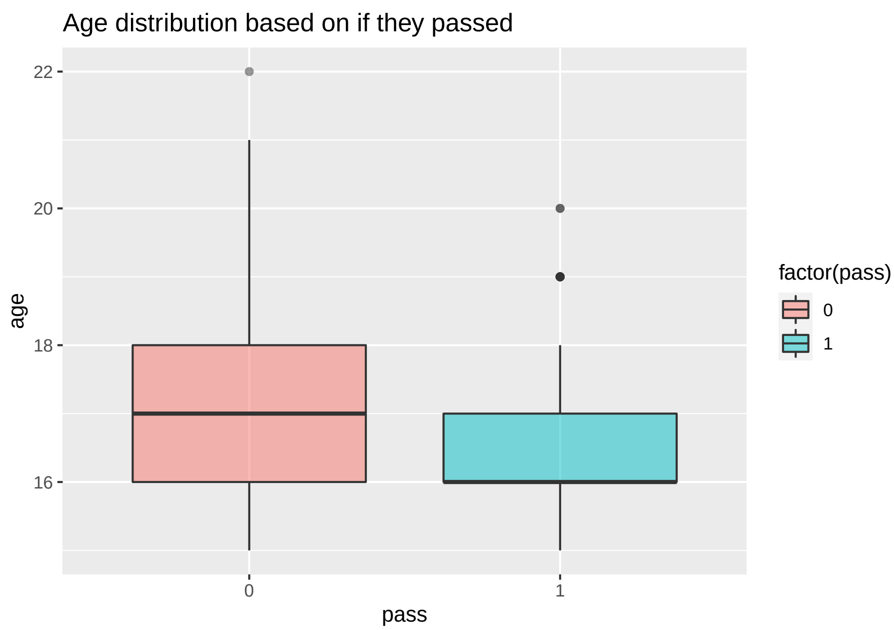



```r
d1 %>% group_by(address) %>% summarise(porcentagepass=mean(pass=="1")*100)
```

```
## # A tibble: 2 x 2
##   address porcentagepass
##   <fct>            <dbl>
## 1 R                 62.5
## 2 U                 68.4
```

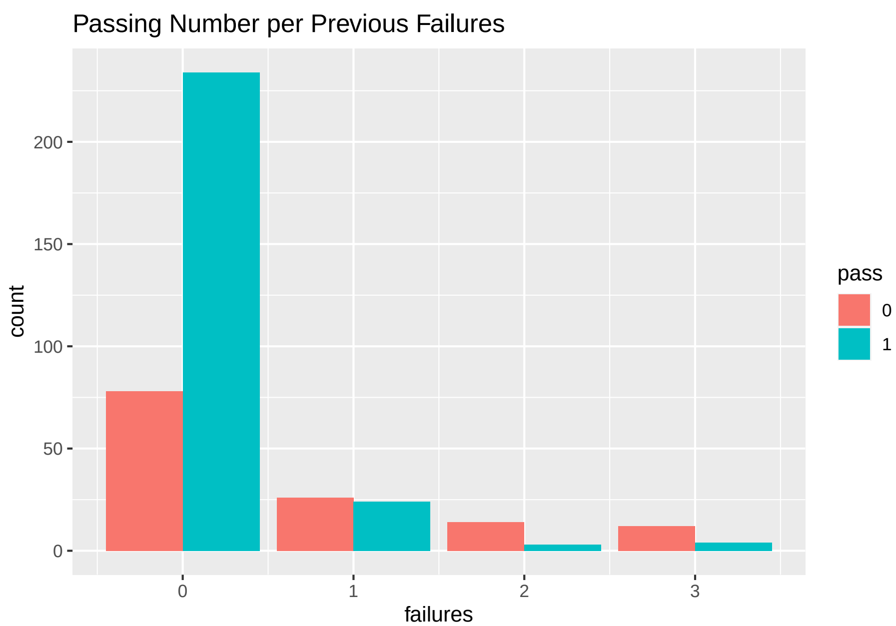
Next is a boxplot representing the age distribution for those who passed and those who did not. We notice that those who passed have a lower age distribution.

```r
ggplot(d1,aes(pass,age)) + geom_boxplot(aes(fill=factor(pass)),alpha=0.5) + ggtitle("Age dis
```



Now we see that there is a relation between having failed in the past and failing now.

```
d1 %>% ggplot(aes(failures,fill=pass))+geom_bar(position="dodge")+ggtitle("Passing Number pe
```



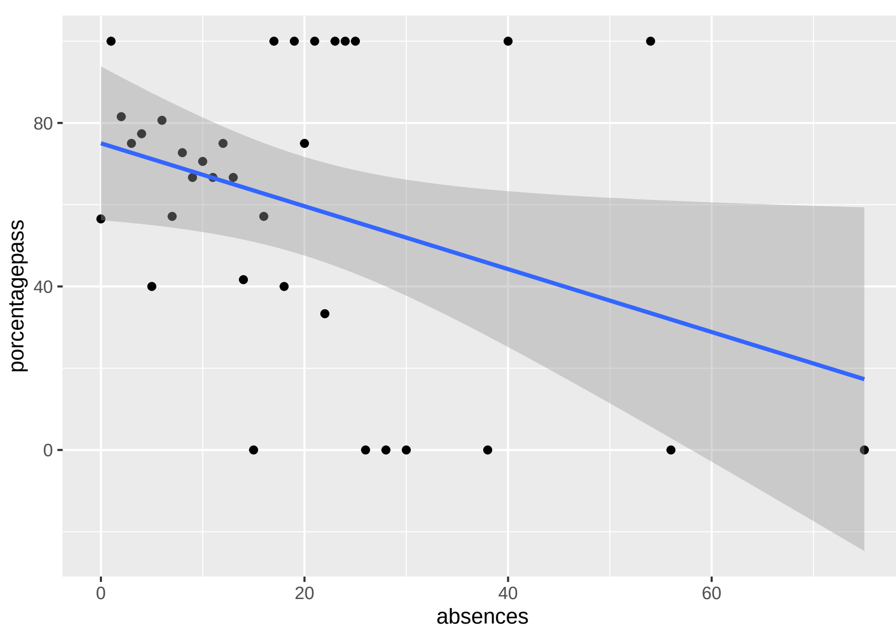
```
d1 %>% group_by(failures) %>% summarise(porcentagepass=mean(pass=="1")*100)
```

```
## # A tibble: 4 x 2
##   failures porcentagepass
##      <dbl>          <dbl>
## 1        0           75
## 2        1           48
## 3        2           17.6
## 4        3           25
```

In the next graph we see a correlation plot with the absences and the percentage of people that passed, shows that the more absences the less people have passed the exam.

```
table1<- d1 %>% group_by(absences) %>% summarise(porcentagepass=mean(pass=="1")*100)
table1 %>% ggplot(aes(absences,porcentagepass))+geom_point()+geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
lm(table1$porcentagepass~table1$absences)
```

```
##
```

```
## Call:
## lm(formula = table1$porcentagepass ~ table1$absences)
##
## Coefficients:
##     (Intercept)  table1$absences
##         75.0223          -0.7695
```

Chapter 5

Modeling

When talking about a data science project, there are mainly two types of work that can be done: regression and classification. Since this project is based on a binomial classification problem, a linear model approach may not be useful. However, we will use the Naive Bayes approach, the Support Vector Machine model and the Random Forest.

5.1 Naive Bayes

We use Naive Bayes algorithm to predict student performance. Naive Bayes classification is a simple but effective algorithm; it is faster compared to many other iterative algorithms; it does not need feature scaling; and its foundation is the Bayes Theorem.

However, Naive Bayes is based on the assumption that conditional probability of each feature given the class is independent of all the other features. The assumption of independent conditional probabilities means the features are completely independent of each other. By assuming the independence assumption of all the features, let's fit a naive bayes model to our training data.

```
#Naive Bayes Method

# Fitting Naive Bayes to the Training set
classifier_NB = naiveBayes(pass ~ ., data = training_set)

# Predicting the Validation set results
y_pred_NB = predict(classifier_NB, newdata = test_set[,-which(names(test_set)=="pass")])

# Checking the prediction accuracy
```

```r
confusionmatrix<- table(test_set$pass, y_pred_NB) # Confusion matrix
normalizedcm<- confusionmatrix/sum(confusionmatrix)
normalizedcm
```

```
##    y_pred_NB
##              0          1
##   0 0.06896552 0.25862069
##   1 0.03448276 0.63793103
```

```r
error <- mean(test_set$pass != y_pred_NB) # Misclassification error
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.7069"
```

5.2 Support Vector MAchines (SVM)

Secondly, we use Support Vector Machines (SVM) for classification. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

```r
#SVM Model
classifier_SVM = svm(pass ~ .,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'linear')
# Predicting the Validation set results
y_pred_SVM = predict(classifier_SVM, newdata = test_set[,-which(names(test_set)=="pass")])

# Checking the prediction accuracy
table(test_set$pass, y_pred_SVM) # Confusion matrix
```

```
##    y_pred_SVM
##      0  1
##   0  6 13
##   1  4 35
```

```r
error <- mean(test_set$pass != y_pred_SVM) # Misclassification error
paste('Accuracy',round(1-error,4))
```

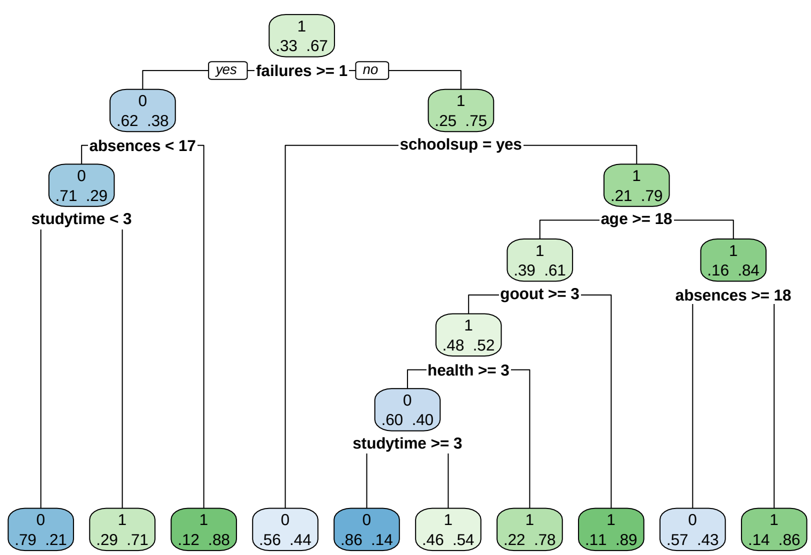
```
## [1] "Accuracy 0.7069"
```

5.3 Random Forest

Random Forest is a prowerful machine learning algorithm which holds a relatively high classification accuracy. Random forests improve predictive accuracy by generating a large number of bootstrapped trees (based on random samples of variables)

```r
#Random Forest
# Fitting Decision Tree Classification Model to the Training set
classifier_rf = rpart(pass ~ ., data = training_set, method = 'class')

# Tree Visualization
rpart.plot(classifier_rf, extra=4)
```




```r
# Predicting the Validation set results
y_pred_rf = predict(classifier_rf, newdata = test_set[,-which(names(test_set)=="pass")], typ
```

```r
# Checking the prediction accuracy
table(test_set$pass, y_pred_rf) # Confusion matrix
```

```
##    y_pred_rf
##      0  1
##   0 11  8
##   1  7 32
```

```r
error <- mean(test_set$pass != y_pred_rf) # Misclassification error
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.7414"
```

Chapter 6

Results

We will now test our accuracy and f1 score using our validation dataset

```
y_pred_NB = predict(classifier_NB, newdata = validation[,-which(names(validation)=="pass")]
error_NB <- mean(validation$pass != y_pred_NB)
paste('Accuracy',round(1-error_NB,4))
```

```
## [1] "Accuracy 0.7551"
```

```
y_pred_SVM = predict(classifier_SVM, newdata = validation[,-which(names(validation)=="pass"
error_SVM <- mean(validation$pass != y_pred_SVM)
paste('Accuracy',round(1-error_SVM,4))
```

```
## [1] "Accuracy 0.7041"
```

```
y_pred_rf = predict(classifier_rf, newdata = validation[,-which(names(validation)=="pass")]
error_rf <- mean(validation$pass != y_pred_rf)
paste('Accuracy',round(1-error_rf,4))
```

```
## [1] "Accuracy 0.6224"
```

```
Accuracy<- array(c(1-error_NB,1-error_SVM,1-error_rf))
method<- array(c("Naive Bayes","SVM","Random Forest"))
results<- data.frame(method,Accuracy)
results
```

```
##          method  Accuracy
## 1   Naive Bayes 0.7551020
## 2           SVM 0.7040816
## 3 Random Forest 0.6224490
```

Chapter 7

References

1. P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7.

2. Data collected from https://archive.ics.uci.edu/ml/datasets/Student+Performance.

Chapter 8

Conclusion

It is very interesting to see that many social, demographic, economic and other variables are useful to predict student performance. This could be used by the teaching environment to adjust their efforts to provide the right conditions so that a student could develop the right skills and avoid failure.

With this work we can conclude that the most effective way to predict student performance was Naive Bayes with an approximate 75.5% accuracy.

This project will be very useful to apply in other countries and scenarios. It will also be interesting to select less variables for prediction and use the most important ones.

I am very pleased with the experience of making this machine learning project and happy to acknowledge the skills.