

Introduction to statistical learning: Project Proposal

Student Performance Data Set

Team Members:

Sandeep Reddy Salkuti (16296868)

Sumanth Medavarapu(16295321)

PardhaSaradhi Ramineni(16300893)

SaiChand Patchala(16292087)

Poojasree Reddem(16296515)

Project Description:

We are willing to find student final grades based on considering different attributes. This data approaches 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. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). The two datasets were modeled under binary/five-level classification and regression tasks. The target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. It is more difficult to predict G3 without G2 and G1, but such prediction is much more useful.

Goal:

The classification goal is to predict student performance in secondary education (high school)

Dataset:

It is the Student Performance dataset uploaded originally in the UCI Machine Learning Repository. The dataset gives information about a student grades, demographic, social and school related features.

Loading libraries:

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for loading libraries and data.

```
1 #Data and packages setup
2 library(dplyr)
3 library(readr)
4 library(stringr)
5 library(ggplot2)
6 library(carpet)
7 library(rpart)
8 library(rattle)
9 library(randomForest)
10 library(outliers)
11 library(pander)
12 #loading the data
13 dat <- read.csv("E:/Fall2020/ISL-1/Project/student-mat.csv", sep = ";")
14 #checking for dimensions
15 dim(dat)
```
- Console:** Shows the execution of the code, including warnings about package versions and masked objects.

```
The downloaded binary packages are in
C:\Users\SandeepReddy\AppData\Local\Temp\Rtmp0UqdPR\downloaded_packages
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':
  filter, lag

The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union

warning message:
package 'dplyr' was built under R version 3.6.3
> library(readr)
> library(stringr)
> library(ggplot2)
warning message:
package 'ggplot2' was built under R version 3.6.3
> library(carpet)
```
- Environment:** Lists loaded objects in the Global Environment, including data frames like `bag.car`, `bag.hitters`, `best_set`, `best_set_summ`, `boost.hitters`, `Boston`, `Car.test`, and `Car.train`.

Loading dataset and viewing the dimensions :

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for loading the dataset and checking its dimensions.

```
12 #loading the data
13 dat <- read.csv("E:/Fall2020/ISL-1/Project/student-mat.csv", sep = ";")
14 #checking for dimensions
15 dim(dat)
16 dat
17 #
```
- Console:** Shows the execution of the code, displaying the dimensions of the dataset and a preview of the data.

```
> dat <- read.csv("E:/Fall2020/ISL-1/Project/student-mat.csv", sep = ";")
> dim(dat)
[1] 395 33
> dat
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime
1	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2
2	GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1
3	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1
4	GP	F	15	U	GT3	T	4	2	health services	other	home	mother	1
5	GP	F	16	U	GT3	T	3	3	other	other	home	father	1
6	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother	1
7	GP	M	16	U	LE3	T	2	2	other	other	home	mother	1
8	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother	2
9	GP	M	15	U	LE3	A	3	2	services	other	home	mother	1
10	GP	M	15	U	GT3	T	3	4	other	other	home	mother	1
11	GP	F	15	U	GT3	T	4	4	teacher	health	reputation	mother	1
12	GP	F	15	U	GT3	T	2	1	services	other	reputation	father	3
13	GP	M	15	U	LE3	T	4	4	health services	other	course	father	1
14	GP	M	15	U	GT3	T	4	3	teacher	other	course	mother	2
15	GP	M	15	U	GT3	A	2	2	other	other	home	other	1
16	GP	F	16	U	GT3	T	4	4	health	other	home	mother	1
17	GP	F	16	U	GT3	T	4	4	services	services	reputation	mother	1


```
project_isl.R* x
Source on Save
Run
Source
19 #printing the internal structure of data
20 str(dat)
20:1 (Top Level)
R Script

> str(dat)
'data.frame': 395 obs. of 29 variables:
 $ sex      : Factor w/ 2 levels "F","M": 1 1 1 1 2 2 1 2 2 ...
 $ age      : Factor w/ 8 levels "15","16","17",...: 4 3 1 1 2 2 2 3 1 1 ...
 $ 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   : 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 : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 1 1 1 2 1 1 ...
 $ studytime : Factor w/ 4 levels "1","2","3","4": 2 2 2 3 2 2 2 2 2 2 ...
 $ failures  : Factor w/ 4 levels "0","1","2","3": 1 1 4 1 1 1 1 1 1 1 ...
 $ 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    : Factor w/ 5 levels "1","2","3","4",...: 4 5 4 3 4 5 4 4 4 5 ...
 $ freetime  : Factor w/ 5 levels "1","2","3","4",...: 3 3 3 2 3 4 4 1 2 5 ...
 $ goout     : Factor w/ 5 levels "1","2","3","4",...: 4 3 2 2 2 2 4 4 2 1 ...
 $ dalc      : Factor w/ 5 levels "1","2","3","4",...: 1 1 2 1 1 1 1 1 1 1 ...
 $ walc      : Factor w/ 5 levels "1","2","3","4",...: 1 1 3 1 2 2 1 1 1 1 ...
 $ health    : Factor w/ 5 levels "1","2","3","4",...: 3 3 3 5 5 5 3 1 1 5 ...
 $ absences  : int 6 4 10 2 4 10 0 6 0 0 ...
 $ performance: Factor w/ 2 levels "0","1": 1 1 1 2 1 2 2 1 2 2 ...
```

```
Console
Terminal x
~/
> summary(dat)
sex      age      famsize  Pstatus Medu  Fedu      Mjob      Fjob
F:208    16      :104    A: 41    0: 3    0: 2    at_home : 59    at_home : 20
M:187    17      : 98    LE3:114  T:354    1: 59    1: 82    health  : 34    health  : 18
          15      : 82          2:103    2:115    other   :141    other   :217
          18      : 82          3: 99    3:100    services:103    services:111
          19      : 24          4:131    4: 96    teacher : 58    teacher : 29
          20      : 3
          (other): 2
reason   guardian  traveltime studytime failures schoolsup famsup    paid
course  :145    father: 90    1:257    1:105    0:312    no :344    no :153    no :214
home    :109    mother:273    2:107    2:198    1: 50    yes: 51    yes:242    yes:181
other   : 36    other : 32    3: 23    3: 65    2: 17
reputation:105    4: 8    4: 27    3: 16

activities nursery  higher  internet  romantic  famrel  freetime  goout  dalc  walc  health
no :194    no : 81    no : 20    no : 66    no :263    1: 8    1: 19    1: 23    1:276    1:151    1: 47
yes:201    yes:314    yes:375    yes:329    yes:132    2: 18    2: 64    2:103    2: 75    2: 85    2: 45
          3: 68    3:157    3:130    3: 26    3: 80    3: 91
          4:195    4:115    4: 86    4: 9    4: 51    4: 66
          5:106    5: 40    5: 53    5: 9    5: 28    5:146

absences  performance
Min. : 0.000    0:232
1st Qu.: 0.000    1:163
Median : 4.000
Mean : 5.709
3rd Qu.: 8.000
Max. :75.000
```

Checking for any missing values

```
Console Terminal x
~/
> sapply(dat,function(x) sum(is.na(x)))
sex      age      famsize    Pstatus      Medu      Fedu      Mjob      Fjob
0        0        0          0          0        0        0        0
reason    guardian  traveltime  studytime  failures  schoolsup  famsup    paid
0        0        0          0          0        0        0        0
activities nursery    higher    internet  romantic  famrel    freetime  goout
0        0        0          0          0        0        0        0
dalc      walc      health    absences  performance
0        0        0          0          0
```

Feature engineering:

```
30 |
31 #Feature Engineering
32 dat <- dat %>% mutate(performance = ifelse(G1 > median(G1), 1, 0))

30:1 (Top Level) ↕

Console Terminal x
~/
> dat <- dat %>% mutate(performance = ifelse(G1 > median(G1), 1, 0))
> |
```

Data Cleaning:

```
19 #Data cleaning
20 dat <- dat[, -which(colnames(dat) %in% c("G1", "G2", "G3"))]
21 dat <- dat[, -which(colnames(dat) %in% c("address", "school"))]
22
23 for (i in c(1:27, 29)) {
24   dat[, i] <- as.factor(dat[, i])
25 }

19:1 (Top Level) ↕

Console Terminal x
~/
[1] Reached max / getOption("max.print") -- omitted 365 rows
> dat <- dat %>% mutate(performance = ifelse(G1 > median(G1), 1, 0))
> dat <- dat[, -which(colnames(dat) %in% c("G1", "G2", "G3"))]
> dat <- dat[, -which(colnames(dat) %in% c("address", "school"))]
> for (i in c(1:27, 29)) {
+   dat[, i] <- as.factor(dat[, i])
+ }
> |
```

Data Exploration:

```
26 #Data Exploration
27 cor.test(as.numeric(dat$studytime), as.numeric(dat$performance))
27:1 (Top Level) ↕
```

Console Terminal x

```
> cor.test(as.numeric(dat$studytime), as.numeric(dat$performance))

Pearson's product-moment correlation

data: as.numeric(dat$studytime) and as.numeric(dat$performance)
t = 1.9829, df = 393, p-value = 0.04807
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.0008669578 0.1962729650
sample estimates:
      cor
0.09952947

> |
```

Data Visualization:

```
Console Terminal x
```

```
> data_drop <- subset(dat, G3 == 0)
> data_stay <- subset(dat, G3 != 0)
> summary(data_drop)
```

school	sex	age	address	famsize	Pstatus	Medu	Fedu
GP:34	F:23	Min. :15.00	R:10	GT3:31	A: 2	Min. :1.000	Min. :1.000
MS: 4	M:15	1st Qu.:16.00	U:28	LE3: 7	T:36	1st Qu.:2.000	1st Qu.:1.000
		Median :17.00				Median :2.000	Median :2.000
		Mean :17.08				Mean :2.316	Mean :2.289
		3rd Qu.:18.00				3rd Qu.:3.000	3rd Qu.:3.000
		Max. :19.00				Max. :4.000	Max. :4.000

Mjob	Fjob	reason	guardian	traveltime	studytime
at_home : 9	at_home : 3	course :19	father: 8	Min. :1.000	Min. :1.000
health : 2	health : 0	home :12	mother:25	1st Qu.:1.000	1st Qu.:1.000
other :14	other :21	other : 1	other : 5	Median :1.000	Median :2.000
services: 9	services:11	reputation: 6		Mean :1.605	Mean :1.974
teacher : 4	teacher : 3			3rd Qu.:2.000	3rd Qu.:2.000
				Max. :4.000	Max. :4.000

failures	schoolsup	famsup	paid	activities	nursery	higher	internet	romantic
Min. :0.0000	no :37	no :15	no :30	no :17	no :10	no : 6	no : 8	no :18
1st Qu.:0.0000	yes: 1	yes:23	yes: 8	yes:21	yes:28	yes:32	yes:30	yes:20
Median :1.0000								
Mean :0.9211								
3rd Qu.:1.7500								
Max. :3.0000								

famrel	freetime	goout	dalc	walc	health
Min. :1.000	Min. :1.000	Min. :1.000	Min. :1.000	Min. :1.000	Min. :1.000
1st Qu.:3.000	1st Qu.:3.000	1st Qu.:2.000	1st Qu.:1.000	1st Qu.:1.000	1st Qu.:3.000
Median :4.000	Median :3.000	Median :3.000	Median :1.000	Median :2.000	Median :4.000
Mean :3.842	Mean :3.132	Mean :3.211	Mean :1.342	Mean :1.921	Mean :3.605
3rd Qu.:4.750	3rd Qu.:4.000	3rd Qu.:4.000	3rd Qu.:2.000	3rd Qu.:2.000	3rd Qu.:5.000
Max. :5.000	Max. :5.000	Max. :5.000	Max. :3.000	Max. :5.000	Max. :5.000

absences	G1	G2	G3
Min. :0	Min. : 4.000	Min. : 0.000	Min. :0
1st Qu.:0	1st Qu.: 6.000	1st Qu.: 0.000	1st Qu.:0
Median :0	Median : 7.000	Median : 5.000	Median :0
Mean :0	Mean : 7.526	Mean : 4.658	Mean :0
3rd Qu.:0	3rd Qu.: 9.000	3rd Qu.: 8.000	3rd Qu.:0
Max. :0	Max. :12.000	Max. :10.000	Max. :0

Console Terminal x

~/

> summary(data_stay)

school	sex	age	address	famsize	Pstatus	Medu	Fedu
GP:315	F:185	Min. :15.00	R: 78	GT3:250	A: 39	Min. :0.000	Min. :0.000
MS: 42	M:172	1st Qu.:16.00	U:279	LE3:107	T:318	1st Qu.:2.000	1st Qu.:2.000
		Median :17.00				Median :3.000	Median :3.000
		Mean :16.66				Mean :2.796	Mean :2.546
		3rd Qu.:18.00				3rd Qu.:4.000	3rd Qu.:3.000
		Max. :22.00				Max. :4.000	Max. :4.000

Mjob	Fjob	reason	guardian	traveltime	studytime
at_home : 50	at_home : 17	course :126	father: 82	Min. :1.000	Min. :1.000
health : 32	health : 18	home : 97	mother:248	1st Qu.:1.000	1st Qu.:1.000
other :127	other :196	other : 35	other : 27	Median :1.000	Median :2.000
services: 94	services:100	reputation: 99		Mean :1.431	Mean :2.042
teacher : 54	teacher : 26			3rd Qu.:2.000	3rd Qu.:2.000
				Max. :4.000	Max. :4.000

failures	schoolsup	famsup	paid	activities	nursery	higher	internet	romantic
Min. :0.0000	no :307	no :138	no :184	no :177	no : 71	no : 14	no : 58	no :245
1st Qu.:0.0000	yes: 50	yes:219	yes:173	yes:180	yes:286	yes:343	yes:299	yes:112
Median :0.0000								
Mean :0.2717								
3rd Qu.:0.0000								
Max. :3.0000								

famrel	freetime	goout	Dalc	walc	health
Min. :1.000	Min. :1.000	Min. :1.000	Min. :1.000	Min. :1.000	Min. :1.000
1st Qu.:4.000	1st Qu.:3.000	1st Qu.:2.000	1st Qu.:1.000	1st Qu.:1.000	1st Qu.:3.000
Median :4.000	Median :3.000	Median :3.000	Median :1.000	Median :2.000	Median :4.000
Mean :3.955	Mean :3.246	Mean :3.098	Mean :1.496	Mean :2.331	Mean :3.549
3rd Qu.:5.000	3rd Qu.:4.000	3rd Qu.:4.000	3rd Qu.:2.000	3rd Qu.:3.000	3rd Qu.:5.000
Max. :5.000	Max. :5.000	Max. :5.000	Max. :5.000	Max. :5.000	Max. :5.000

absences	G1	G2	G3
Min. : 0.000	Min. : 3.00	Min. : 5.00	Min. : 4.00
1st Qu.: 2.000	1st Qu.: 9.00	1st Qu.: 9.00	1st Qu.: 9.00
Median : 4.000	Median :11.00	Median :11.00	Median :11.00
Mean : 6.317	Mean :11.27	Mean :11.36	Mean :11.52
3rd Qu.: 8.000	3rd Qu.:14.00	3rd Qu.:14.00	3rd Qu.:14.00

Box Plot of G1, G2, G3

```

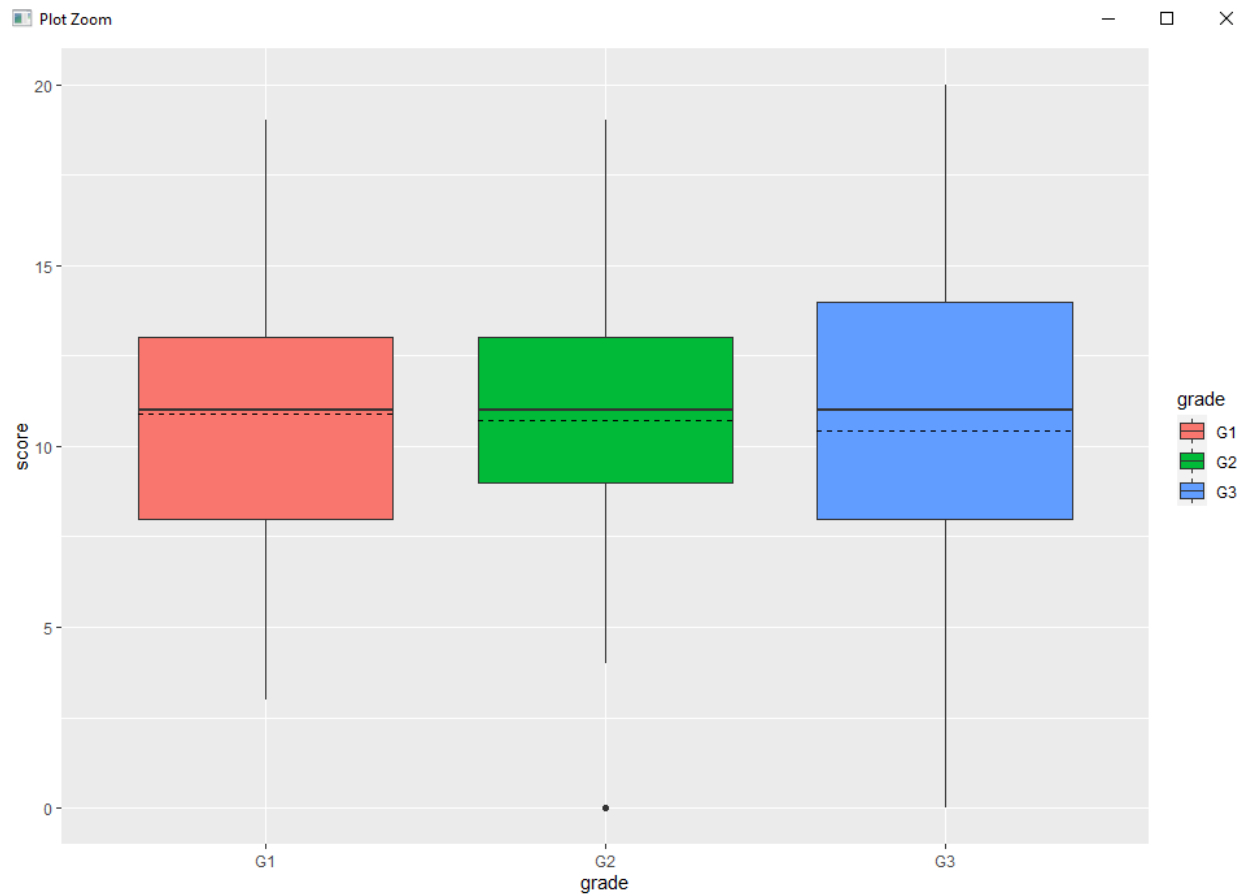
52 #box plot of G1, G2, G3
53 grades <- gather(dat,key = grade,value = score, G1,G2,G3)
54
55 ggplot(grades, aes(x=grade, y=score, fill=grade)) + geom_boxplot() +
56   stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..),
57     width = .75, linetype = "dashed")
58
55:1 (Top Level) ↕

```

```

Console Terminal
~/
> library("tidyr", lib.loc="C:/Users/SandeepReddy/Anaconda3/envs/rstudio/lib/R/library")
warning message:
package 'tidyr' was built under R version 3.6.3
> grades <- gather(dat,key = grade,value = score, G1,G2,G3)
> ggplot(grades, aes(x=grade, y=score, fill=grade)) + geom_boxplot() +
+   stat_summary(fun.y = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..),
+     width = .75, linetype = "dashed")
warning message:
`fun.y` is deprecated. Use `fun` instead.
> |

```



Visualization of relationship between G3 and other predictor variables

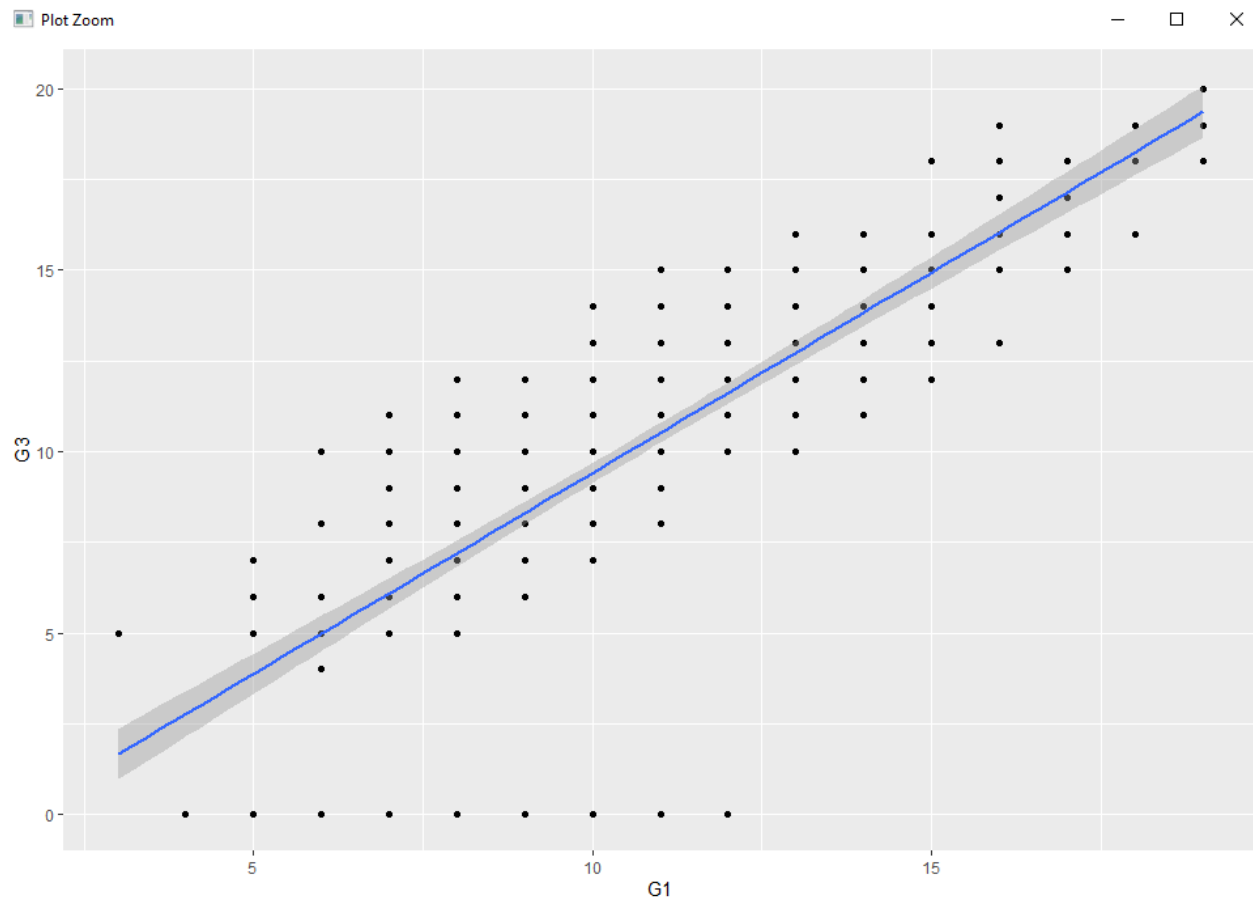
G3 versus G1

```
58 #visualization of G3 vs G1
59 ggplot(dat,aes(x=G1,y=G3)) +
60   geom_point() + geom_smooth(method = 'lm')
```

59:1 (Top Level) ⚡

Console Terminal ×

```
> ggplot(dat,aes(x=G1,y=G3)) +
+   geom_point() + geom_smooth(method = 'lm')
`geom_smooth()` using formula 'y ~ x'
>
```

The graph above shows strong linear relationship.

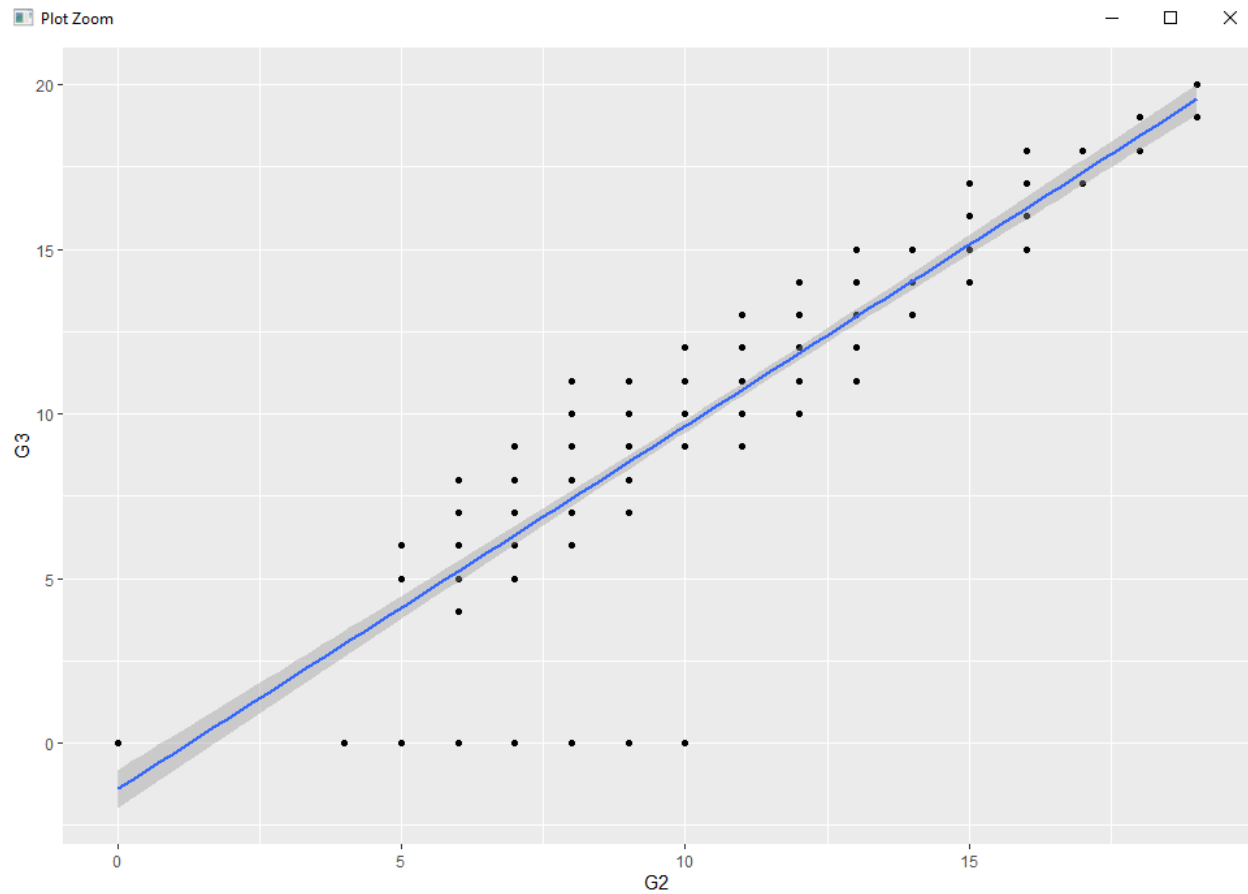
Visualization of G3 versus G2

```
61 #visualization of G3 vs G2
62 ggplot(dat,aes(x=G2,y=G3)) +
63   geom_point() + geom_smooth(method = 'lm')
```

62:1 (Top Level) ↕

Console **Terminal** ✕

```
> ggplot(dat,aes(x=G2,y=G3)) +
+   geom_point() + geom_smooth(method = 'lm')
`geom_smooth()` using formula 'y ~ x'
>
```



The above graph shows an even stronger relationship between G3 and G2

G3 versus study time

```

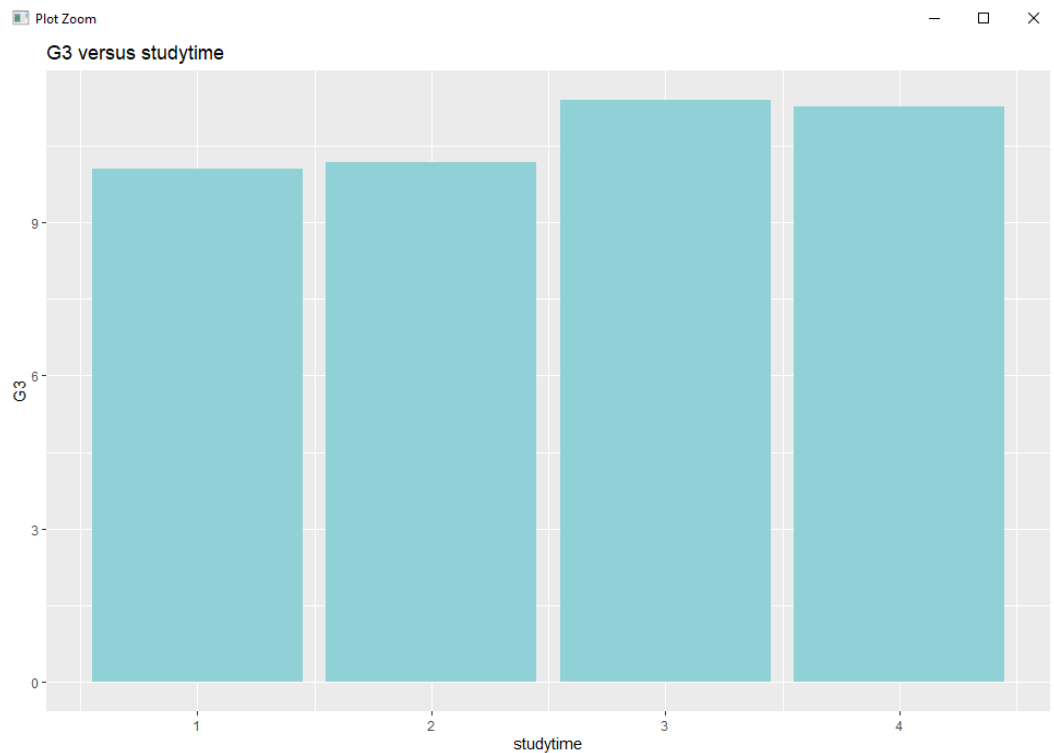
73 # G3 versus study time
74 ggplot(dat,aes(x=studytime,y=G3)) + stat_summary(fun.y="mean", geom="bar",fill="#8fd1d6") + ggtitle("G3 versus studytime")
75
76
71:1 (Top Level) ⚙ R Scrip

```

```

> ggplot(dat,aes(x=studytime,y=G3)) + stat_summary(fun.y="mean", geom="bar",fill="#8fd1d6") + ggtitle("G3 versus studytime")
warning message:
'fun.y' is deprecated. Use 'fun' instead.
>

```



From above graph students who study for longer time get better final grades than students who study for shorter time.

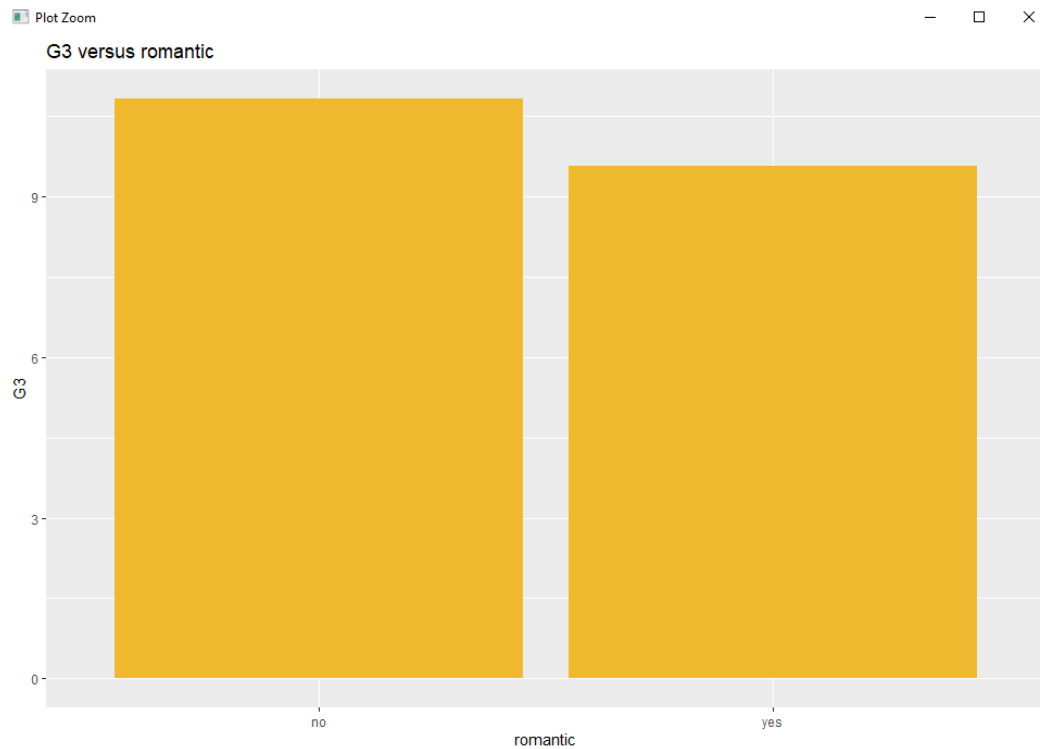
G3 versus romantic

```
77 #G3 versus romantic
78 ggplot(dat, aes(x=romantic, y=G3)) + stat_summary(fun.y="mean", geom="bar", fill="#efb92f") + ggtitle("G3 versus romantic")
79
80
```

76:1 (Top Level) ↕ R Scrip

Console Terminal x

```
~/
> ggplot(dat, aes(x=romantic, y=G3)) + stat_summary(fun.y="mean", geom="bar", fill="#efb92f") + ggtitle("G3 versus romantic")
Warning message:
'fun.y' is deprecated. Use 'fun' instead.
>
```



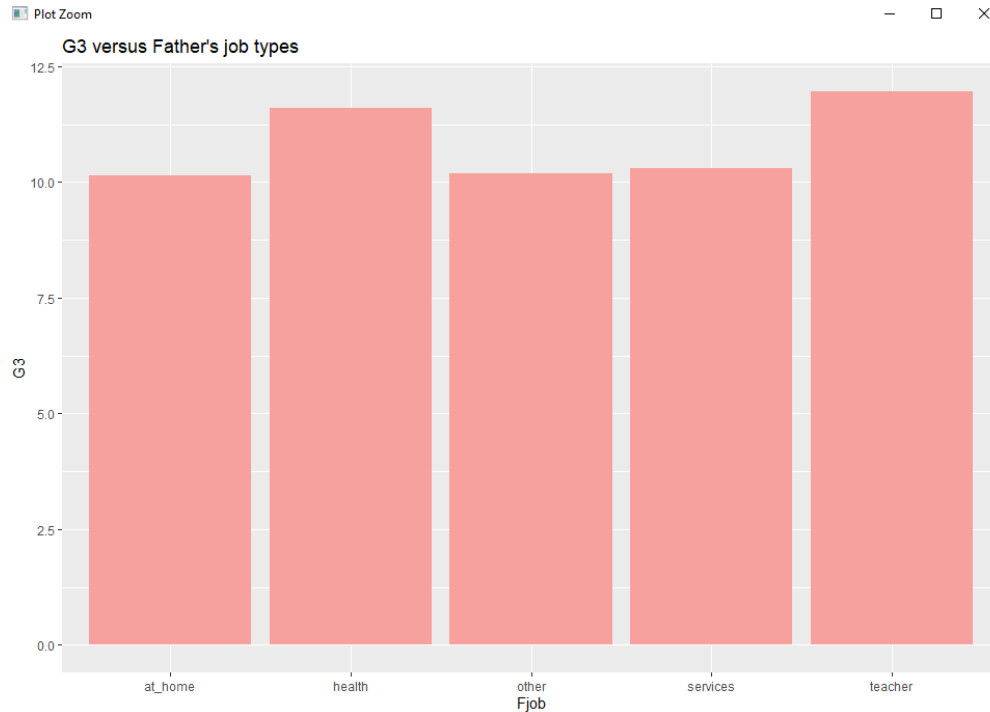
Single students have better scores

G3 versus father job types

```
77 # G3 versus father job type
78 ggplot(dat, aes(x=Fjob, y=G3)) + stat_summary(fun.y="mean", geom="bar", fill="#f7a19e") + ggtitle("G3 versus Father's job types")
79
80 <
76:1 (Top Level) ± R Script
```

Console Terminal

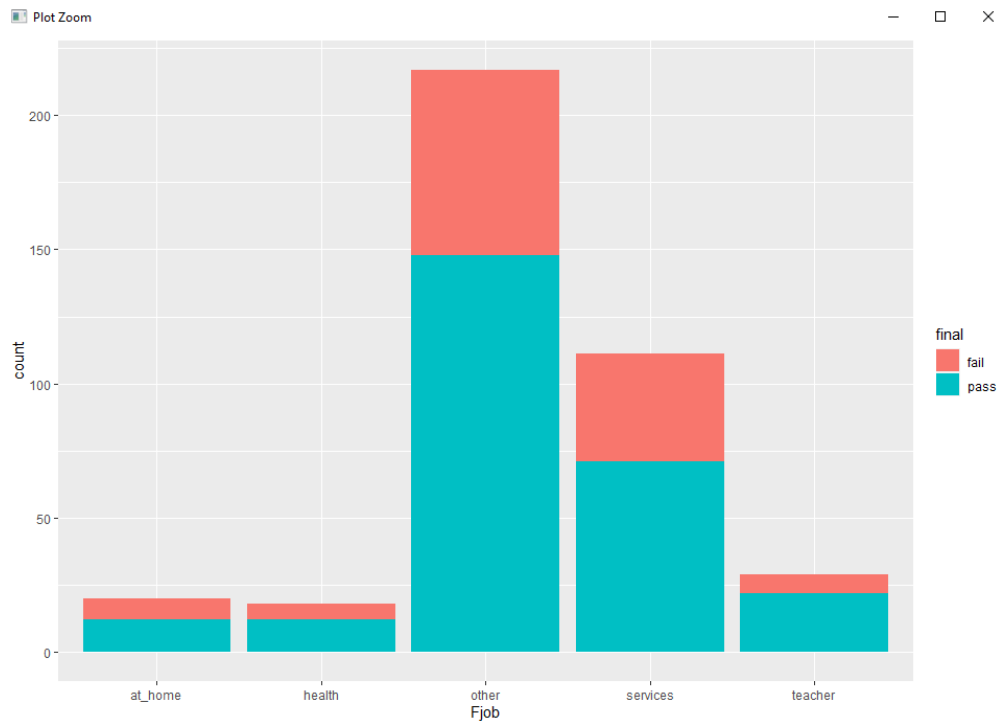
```
~/ |
`fun.y` is deprecated. Use `fun` instead.
> ggplot(dat, aes(x=Fjob, y=G3)) + stat_summary(fun.y="mean", geom="bar", fill="#f7a19e") + ggtitle("G3 versus Father's job types")
warning message:
`fun.y` is deprecated. Use `fun` instead.
> |
```



Students whose father works in education or healthcare field have slightly higher final grade

Bar chart showing count of pass and fail for fathers job types

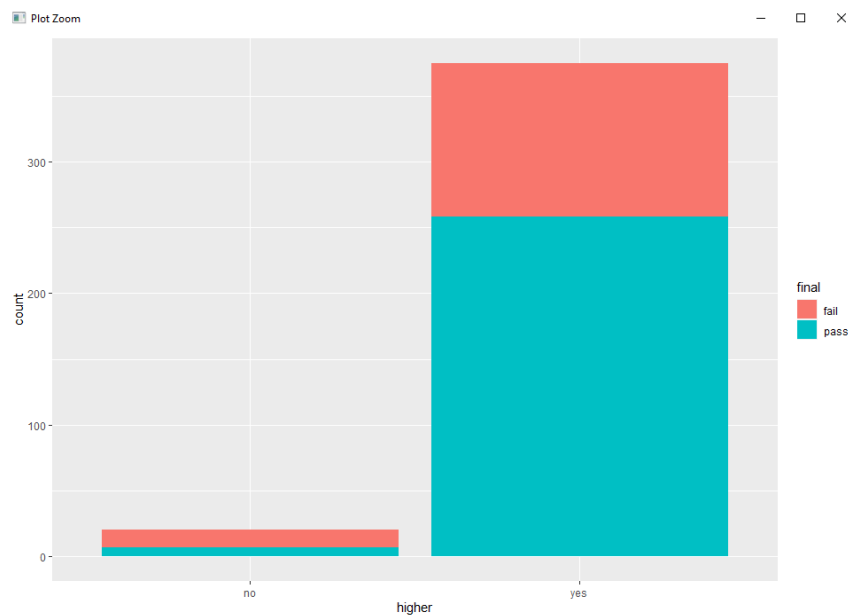
```
77 # convert the target variable (G3) into binary (either pass or fail)
78 # grades less than 10 will be considered as fail and more than or equal to 10 will be considered pass
79 # assign the result into a new variable called final
80 dat$final <- factor(ifelse(dat$G3 >= 10, 1, 0), labels = c("fail", "pass"))
81
82 # remove G3 variable from the dataset
83 dat$G3 <- NULL
84
85 ggplot(dat, aes(x=Fjob, group=final, fill=final)) + geom_bar()
```



The above graph makes it easier to see the relationship between father's occupation and the student's performance

Student's who want to pursue higher education

```
> ggplot(dat, aes(x=higher, group=final, fill=final)) + geom_bar()
> |
```



Students who do not plan to pursue higher education are more likely to fail and vice versa. Yes = 258 pass & 117 fail No = 7 pass & 13 fail

Data preprocessing:

```
97 #data preprocessing
98
99 # create normalize function which takes a vector x of numeric values, and for each value in x subtracts the min value in x and divides
100 # by the range of values in x. A vector will be returned.
101 normalize <- function(x){
102   return ( (x- min(x)) / ( max(x) - min(x)))
103 }
104
105 # normalizing data
106 cols <- c('age','Medu','Fedu','traveltime','studytime','failures','famrel','freetime','goout','Dalc','Walc','health','absences','G1','G2')
107 dat[cols] <- lapply(dat[cols], normalize)
108
109:2 (Top Level) ↕ R Script
```

```
~/
> ggplot(dat, aes(x=higher, group=fina1,fill=fina1)) + geom_bar()
> normalize <- function(x){
+   return ( (x- min(x)) / ( max(x) - min(x)))
+ }
> cols <- c('age','Medu','Fedu','traveltime','studytime','failures','famrel','freetime','goout','Dalc','Walc','health','absences','G1','G2')
> dat[cols] <- lapply(dat[cols], normalize)
> |
```

Training and test data construction:

```
104 #training and test data construction
105 set.seed(123)
106 split <- 0.70
107
108 trainIndex <- createDataPartition(dat$final, p=split, list=FALSE)
109
110 # Create training and test set
111 dat_train <- dat[ trainIndex,]
112 dat_test <- dat[-trainIndex,]
113
114 # Check the dimension of both training and test dataset
115 dim(dat_train)
116
109:1 (Top Level) ↕
```

```
~/
> set.seed(123)
> split <- 0.70
>
> trainIndex <- createDataPartition(dat$final, p=split, list=FALSE)
> dat_train <- dat[ trainIndex,]
> dat_test <- dat[-trainIndex,]
> dim(dat_train)
[1] 277 33
> |

> dim(dat_test)
[1] 118 33
> |
```

```
project_isl.R* x
Source on Save
Run
Source

115 #feature selection step wise regression
116 # base model with intercept only
117 base.mod <- glm(final ~ 1 , dat= dat_train,family=binomial) # base intercept only model
118
119 # full model with all predictor variables
120 all.mod <- glm(final ~ . , dat= dat_train,family=binomial)
121
122 # perform step-wise algorithm
123 stepMod <- step(base.mod, scope = list(lower = base.mod, upper = all.mod), direction = "both", trace = 0, steps = 1000)
127:1 (Top Level)

Console Terminal x
~/
> base.mod <- glm(final ~ 1 , dat= dat_train,family=binomial)
> all.mod <- glm(final ~ . , dat= dat_train,family=binomial)
warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
> stepMod <- step(base.mod, scope = list(lower = base.mod, upper = all.mod), direction = "both", trace = 0, steps = 1000)
There were 50 or more warnings (use warnings() to see the first 50)
> formula(stepMod)
final ~ G2 + Fjob + age + school + walc + famrel + dalc + absences +
health + G1
> stepMod

Call:  glm(formula = final ~ G2 + Fjob + age + school + walc + famrel +
dalc + absences + health + G1, family = binomial, data = dat_train)

Coefficients:
(Intercept)      G2      Fjobhealth      Fjobother      Fjobservices      Fjobteacher      age      schoolMS      walc      famrel
-34.462      56.473      -2.223      3.584      -1.039      8.635     -11.190      4.942      5.444      5.914
dalc      absences      health      G1
-4.830      -7.024      -3.381      12.379

Degrees of Freedom: 276 Total (i.e. Null); 263 Residual
Null Deviance: 350.8
Residual Deviance: 48.52      AIC: 76.52
>

129 #Model building
130 set.seed(1337)
131
132 # Set train_control to 10-fold cross validation
133 train_control <- trainControl(method="cv", number=10)
134
135 # Train the model using glm
136 model <- train(formula(stepMod), dat = dat_train, method = "glm",trControl=train_control,family = binomial)
137
138 # view the summary of the model
139 summary(model)
140
139:1 (Top Level)

Console Terminal x
~/
> set.seed(1337)
> train_control <- trainControl(method="cv", number=10)
> model <- train(formula(stepMod), dat = dat_train, method = "glm",trControl=train_control,family = binomial)
There were 11 warnings (use warnings() to see them)
> summary(model)

Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.66170  -0.00173   0.00008   0.01521   2.03701

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -34.462     7.629   -4.517 6.26e-06 ***
G2             56.473    12.800   4.412 1.02e-05 ***
Fjobhealth    -2.223     2.795   -0.796 0.42630
Fjobother      3.584     1.518    2.361 0.01821 *
Fjobservices  -1.039     1.445   -0.719 0.47184
Fjobteacher     8.635    38.062    0.227 0.82053
age           -11.190     3.515   -3.184 0.00145 **
schoolMS       4.942     1.739    2.842 0.00448 **
```


Model evaluation:

```
140 #prediction
141 glm.probs <- predict(model, newdata = dat_test, type = "raw")
142 #model evaluation
143 confusionMatrix(table(glm.probs, dat_test$final), positive = "pass")
144
143:1 (Top Level) ↕
```

Console Terminal x

~/

```
> glm.probs <- predict(model, newdata = dat_test, type = "raw")
> confusionMatrix(table(glm.probs, dat_test$final), positive = "pass")
Confusion Matrix and Statistics

glm.probs fail pass
fail      34     7
pass       5    72

              Accuracy : 0.8983
              95% CI   : (0.8291, 0.9463)
No Information Rate : 0.6695
P-Value [Acc > NIR] : 6.294e-09

              Kappa : 0.7731

McNemar's Test P-Value : 0.7728

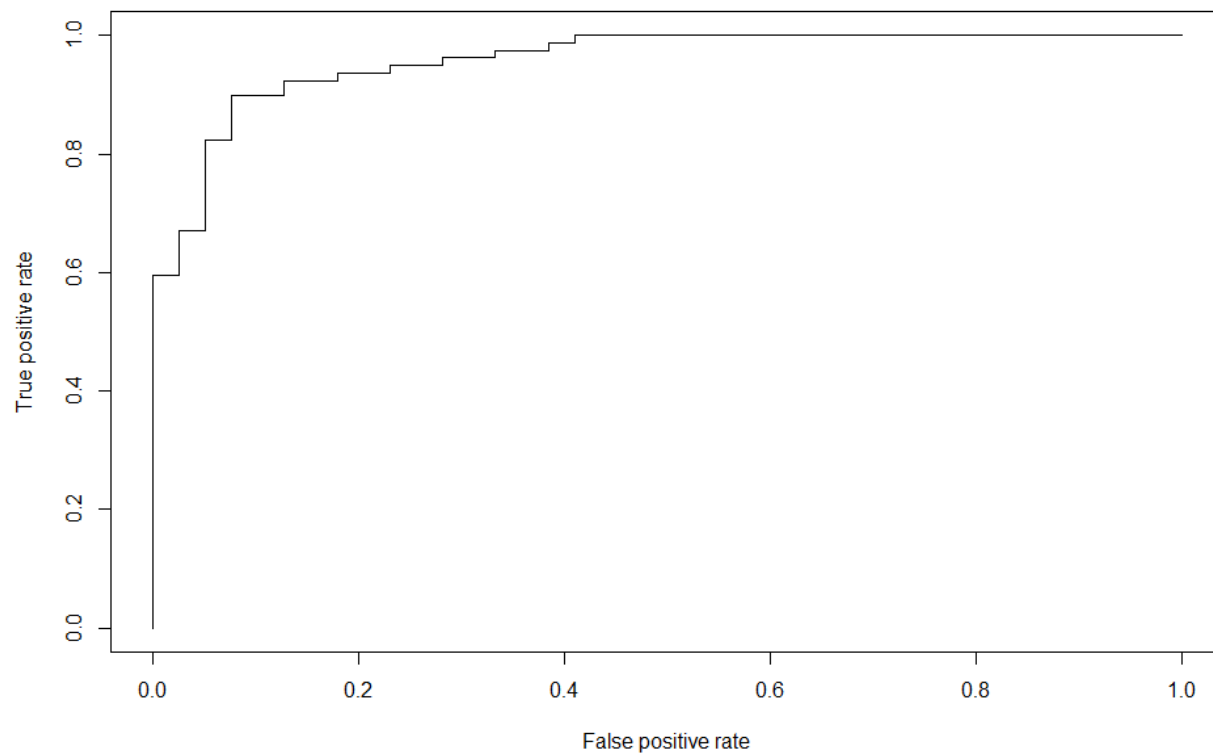
              Sensitivity : 0.9114
              Specificity : 0.8718
              Pos Pred Value : 0.9351
              Neg Pred Value : 0.8293
              Prevalence : 0.6695
              Detection Rate : 0.6102
              Detection Prevalence : 0.6525
              Balanced Accuracy : 0.8916

              'Positive' Class : pass
> |
```

Accuracy of model is 89%

ROC curve:

```
> glm.probs <- predict(model, newdata = dat_test, type="response")
> pr <- prediction(glm.probs, dat_test$final)
> prf <- performance(pr, measure = "tpr", x.measure = "fpr")
> plot(prf)
> |
```



```
> auc <- performance(pr, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.9581305
> |
```

The auc value indicates that the model have 95% chance of accurately discriminate between positive and negative classes.

Logistic Regression:

```
Console Terminal
~/
> logistic.model <- train(data = dat, performance ~ ., method = "glm", trcontrol = trainControl(method = "boot",
+ verboseIter = T,
+ number = 50))
+
+ Resample01: parameter=none
- Resample01: parameter=none
+ Resample02: parameter=none
- Resample02: parameter=none
+ Resample03: parameter=none
- Resample03: parameter=none
+ Resample04: parameter=none
- Resample04: parameter=none
+ Resample05: parameter=none
- Resample05: parameter=none
+ Resample06: parameter=none
- Resample06: parameter=none
+ Resample07: parameter=none
- Resample07: parameter=none
+ Resample08: parameter=none
- Resample08: parameter=none
+ Resample09: parameter=none
- Resample09: parameter=none
+ Resample10: parameter=none
- Resample10: parameter=none
+ Resample11: parameter=none
```

We get 61.7% accuracy rate

Decision Tree:

```
> index <- sample(1:length(dat$performance), size = 0.85 * length(dat$performance))
> training <- dat[index, ]
> test <- dat[-index, ]
> tree.model <- rpart(data = training, formula = performance ~ ., control = rpart.control(cp = 0.015))
> sum(predict(tree.model, test, type = "class") == test$performance) / length(test$performance)
[1] 0.5833333
> |

-
> sapply(seq(3, 30, by = 3), function(k) {
+   sprintf("%.7f", # get output values with 7 decimal places
+     sum(
+       predict(rpart(data = training, formula = performance ~ ., control = rpart.control(cp = 0.3597, minsplit = k)),
+         test, type = "class") == test$performance) / length(test$performance)
+     })
+ })
[1] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[10] "0.5333333"
> |
```

We somehow get upto 60% accuracy rate

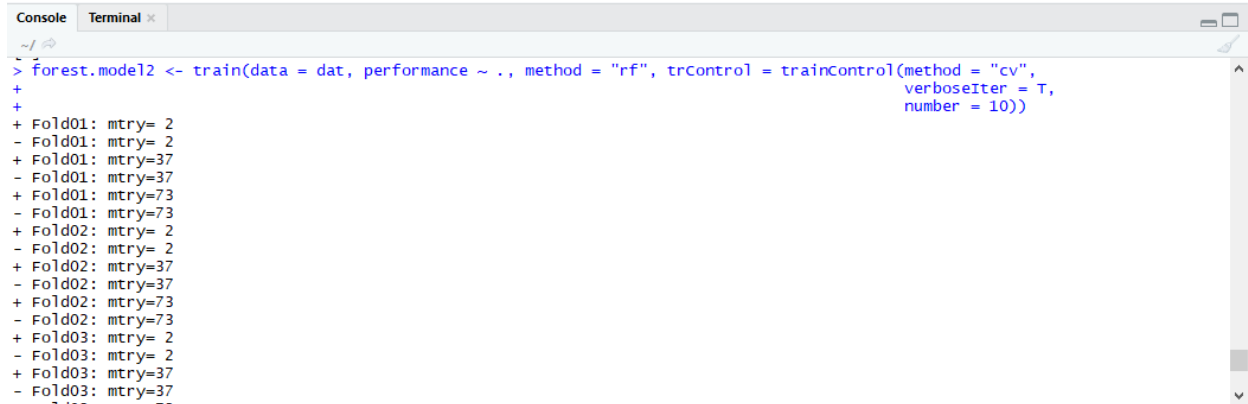
```
Console Terminal
~/
> sapply(1:100, function(k) {
+   sprintf("%.7f",
+     sum(
+       predict(rpart(data = training, formula = performance ~ ., control = rpart.control(cp = 0.3597, minbucket = k)),
+         test, type = "class") == test$performance) / length(test$performance)
+     })
+ })
[1] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[10] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[19] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[28] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[37] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[46] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[55] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[64] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[73] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[82] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[91] "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333" "0.5333333"
[100] "0.5333333"
> |
```

From above none of the values produce a deviation from constant accuracy rate of 60%

Random Forest:

```
> forest.model <- randomForest(data = training, performance ~ ., importance=TRUE, ntree=2000)
> sum(predict(forest.model, test, type = "class") == test$performance) / length(test$performance)
[1] 0.6833333
```

We get 68% accuracy rate which is higher so far



```
Console Terminal
> forest.model2 <- train(data = dat, performance ~ ., method = "rf", trControl = trainControl(method = "cv",
+ verboseIter = T,
+ number = 10))
+
+ Fold01: mtry= 2
- Fold01: mtry= 2
+ Fold01: mtry=37
- Fold01: mtry=37
+ Fold01: mtry=73
- Fold01: mtry=73
+ Fold02: mtry= 2
- Fold02: mtry= 2
+ Fold02: mtry=37
- Fold02: mtry=37
+ Fold02: mtry=73
- Fold02: mtry=73
+ Fold03: mtry= 2
- Fold03: mtry= 2
+ Fold03: mtry=37
- Fold03: mtry=37
```

After doing cross validation accuracy is 64.3% which is smaller value than 68% we got initially.

SVM:

```
> svm.model <- train(data = dat, performance ~ ., method = "svm variant", trControl = trainControl(method = "cv", verboseIter = T, number = 10))
```

It gets 67.3% which is remarkable among all accuracies.

Conclusion:

Out of all models after doing data cleaning, data preprocessing and checking accuracies of all classification algorithms SVM yields better for the data set.