KL7010 PRINCIPLES OF DATA SCIENCE ASSESSMENT

BY
ELIJAH ADEOYE
HENRY

W21051498

Table of Contents

Introduction

Student performance in assessments can be challenging, with several elements contributing to student grades. Predicting student pass rates while also identifying the factors that can influence the scoring of high grades will enhance understanding and development of best practices.

Past Research

Most research conducted has to do with estimating possible features responsible for student grades, one of which is the systematic review conducted by (Romero and Ventura, 2010) to determine factors that affect student performance through information mining procedures. Some of the features identified include a personal and inner appraisal, previous records, extra-curricular activities, and social attributes. Decision trees and neural networks were used since they are the most regularly utilised information-digging strategies (Arun, 2021).

Another research (Alturki and Alturki, 2021) considered how cumulative grade point average could be a metric for evaluating student performances in each semester. Also, the research identified the features of normal class tests and assignments, previous academic failure, and study duration as appreciable factors for predicting student performances. Another study (Francis, 2019) found that family attributes and academic attributes were the deciding features for prediction. The cumulative grade point average of students and their assessment marks were also the most frequently used attributes by the researchers.

Goals and objective

To train a classification model able to predict the value of student passing while also identifying the influential features responsible for student passing or failing an assessment.

Dataset Collection

```
> df1 <- read.csv("student_grades.csv")
```

```
> dim(df1)
[1] 395 31
```

The dim result shows the number of rows which is 395, and the number of columns which is 31.

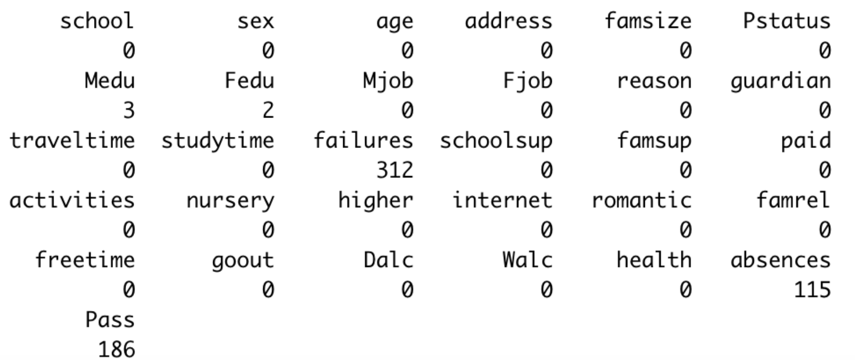
The variable Pass is the response variable and is also used for predicting the value. These classifications will be predicted based on other independent variables.

Checking for any missing data using

```
> sum(is.na(df1))
[1] 0
```

It was noticed that some features (Medu and Fedu) have some zero values, which should not be part of the feature, while the zero in failures, absences and Pass are essential in the data.
as shown in Table 1

```
> colSums(df1 == 0)
```



```
    school        sex        age    address    famsize    Pstatus
         0          0          0          0          0          0
      Medu       Fedu       Mjob       Fjob     reason   guardian
         3          2          0          0          0          0
traveltime  studytime   failures  schoolsup     famsup       paid
         0          0        312          0          0          0
activities    nursery     higher   internet   romantic     famrel
         0          0          0          0          0          0
  freetime      goout       Dalc       Walc     health   absences
         0          0          0          0          0        115
      Pass
       186
```

Table 1

Attribute	Number of Zero Null values
school	0
sex	0
age	0
address	0
Family size	0
Parental status	0
Mother’s education	3
Father’s education	2
Mother’s job	0
Father’s job	0
reason	0
Guardian	0
Travel time	0
Study time	0
failures	312
School support	0
Family support	0
Paid support	0
Extra-curricular activities	0
Nursery	0
Higher education	0
Internet	0
Romantic relationship	0
Family relationship	0
Free time	0
Going out	0
Weekday alcohol consumption	0
Weekend alcohol consumption	0
Health	0
Absences	115
Pass	186

To display the internal structure of a dataset using str (). The output shows a one-liner output for the student grades dataset, providing more information about the dataset and its constituents.

```
> str(df1)
```

The output shows the variables from school, sex, address, famsize, Pstatus, Mjob, Fjob, reason, guardian, schoolsup, famsup, paid, activities, nursery, higher, internet, and romantic, which all have a character data type. In contrast, age, Medu, Fedu, traveltime, studytime, failures, famrel, freetime, gout, Dalc, Walc, health, absences and Pass all have the integer data type and shows a preview of what is in the columns.

To check for data related to the dataset, the summary() function is used, but type conversion should be performed on the character variables.

```
> df1$school <- as.factor(as.character(df1$school))
> df1$address <- as.factor(as.character(df1$address))
> df1$famsize <- as.factor(as.character(df1$famsize))
> df1$Pstatus <- as.factor(as.character(df1$Pstatus))
> df1$Mjob <- as.factor(as.character(df1$Mjob))
> df1$Fjob <- as.factor(as.character(df1$Fjob))
> df1$reason <- as.factor(as.character(df1$reason))
> df1$guardian <- as.factor(as.character(df1$guardian))
> df1$schoolsup <- as.factor(as.character(df1$schoolsup))
> df1$famsup <- as.factor(as.character(df1$famsup))
> df1$paid <- as.factor(as.character(df1$paid))
> df1$activities <- as.factor(as.character(df1$activities))
> df1$nursery <- as.factor(as.character(df1$nursery))
> df1$internet <- as.factor(as.character(df1$internet))
> df1$higher <- as.factor(as.character(df1$higher))
> df1$romantic <- as.factor(as.character(df1$romantic))
```

```
> summary(df1)
```

Table 2 shows the summary for the data, containing the mean, median, first quartile and third quartile for the categorical and numeric variables but contains the data spread for the character variables.

Table 2: Summary

School	349 GP	46 MS
Sex	208 Females	187 Males
Address	88 Rural	307 Urban
Family size	281 greater than 3	114 less than 3

Parental status	41 apart	354 together

Mother's job	59 at home	34 in health	114 other jobs	103 in service	58 in teaching
Father's job	20 at home	18 in health	217 other jobs	111 in service	29 in teaching

Reason	145 course	109 home	36 others	105 reputation

Guardian	90 father	273 mothers	32 others

School support	51 yes	344 no
Family support	242 yes	153 no
Paid	181 yes	214 no
Activities	201 yes	194 no
Nursery	314 yes	81 no
Higher education	375 yes	20 no
Internet	329 yes	66 no
Romantic	132 yes	263 no

Attribute	**Mean**	**Median**	**1st Quartile**	**3rd Quartile**
age	16.7	17.0	16.0	18.0
Medu	2.749	3.000	2.000	4.000
Fedu	2.522	2.000	2.000	3.000
Travel time	1.448	1.000	1.000	2.000
Study time	2.035	2.000	1.000	4.000
failures	0.3342	0.000	0.000	0.000
Famrel	3.944	4.000	4.000	5.000
Free time	3.235	3.000	3.000	4.000
Goout	3.109	3.000	2.000	4.000
Dalc	1.481	1.000	1.000	2.000
Walc	2.291	2.000	1000	3.000
Health	3.554	4.000	1.000	5.000
Absences	5.709	4.000	0.000	8.000
Pass	0.5291	1.0000	0.0000	1.0000

The summary output above shows that the student grade dataset varies with age, where the min age is 15years and Max is 22years. In contrast, Medu and Fedu show a max of 4 and travel time where the min is 1 and a max of 4. Also, study time has a min of 1 and a max of 4. Family relationship (famrel) has a min of 1 and a max of 5. Free time varies from 1 to a max of 5. Going out has a min of 1 with a max of 5. Dac and Walc have a min of 1 with a max of 5.

Data Preprocessing

Some features were identified from the research by (Alturki and Alturki, 2021) and (Francis, 2019), which include past failures, study time, travel time, free time, family relationship, Father and mother's education, outdoor activities such as going out, travel time, weekday and weekend alcohol consumption and in addition age, health and absences were included from domain knowledge. The filter technique of feature selection is used to identify significant components that can accurately predict the outcome based on the relationship with the Pass variable. The following variable was selected; age, Medu, Fedu, traveltime, studytime, failures, famrel, freetime, goout, Dalc, Walc, health and absences.

Exploration

Investigation of the relationship of the features using the multivariate method, where each feature is investigated at a time, and the relationships between the features. Using the graphical approach where boxplot and histogram are used to compare the distribution of features across the student grade dataset.

Data transformation is performed, where the factor types are converted to numeric types for exploration

```
> df1$school <- as.numeric(as.factor(df1$school))

> df1$sex <- as.numeric(as.factor(df1$sex))

> df1$address <- as.numeric(as.factor(df1$address))

> df1$famsize <- as.numeric(as.factor(df1$famsize))

> df1$Pstatus <- as.numeric(as.factor(df1$Pstatus))

> df1$Mjob <- as.numeric(as.factor(df1$Mjob))

> df1$Fjob <- as.numeric(as.factor(df1$Fjob))

> df1$reason <- as.numeric(as.factor(df1$reason))

> df1$guardian <- as.numeric(as.factor(df1$guardian))

> df1$schoolsup <- as.numeric(as.factor(df1$schoolsup))

> df1$famsup <- as.numeric(as.factor(df1$famsup))

> df1$paid <- as.numeric(as.factor(df1$paid))

> df1$activities <- as.numeric(as.factor(df1$activities))

> df1$nursery <- as.numeric(as.factor(df1$nursery))

> df1$higher <- as.numeric(as.factor(df1$higher))

> df1$internet <- as.numeric(as.factor(df1$internet))
> df1$romantic <- as.numeric(as.factor(df1$romantic))
```

The data can now be classed as high-quality data because it is relevant, complete, accurate, consistent and uniform to the performance of student grades.

The dataset df1 is passed to another variable, student_grades, for a clearer presentation

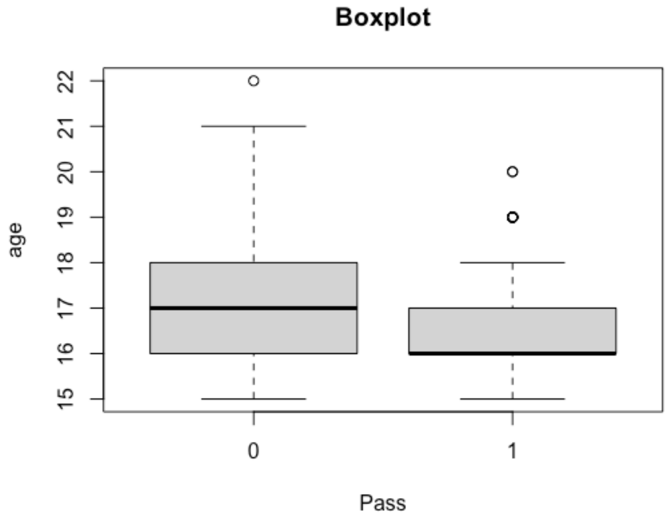
```
student_grades  <- df1
```

We are exploring the relationships between the features and the response variable.

Boxplot

For the relationship between the age and Pass feature

```
> boxplot(student_grades$age ~ student_grades$Pass, xlab = "Pass", ylab = "age", main="Boxplot")
```



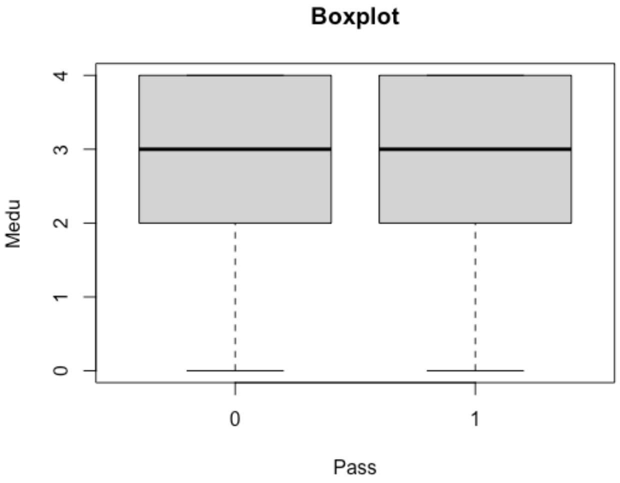
The long upper whisker for no plot means that students' grades vary amongst the positive quartile group, similar to the slightest positive quartile group. While the median for the yes plot, which is closer to the bottom, means that the distribution is positively skewed. There is also the presence of outliers within the data.

The boxplot can be interpreted that most older students of 17 years failed, unlike younger students of 16 years who passed likely due to added responsibility, trying out new habits such as alcohol consumption, going out more and more.

For the boxplot, note that on the x-axis, the value 0= no plot, i.e., students who did not perform well and 1= yes plot, i.e., students who performed well.

The relationship between the mother's education and Pass feature

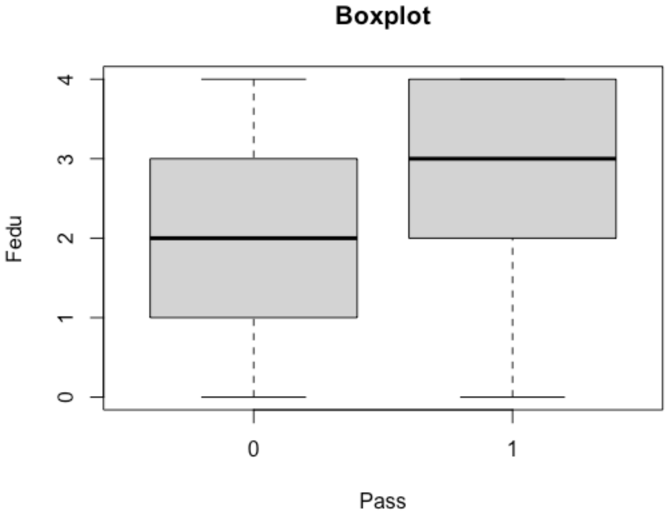
```
> boxplot(student_grades$Medu ~ student_grades$Pass, xlab = "Pass", ylab = "Medu", main="Boxplot")
```



The boxplot shows that for both the no and yes plots, the student's mother's education level has a median of 3, which could mean that the mother has further education. Both plots are pretty similar, suggesting there is no difference between the no and yes plots on Medu.

The relationship between the father's education and Pass feature

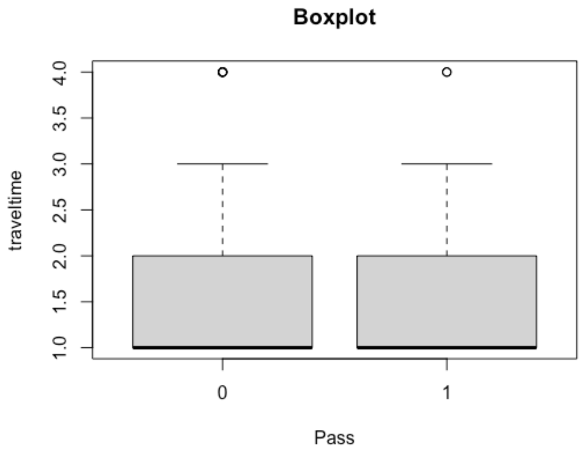
```
> boxplot(student_grades$Fedu ~ student_grades$Pass, xlab = "Pass", ylab = "Fedu", main="Boxplot")
```



The boxplot shows some form of difference. The median for the no plot has a student father education of 2, which could mean that the student father's education has school-level qualifications. The yes plot has a median of 3, which could mean that their father's education level has further education.

For the relationship between the travel time and Pass feature

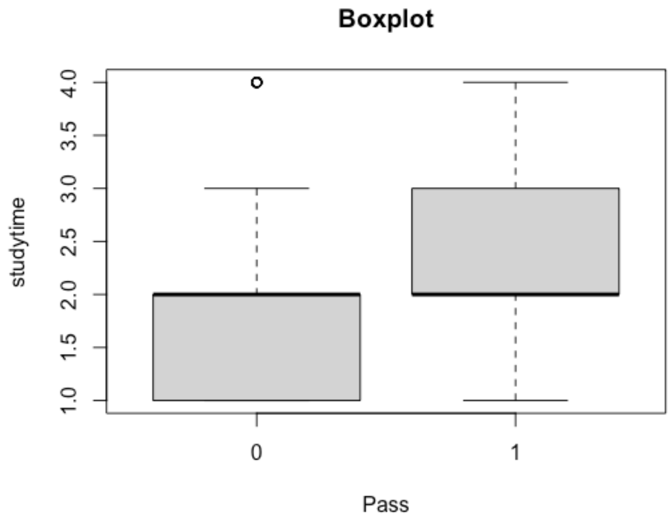
```
> boxplot(student_grades$traveltime ~ student_grades$Pass, xlab = "Pass", ylab = "traveltime", main="Boxplot")
```



Both plots show some similarities with of median of 1, which could mean that the average time to get to school is less than or equal to 15mins. Also, there is the presence of outliers in this relationship.

For the relationship between the study time and the Pass feature

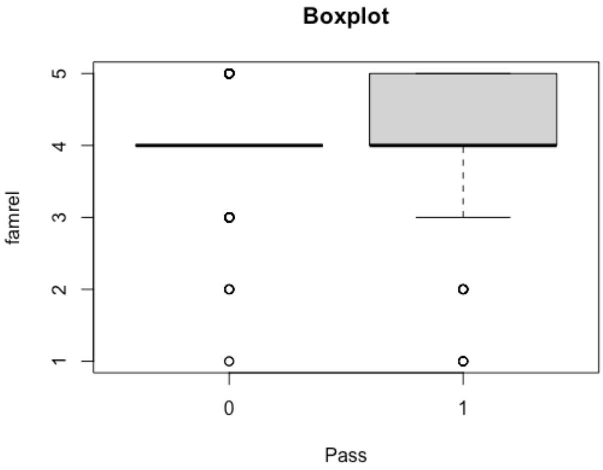
```
> boxplot(student_grades$studytime ~ student_grades$Pass, xlab = "Pass", ylab = "studytime", main="Boxplot")
```



The different plots have the same median of 2, which could mean that they both studied between 2hrs and 5hrs. The yes plot has a greater spread in terms of range and interquartile range than the no plot. Also, there is the presence of outliers in this relationship.

For the relationship between the family relationship and the Pass feature

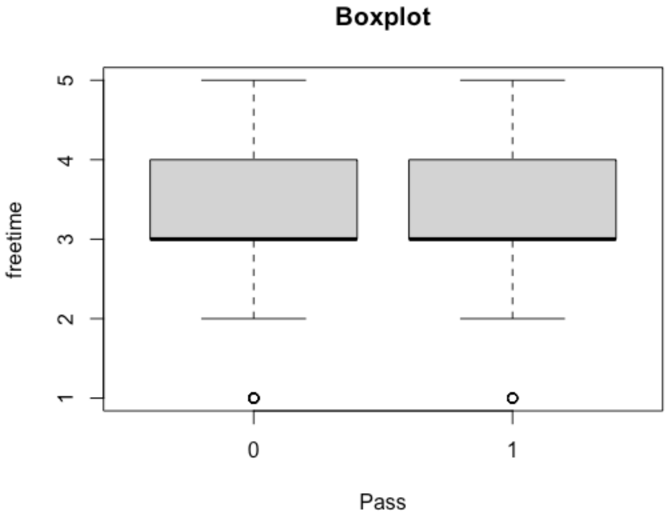
```
> boxplot(student_grades$famrel ~ student_grades$Pass, xlab = "Pass", ylab = "famrel",
main="Boxplot")
```



Both plots have the same median of 4, which infers a not too lousy family relationship. The no plot is a single dash because there is not enough data for that category. Also, there are outliers in this relationship, suggesting more significant variability.

For the relationship between the free time and Pass feature

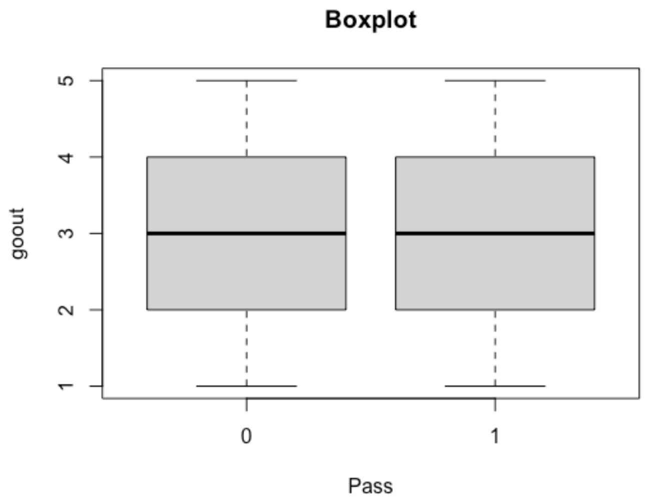
```
> boxplot(student_grades$freetime ~ student_grades$Pass, xlab = "Pass", ylab = "freetime",
main="Boxplot")
```



The plots show similarity with a median of 3, suggesting that they both have average free time. Also, there is the presence of outliers in this relationship.

For the relationship between the going out and pass feature.

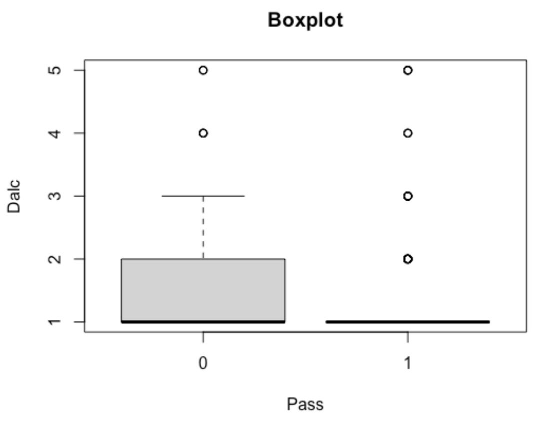
```
> boxplot(student_grades$goout ~ student_grades$Pass, xlab = "Pass", ylab = "goout",
main="Boxplot")
```



The plots show similarity with a median of 3, which could mean a medium level going out with friends where both classes of students had the same go out rate.

For the relationship between the weekday alcohol consumption and pass feature

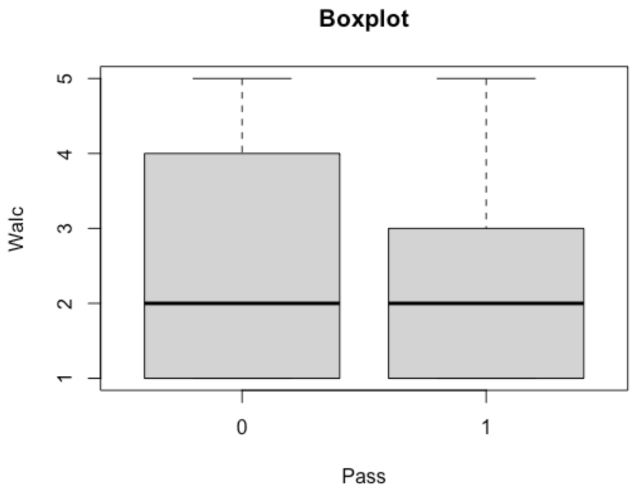
```
> boxplot(student_grades$Dalc ~ student_grades$Pass, xlab = "Pass", ylab = "Dalc",
main="Boxplot")
```



The plots have a similar median of 1, which could mean a low weekday alcohol consumption. It can be interpreted that both categories of student performance have a medium weekday alcohol consumption. The presence of outliers in the yes plot suggests a more significant variability.

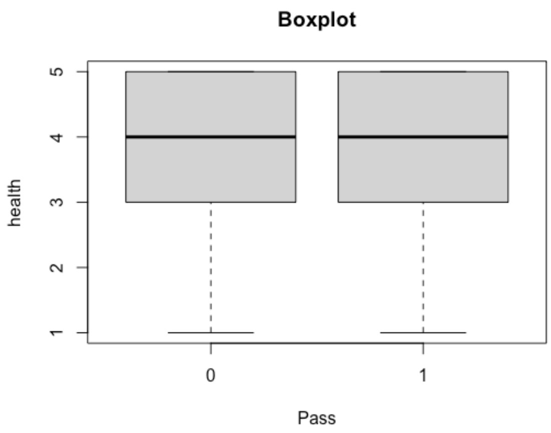
For the relationship between the weekend alcohol consumption and pass feature

```
> boxplot(student_grades$Walc ~ student_grades$Pass, xlab = "Pass", ylab = "Walc",
main="Boxplot")
```



The plots have a similar median of 2, which means a low weekend alcohol consumption. It can be interpreted that both categories of students have a not so low weekend alcohol consumption. For the relationship between the health and pass feature

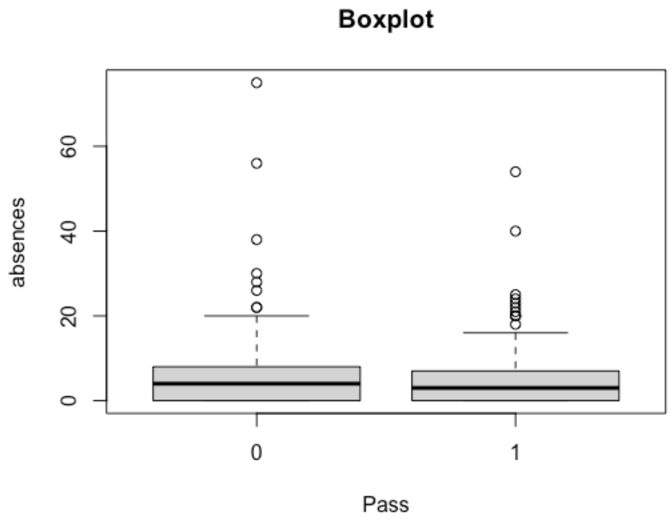
```
> boxplot(student_grades$health ~ student_grades$Pass, xlab = "Pass", ylab = "health",
main="Boxplot")
```



Both plots show similarity and have the same median of 4, which suggest a good health status.

For the relationship between the absences and pass feature

```
> boxplot(student_grades$absences ~ student_grades$Pass, xlab = "Pass", ylab =
"absences", main="Boxplot")
```



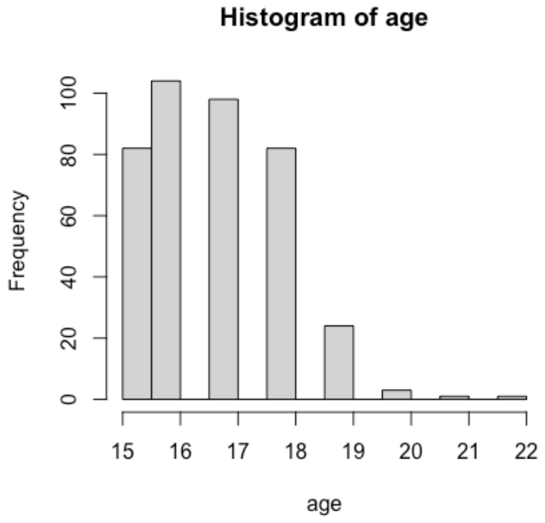
A similar median of 4 suggests a low absences rate, but the presence of outliers suggests more significant variability.

Some features have been identified as having some form of relationship with Pass from the boxplot analysis, but more investigation is required using the bar graph.

Histogram

Histogram of age feature

```
> install.packages("e1071")

> library(e1071)

> hist(student_grades$age, xlab = "age", main = "Histogram of age")
```



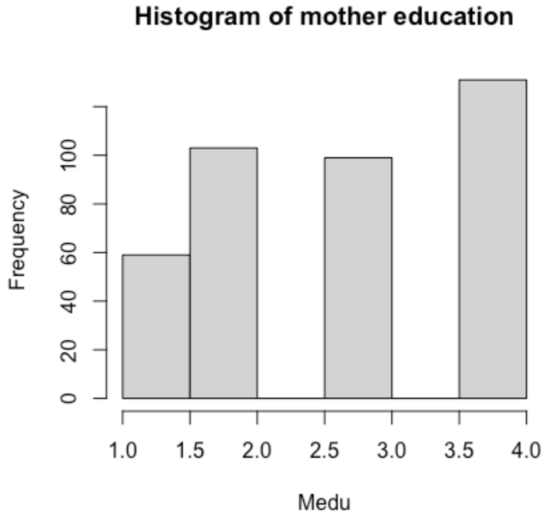
```
> skewness(student_grades$age)

[1] 0.4627348

> kurtosis(student_grades$age)

[1] -0.03144581
```

From the histogram

- Skewness is to the right, which is at 0.46
- Kurtosis is at -0.03, which is a mesokurtic distribution
- The centre is about 18
- The spread of the age from 15 to 22
- There are gaps in the age except for 15 and 16.

Histogram of mother education feature
Since there are missing values in form of zero, the zero value will be ignored for this plot

```
> hist(xlab = "Medu", main = "Histogram of mother education", student_grades$Medu[
!student_grades$Medu==0 ])
```



```
> skewness(student_grades$Medu)

[1] -0.3159667

> kurtosis(student_grades$Medu)

[1] -1.101069
```

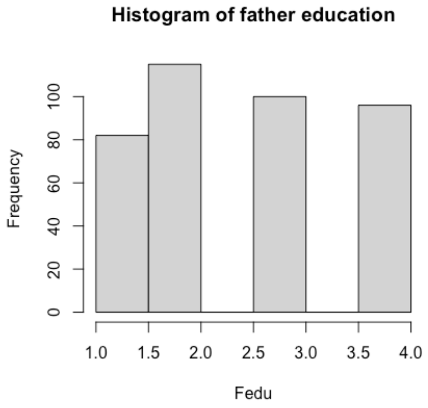
From the histogram

- There are gaps at 2.0 and 3.0, which could mean that there are no values for those points. Using a bar chart could be better for categorical data visualisation.

- Skewness is to the left, which is at -0.31
- Kurtosis is at -1.10, which is a platykurtic distribution
- The centre is about 4
- The spread of the Medu from 0 to 4

Histogram of father education feature

```
> hist(xlab = "Fedu", main = "Histogram of father education", student_grades$Fedu[
!student_grades$Fedu==0 ])
```



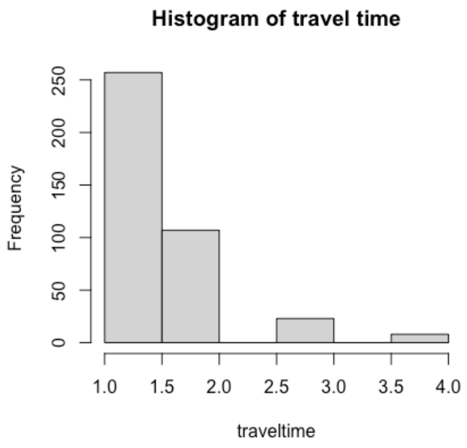
```
> kurtosis(student_grades$Fedu)

[1] -1.207682

> skewness(student_grades$Fedu)

[1] -0.03143195
```

From the histogram

- There are gaps at 2.0 and 3.0, which could mean that there are no values for those points. Using a bar chart could be better for categorical data visualisation.
- Skewness is symmetry, which is at -0.03
- Kurtosis is at -1.21, which is a platykurtic distribution
- The centre is about 2
- The spread of the Fedu from 0 to 4

Histogram of travel time

```
> hist(student_grades$traveltime, xlab = "traveltime", main = "Histogram of travel time")
```



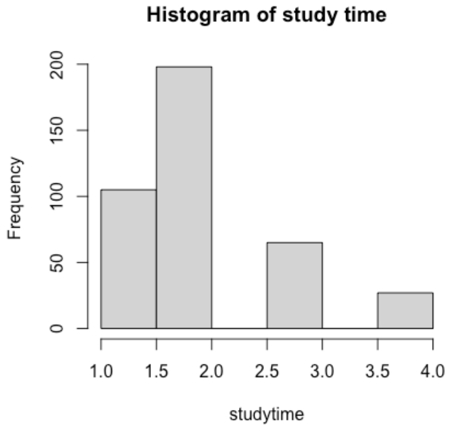
```
> skewness(student_grades$traveltime)

[1] 1.594844

> kurtosis(student_grades$traveltime)

[1] 2.272671
```

From the histogram

- There are gaps because traveltime is categorical data; hence, a bar graph is the best graph to visualise the distribution.
- Skewness is to the right, which is at 1.59
- Kurtosis is at 2.27, which is a leptokurtic distribution
- The centre is about 1
- The spread of the traveltime from 1 to 4

Histogram of study time

```
> hist(student_grades$studytime, xlab = "studytime", main = "Histogram of study time")
```



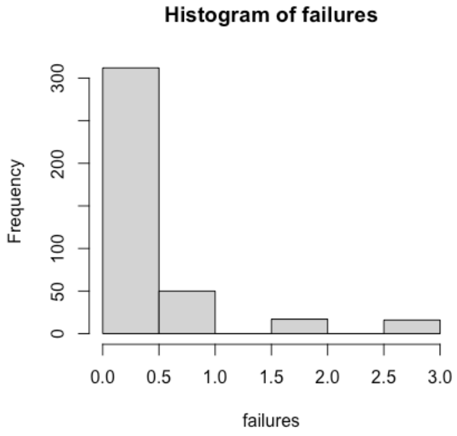
```
> skewness(student_grades$studytime)

[1] 0.6273492

> kurtosis(student_grades$studytime)

[1] -0.04442326
```

From the histogram

- There are gaps because studytime is categorical data; hence the best graph to visualise the distribution is a bar graph.
- Skewness is to the right, which is at 0.63
- Kurtosis is at -0.04, which is a mesokurtic distribution
- The centre is about 1.5
- The spread of the studytime from 1 to 4

Histogram of failures

```
> hist(student_grades$failures, xlab = "failures", main = "Histogram of failures")
```



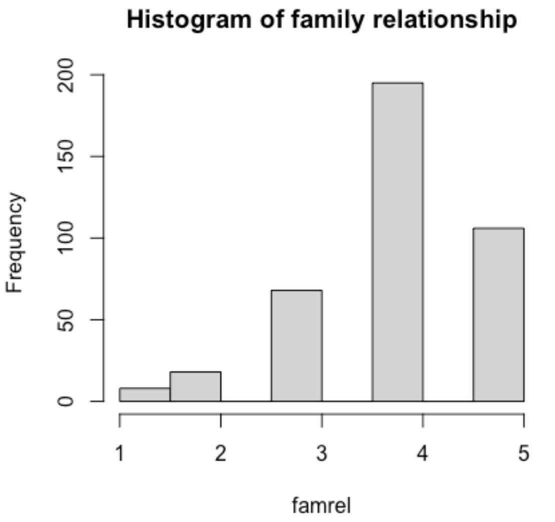
```
> skewness(student_grades$failures)

[1] 2.368927

> kurtosis(student_grades$failures)

[1] 4.886365
```

From the histogram

- Skewness is to the right, which is at 2.37
- Kurtosis is at 4.89, which is a leptokurtic distribution
- The centre is about 1.5
- The spread of the failures from 0.0 to 3.0
- There is a gap in the failures between 1.0 and 2.0, which could mean no data at those points.

Histogram of family relationship

```
> hist(student_grades$famrel, xlab = "famrel", main = "Histogram of family relationship")
```



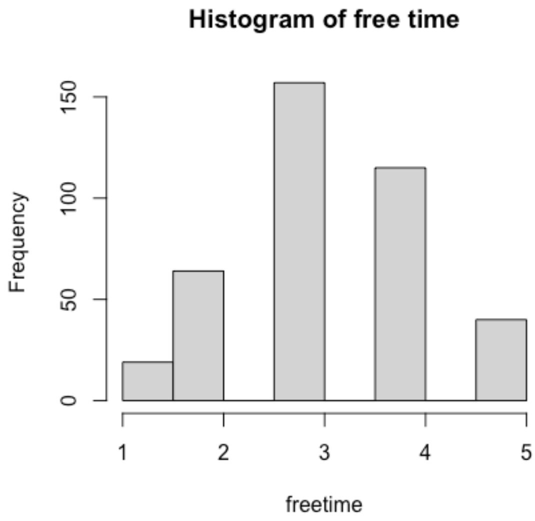
```
> skewness(student_grades$famrel)

[1] -0.9446644

> kurtosis(student_grades$famrel)

[1] 1.089459
```

From the histogram

- There are gaps because famrel is categorical data; hence the best graph to visualise the distribution is a bar graph.
- Skewness is to the left, which is at -0.945
- Kurtosis is at 1.089, which is a leptokurtic distribution
- The centre is about 3
- The spread of the famrel from 1 to 5

Histogram of free time

```
> hist(student_grades$freetime, xlab = "freetime", main = "Histogram of free time")
```



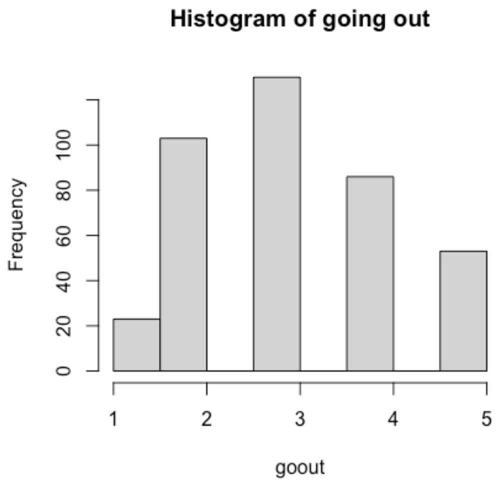
```
> skewness(student_grades$freetime)

[1] -0.1621122

> kurtosis(student_grades$freetime)

[1] -0.3267391
```

From the histogram

- Skewness is symmetric, which is at -0.16
- Kurtosis is at -0.33, which is a mesokurtic distribution
- The centre is about 3
- The spread of the freetime from 1 to 5
- There is a gap in freetime between 2, 3 and 4, which could mean no data at those points.

Histogram of going out

```
> hist(student_grades$goout, xlab = "goout", main = "Histogram of going out")
```



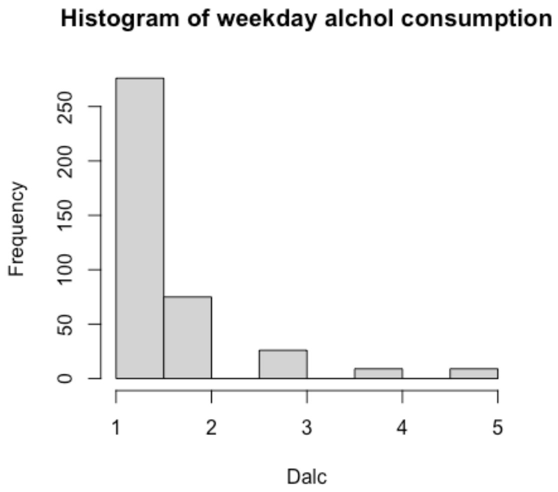
```
> skewness(student_grades$goout)

[1] 0.1156191

> kurtosis(student_grades$goout)

[1] -0.7869343
```

From the histogram

- Skewness is symmetric, which is at 0.116
- Kurtosis is at -0.79, which is a mesokurtic distribution
- The centre is about 3
- The spread of the goout from 1 to 5
- There is a gap in goout between 2, 3 and 4, which could mean no data at those points.

Histogram of weekday alcohol consumption

```
> hist(student_grades$Dalc, xlab = "Dalc", main = "Histogram of weekday alchol
consumption")
```



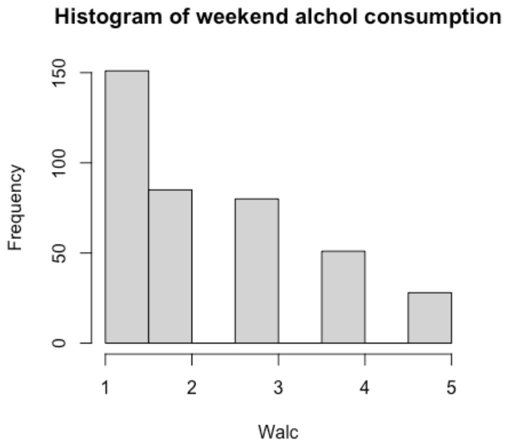
```
> skewness(student_grades$Dalc)

[1] 2.174151

> kurtosis(student_grades$Dalc)

[1] 4.645449
```

From the histogram

- Skewness is to the right, which is at 2.17
- Kurtosis is at 4.64, which is a leptokurtic distribution
- The centre is about 3
- The spread of the Dalc from 1 to 5
- There is a gap in Dalc between 2, 3 and 4, which could mean no data at those points.

Histogram of weekend alcohol consumption

```
> hist(student_grades$Walc, xlab = "Walc", main = "Histogram of weekend alchol
consumption")
```



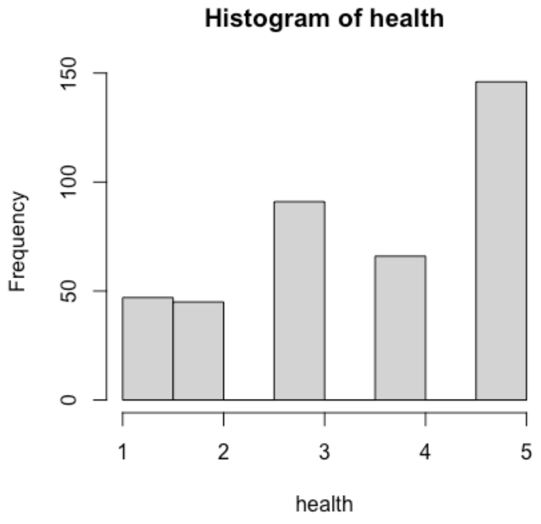
```
> skewness(student_grades$Walc)
[1] 0.60732
> kurtosis(student_grades$Walc)
[1] -0.8071666
```

From the histogram

- Skewness is to the right, which is at 0.61
- Kurtosis is at -0.81, which is a mesokurtic distribution
- The centre is about 3
- The spread of the Walc from 1 to 5
- There is a gap in Walc between 2, 3 and 4, which could mean no data at those points.

Histogram of health

```
> hist(student_grades$health, xlab = "health", main = "Histogram of health")
```



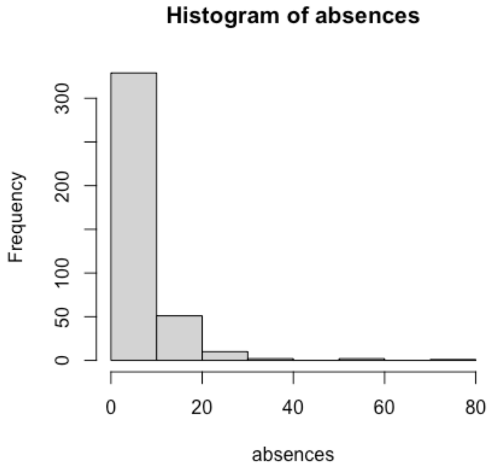
```
> skewness(student_grades$health)

[1] -0.4908534

> kurtosis(student_grades$health)

[1] -1.026469
```

From the histogram

- Skewness is to the left, which is at -0.49
- Kurtosis is at -1.03, which is a platykurtic distribution
- The centre is about 3
- The spread of the health from 1 to 5
- There is a gap in health between 2, 3 and 4, which could mean no data at those points.

Histogram of absences

```
> hist(student_grades$absences, xlab = "absences", main = "Histogram of absences")
```



```
> skewness(student_grades$absences)
[1] 3.643741

> kurtosis(student_grades$absences)

[1] 21.30651
```

From the histogram

- Skewness is to the right, which is at 3.64
- Kurtosis is at 21.3, which is a leptokurtic distribution
- The centre is about 20
- The spread of the absences from 0 to 80
- There are gaps in absences between 40 and 60, which could mean no data at those points.

From the histogram below, positive skewness was noticed within the following variable

failures	Dalc	Walc	age	absences

Negative skewness was noticed within the following variable

freetime	health

The kurtosis was discovered to be a platykurtic distribution for the following

age	Medu	Fedu	studytime	health

Leptokurtic distribution for the following

traveltime	failures	famrel	Dalc	absences

Mesokurtic distribution

studytime	freetime	goout	Walc

It is noticed that histogram gives quality insights for the numeric data, but it is irregular for categorical data hence the need to use a bar chart to explore the categorical features.

Pursuing possible predictors of student grade performance within the set of variables by plotting graphs contrasting performance with some variables.

```
> library(RColorBrewer)
> library(rattle)
> library(car)
```

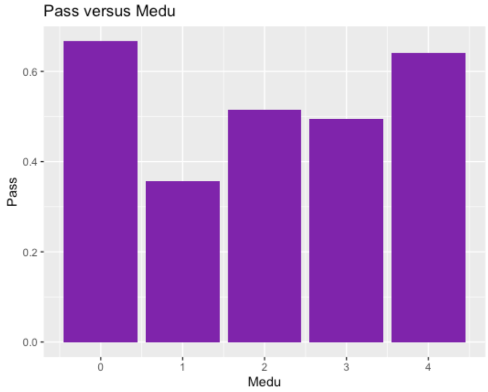
Bar Chart

The ggplot2 package provides more insights while plotting the Pass feature on the individual features.

Pass versus mother education

```
> install.packages("ggplot2")
> library(ggplot2)
```

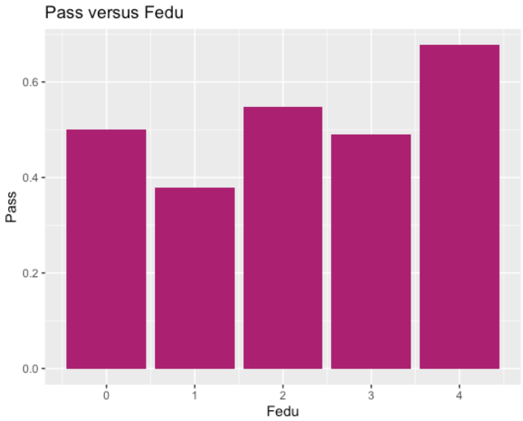
```
> ggplot(student_grades,aes(x=Medu,y=Pass)) + stat_summary(fun.y="mean",
geom="bar",fill="#7F20AB") + ggtitle("Pass versus Medu")
```



From the graph, it is noticed that a large section of the data has 0, which has no bearing on student grades since it represents missing data. Medu is categorical data, where the value of 4 is high, which could mean that most students whose mothers have a higher education performed well. It may be assumed that these mothers spend time teaching their kids hence the better grades.

Pass versus Father education

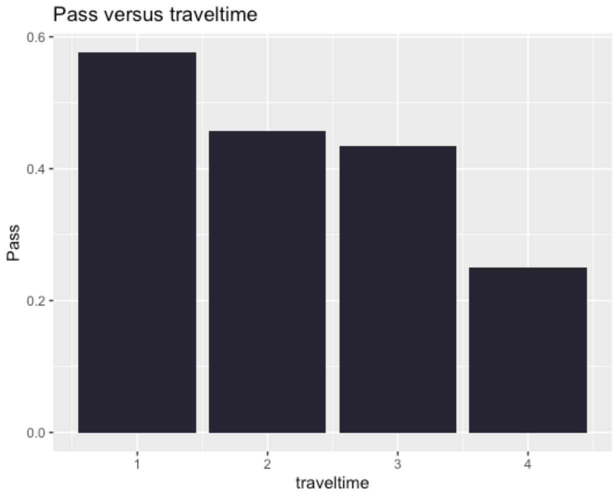
```
> ggplot(student_grades,aes(x=Fedu,y=Pass)) + stat_summary(fun.y="mean", geom="bar",fill="#AB2070") + ggtitle("Pass versus Fedu")
```



From the graph, it is noticed that a large section of the data has 0, which has no bearing on student grades since it represents missing data. Fedu is a categorical data, where the value of 4 is the highest value, which means students whose father has a higher education performed better.

Pass versus travel time

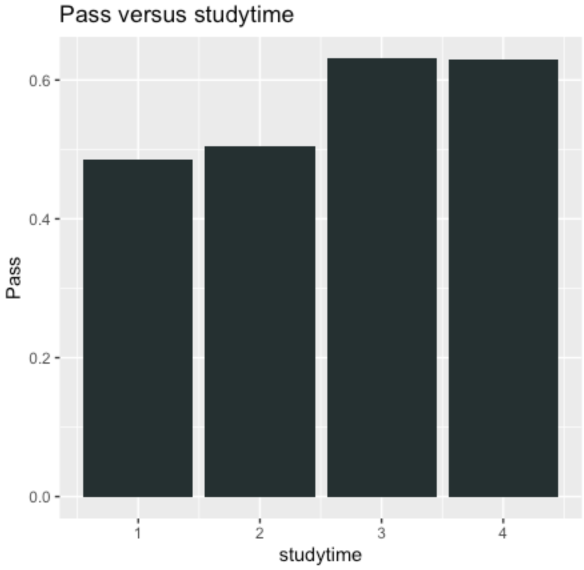
```
> ggplot(student_grades,aes(x=traveltime,y=Pass)) + stat_summary(fun.y="mean", geom="bar",fill="#282531") + ggtitle("Pass versus traveltime")
```



As seen from the graph, traveltime is categorical data, with the 1 being the highest value which could mean that students whose travel time to school is less than 15mins performed better than others.

Pass versus study time

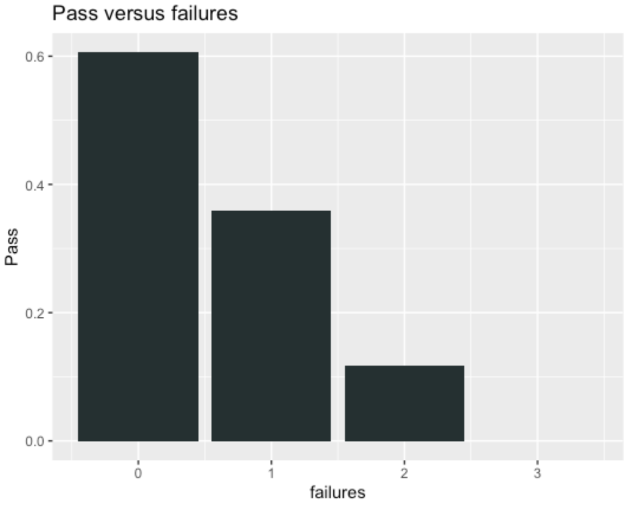
```
> ggplot(student_grades,aes(x=studytime,y=Pass)) + stat_summary(fun.y="mean", geom="bar",fill="#253031") + ggtitle("Pass versus studytime")
```



As seen from the graph, studytime is categorical data, with 3 and 4 being the highest values which could mean that students whose study time is equal to or above 3 hrs and 4hrs performed better than others.

Pass Versus failures

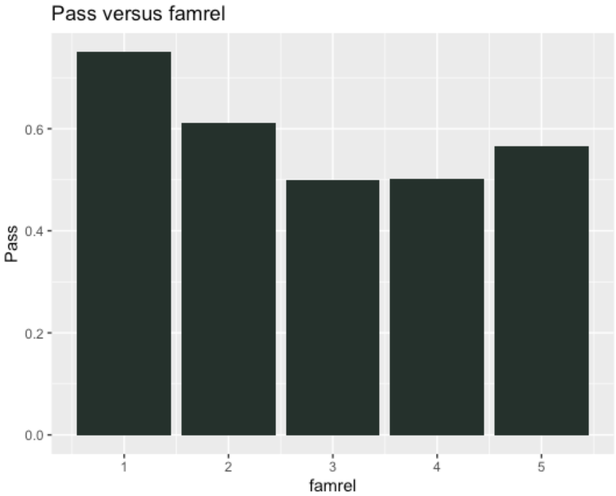
```
> ggplot(student_grades,aes(x=failures,y=Pass)) + stat_summary(fun.y="mean", geom="bar",fill="#253031") + ggtitle("Pass versus failures")
```



As seen from the graph, students with a low failure rate performed better than others with a higher failure rate.

Pass versus family relationship

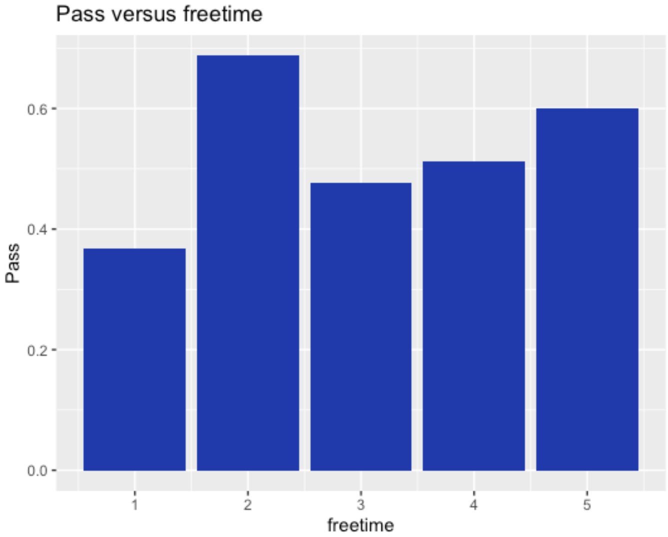
```
> ggplot(student_grades,aes(x=famrel,y=Pass)) + stat_summary(fun.y="mean", geom="bar",fill="#25312C") + ggtitle("Pass versus famrel")
```



As seen from the graph, famrel is a categorical data with the highest value of 1, which could mean that students whose family relationship is poorly performed moderately better than others. It can be assumed that since there was no family relationship to distract the students, there was more time to invest in their studies hence the better grades.

Pass versus free time

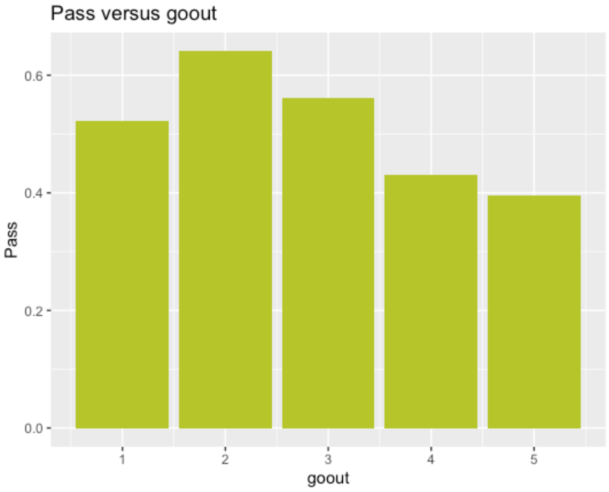
```
> ggplot(student_grades,aes(x=freetime,y=Pass)) + stat_summary(fun.y="mean", geom="bar",fill="#203AAB") + ggtitle("Pass versus freetime")
```



The graph looks irregular, which could mean that students with lower free time performed moderately better.

Pass versus going out

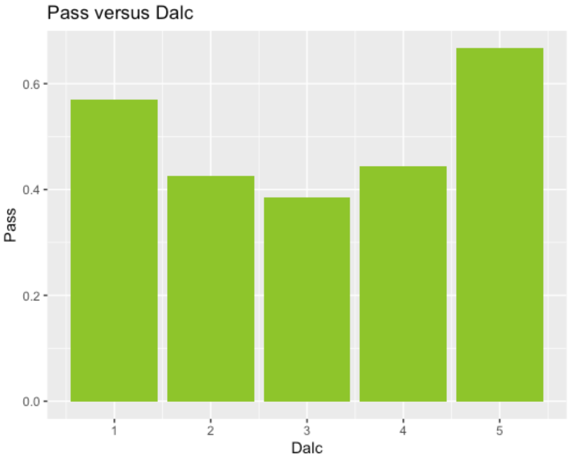
```
> ggplot(student_grades,aes(x=goout,y=Pass)) + stat_summary(fun.y="mean",
geom="bar",fill="#B5C52B") + ggtitle("Pass versus goout")
```



As seen from the graph, students who have a reduced going out time has the highest value of 2 and performed better than students with more going out time. It could mean that the less a student goes out, the better the grades.

Pass versus weekday alcohol consumption

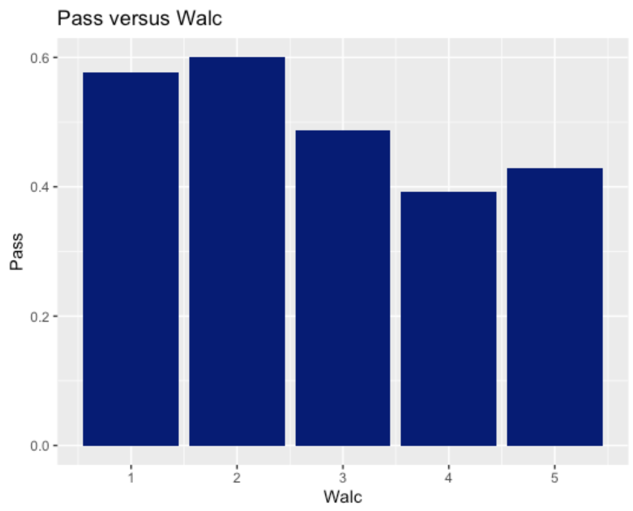
```
> ggplot(student_grades,aes(x=Dalc,y=Pass)) + stat_summary(fun.y="mean",
geom="bar",fill="#8EC52B") + ggtitle("Pass versus Dalc")
```



The graph looks irregular, which shows that students who had a high weekday alcohol consumption of 5 performed better.

Pass versus weekend alcohol consumption

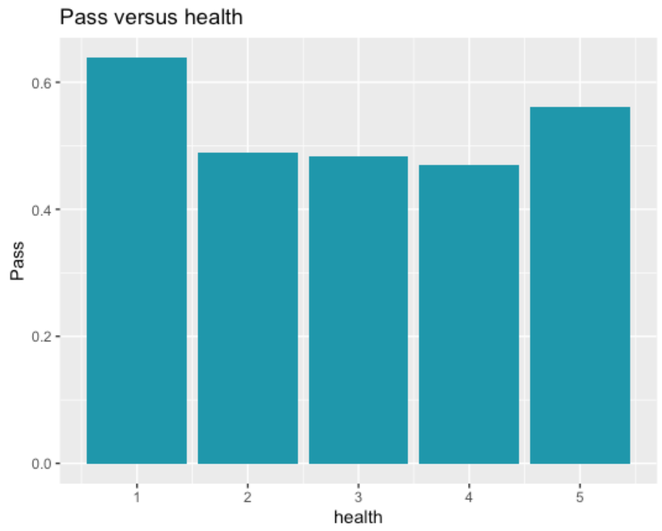
```
> ggplot(student_grades,aes(x=Walc,y=Pass)) + stat_summary(fun.y="mean",
geom="bar",fill="#071C74") + ggtitle("Pass versus Walc")
```



As seen from the graph, students with low weekend alcohol consumption performed better than students with more weekend alcohol consumption. It could mean that the high weekend alcohol consumption negatively imparts student grades.

Pass versus health

```
> ggplot(student_grades,aes(x=health,y=Pass)) + stat_summary(fun.y="mean",
geom="bar",fill="#2097AB") + ggtitle("Pass versus health")
```



The graph shows that the group of students who had bad health performed better than students who had good health. It could mean that students with terrible health had more time to study on the hospital beds or flexible marking scheme hence the better grades.

There is a need to investigate the correlation between different features and the response variable.

Correlation

The correlation Matrix can find the correlation between different paired features in the student grades dataset.

Where the Pearson correlation coefficients and p-value can give more insights

```
> install.packages("corrplot ")
> library(corrplot)

> install.packages("PerformanceAnalytics")
> library(PerformanceAnalytics)
```

```
> cor.test(student_grades$age, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$age and student_grades$Pass
t = -3.0743, df = 393, p-value = 0.002257
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.24816406 -0.05541235
sample estimates:
       cor 
-0.1532455
```

```
> cor.test(student_grades$Medu, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$Medu and student_grades$Pass
t = 3.2986, df = 393, p-value = 0.00106
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.06654023 0.25861630
sample estimates:
      cor 
0.1641336
```

```
> cor.test(student_grades$Fedu, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$Fedu and student_grades$Pass
t = 3.4755, df = 393, p-value = 0.0005668
alternative hypothesis: true correlation is not equal to 0
```

```
95 percent confidence interval:
 0.07529236 0.26680555
sample estimates:
      cor
0.1726805
```

```
> cor.test(student_grades$traveltime, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$traveltime and student_grades$Pass
t = -2.7175, df = 393, p-value = 0.006869
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.23137873 -0.03764164
sample estimates:
       cor
-0.1358083
```

```
> cor.test(student_grades$studytime, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$studytime and student_grades$Pass
t = 2.0005, df = 393, p-value = 0.04613
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.00174863 0.19712053
sample estimates:
      cor
0.1004023
```

```
> cor.test(student_grades$failures, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$failures and student_grades$Pass
t = -6.853, df = 393, p-value = 2.808e-11
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4121019 -0.2356415
sample estimates:
      cor
-0.326716
```

```
> cor.test(student_grades$famrel, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$famrel and student_grades$Pass
t = -0.26494, df = 393, p-value = 0.7912
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1118868  0.0854204
sample estimates:
        cor 
-0.01336326 
```

```
> cor.test(student_grades$freetime, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$freetime and student_grades$Pass
t = -0.020923, df = 393, p-value = 0.9833
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.09971608  0.09762576
sample estimates:
         cor 
-0.001055436 
```

```
> cor.test(student_grades$goout, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$goout and student_grades$Pass
t = -3.089, df = 393, p-value = 0.002151
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.24885317 -0.05614454
sample estimates:
       cor 
-0.1539626 
```

```
> cor.test(student_grades$Dalc, student_grades$Pass)

	Pearson's product-moment correlation

data:  student_grades$Dalc and student_grades$Pass
```

```
t = -1.42, df = 393, p-value = 0.1564
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.16892605  0.02741842
sample estimates:
        cor
-0.07144589
```

```
> cor.test(student_grades$Walc, student_grades$Pass)

            Pearson's product-moment correlation

data:  student_grades$Walc and student_grades$Pass
t = -2.5895, df = 393, p-value = 0.009968
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.22531539 -0.03125234
sample estimates:
      cor
-0.129524
```

```
> cor.test(student_grades$health, student_grades$Pass)

            Pearson's product-moment correlation

data:  student_grades$health and student_grades$Pass
t = -0.20826, df = 393, p-value = 0.8351
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1090629  0.0882576
sample estimates:
        cor
-0.01050491
```

```
> cor.test(student_grades$absences, student_grades$Pass)

            Pearson's product-moment correlation

data:  student_grades$absences and student_grades$Pass
t = -1.1233, df = 393, p-value = 0.262
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1543802  0.0423364
sample estimates:
        cor
-0.05657094
```

Correlation analysis was carried out, and the result is shown below

Features	**Correlation coefficient**
age	-0.1532455
Medu	0.1641336
Fedu	0.1726805
Travel time	-0.1358083
Study time	0.1004023
failures	-0.326716
Famrel	-0.01336326
freetime	-0.001055436
goout	-0.1539626
Dalc	-0.07144589
Walc	-0.129524
health	-0.01050491
absences	-0.05657094

There is a positive relationship between the following features, Medu, Fedu, studytime and Pass. There is a negative relationship between the following features age, traveltime, failures, famrel, freetime, gout, Dalc, Walc, health and absences with Pass.

It was also noticed that the correlation between the variables and response variable is weak since the correlation coefficient is less than 0.25.

In order to search for outliers within the data, clustering analysis will be carried out; it is used to group data into similar groups of students with good or bad grades.

Cluster Analysis

The package needs to be installed on Rstudio

```
> install to perform cluster analysis. packages("ggfortify ")

> library(ggfortify)

> install.packages("stats ")

> library(stats)

> install.packages("dplyr")

> library(dplyr)
```

It is an unsupervised learning method where the index of predictor variables is selected and converted to unlabeled. Creating a copy of the dataset consisting of the categorical and numeric feature without the response variable and naming it “mydata."

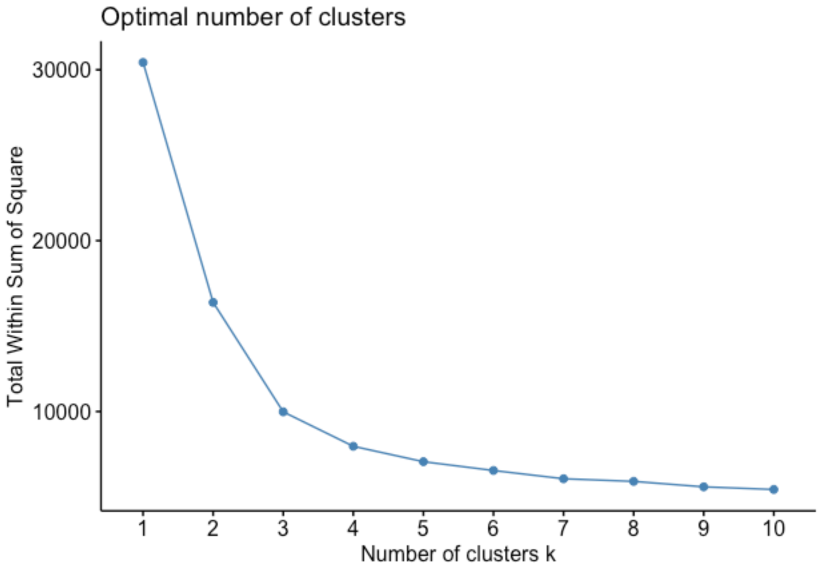
```
> mydata = select(student_grades,c(3,7,8,13,14,15,24,25,26,27,28,29,30))
```

For cluster analysis, the following packages are required

```
> install.packages("factoextra")

> library(factoextra)

> install.packages("NbClust")

> library(NbClust)
```

Establishing the value for k which refers to how many clusters is optimal for the student grades dataset using the elbow method

```
> fviz_nbclust(stud_gradepred, kmeans, method = "wss") + geom_vline(xintercept = 0,
linetype = 2, color = "steelblue")
```



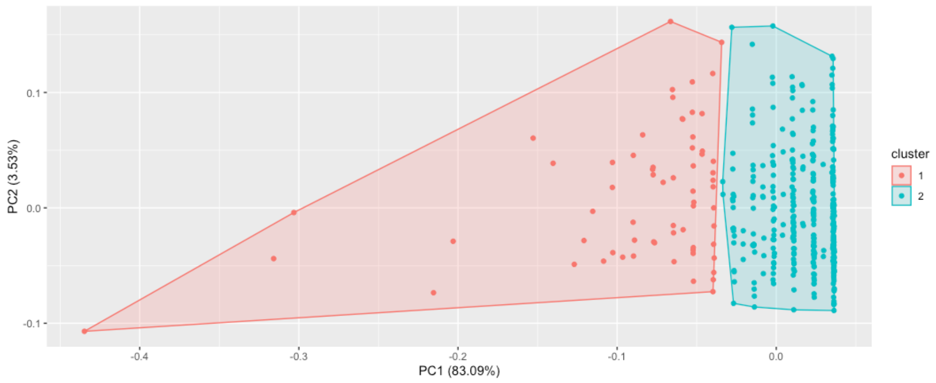
Looking at the point where the line begins to level off, giving the appearance of an elbow shape, i.e. k = 2

Using the kmeans() function to check the number of centroids, which cluster each observation is assigned to.

```
> clusters_k2 <- kmeans(mydata, centers = 2, iter.max = 10)

> clusters_k2$centers

       age     Medu     Fedu   traveltime  studytime    failures
1 17.31250 2.906250 2.562500   1.375000   1.828125   0.5625000
2 16.57704 2.719033 2.513595   1.462236   2.075529   0.2900302

     Famrel    freetime    goout      Dalc      Walc      health     absences
1 3.796875   3.265625   3.312500   1.765625   2.796875   3.468750   19.234375
2 3.972810   3.229607   3.069486   1.425982   2.193353   3.570997    3.093656
```

Evaluating if the clusters are distinct by plotting the cluster, using the function autoplot()

```
> autoplot(clusters_k2,mydata,frame=TRUE)
```

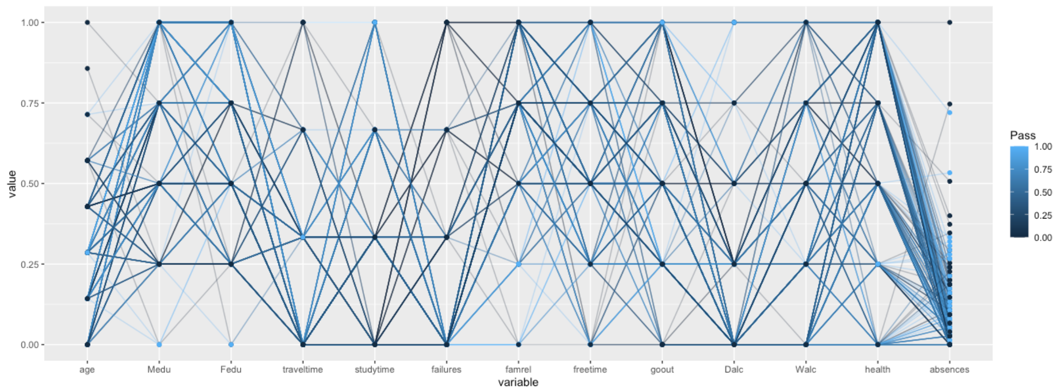


The plot shows the location of the 2 cluster centroids and also shows how each centroid finds the optimum location for class membership identification. Cluster 2 exhibit a lower similarity level, hence the closest points between the observations, while cluster 1 exhibit high similarity, hence the spread of the observations. It means that the cluster analysis has been successfully deployed.

Parallel coordinate plot

The parallel coordinate plot can compare the Pass features and other features. Using the GGally package, creating a copy of the dataset consisting of the categorical and numeric feature and naming it “stud_gradepred."

```
> install.packages("GGally")

> library(GGally)

> stud_gradepred = select(student_grades,c(3, 7, 8, 13, 14, 15, 24, 25, 26, 27, 28, 29, 30, 31))

> ggparcoord(stud_gradepred, columns = 1:13,groupColumn = 14, showPoints = TRUE, alphaLines = 0.3, scale = "uniminmax")
```



Patterns were identified within the student grade dataset, and it shows that there appears to be some variability in the data as it relates to Pass. Students that fail are younger have low study time and absences, they also have a medium Medu, Fedu, freetime, gout, Dalc and a significant travel time, failures, famrel, Walc, health. Students that pass have a low traveltime, failures, famrel, Dalc, health, absences, and they have a medium age, goout and a large Medu, Fedu, studytime, freetime, Walc.

In order to build the model, the dataset has to be split into both training and test sets with the probability of 0.8 for a 1 and 0.2 for a 0. While the model is built on the training set, the test set is to evaluate the model. To ensure that both train and test represent the dataset, the proportion of the data split has to be set.

```
> set.seed(42)

> student_grades[,"train"] <- ifelse(runif(nrow(student_grades))<0.8, 1, 0)
```

Dividing the dataset up into the training and testing sets, based upon the value of "train" for each observation. If the value of "train" is 1, then the observation is moved to the training set, while if the value is 0, then the observation is used for the testing set.

```
> train_data <- student_grades[student_grades$train == "1",]

> test_data <- student_grades[student_grades$train == "0",]

> trainColNum <- grep ("train", names(student_grades))

> train_data <- train_data[-trainColNum]

> test_data <- test_data[-trainColNum]
```

Removing the "train" feature because the randomness inherent in that feature will inhibit the effectiveness of the classification.

```
> dim(train_data)

[1] 316  31

> with(train_data, table(Pass, useNA = "always"))

Pass

   0    1 <NA>

 151  165    0
```

That means about 151 had zero grades, and 165 had a good grade on the train data.

```
> dim(test_data)

[1] 79 31

> with(test_data, table(Pass, useNA = "always"))

Pass

  0    1 <NA>

 35   44    0
```

That means about 35 had zero grades, and 44 had good grades on the test data.

```
Converting the train data to a matrix
> stu_mat <- model.matrix(Pass~. , train_data)

#And the response feature
> stu_res <- ifelse(train_data$Pass=="1", 1, 0)
```

Analysis

Key characteristics shared, such as multivariate and feature importance, are filtered out. Due to the binary nature of the Pass variable, the Logistic regression is used for analysis, while the Xgboost model is used because of the mixture of numerical and categorical features. A regression tree is used when continuous-valued outputs need to be predicted. The prediction of student pass rates using a classification model where there are 2 classes (1= yes and 0= no) denote the outcome of student grade. For the analysis, character variables are ignored; the focus is numerical and categorical variables. Due to the size of the dataset, cross-validation is performed to observe the model efficiency. Splitting the data into a train/test split of 80:20, which handles the class problem.

Fitting using the Logistic Regression Model

Since a binary outcome is expected, where the response variable, which is Pass, has two values: 0 and 1, Logistic regression is best suited. It is used to visualise the relationship between the dependent and one or more independent variables by estimating probabilities. This type of analysis can help to predict the likelihood of an event happening or a choice. Glmnet is used for cross-validation with the code below.

```
model1 = cv.glmnet(x, y, family = "binomial", keep=TRUE)
where x = stu_mat and y = stu_res
```

Using alpha to determine the weight and how valuable the features are

```
> install.packages("glmnet")

> library(glmnet)
> model1 <-cv.glmnet(stu_mat, stu_res, alpha = 1, family = "binomial", type.measure
="mse", alignment = c("lambda"), nfolds = 50, gamma = 1)

> lambda_min <- model1$lambda.min
```

```
> coef(model1, s = lambda_min)
```

A negative value indicates a negative relationship, while a positive value indicates a positive relationship. It can be noticed that Medu, Fedu, studytime, freetime are predictors for student grades.

To predict the predictors, test data needs to be processed as a matrix

```
> stu_mat_test <- model.matrix(Pass~. , test_data)
```

Computing using the general linear model

```
> glm_prob <- predict(model1, newx = stu_mat_test, s = lambda_min, type = "response")
```

Using the discriminant value between 0 to 1

```
> glm_pred <- ifelse(glm_prob>=0.5, 1, 0)
```

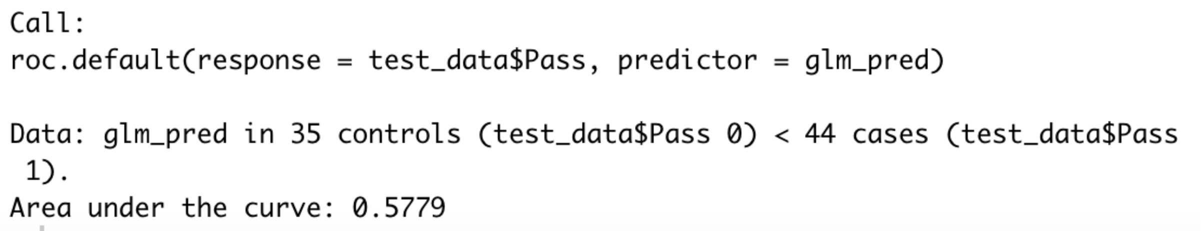
Computing the confusion matrix

```
> glm_confu_mat <- table(pred = glm_pred, actual = test_data$Pass)

> glm_confu_mat

    actual
pred  0  1
   0 15 12
   1 20 32
```

From the result above, for the failed row, 15 were correctly predicted as zero while 12 were incorrectly predicted, while for the passed row, 20 were incorrectly predicted while 32 were correctly predicted.

Checking the auc score

```
> install.packages("pROC")

> library(pROC)

> auc = roc(test_data$Pass, glm_pred)

> print(auc)
```



```
Call:
roc.default(response = test_data$Pass, predictor = glm_pred)

Data: glm_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass
 1).
Area under the curve: 0.5779
```

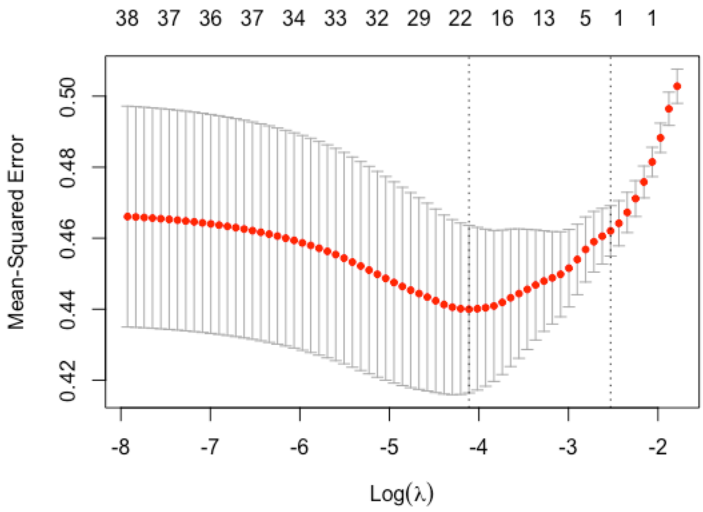
The area under the curve is 0.5779

Computing the mean which is the accuracy

```
> mean(glm_pred == test_data$Pass)

[1] 0.5949367
```

The accuracy is 59%

```
> plot(model1)
```



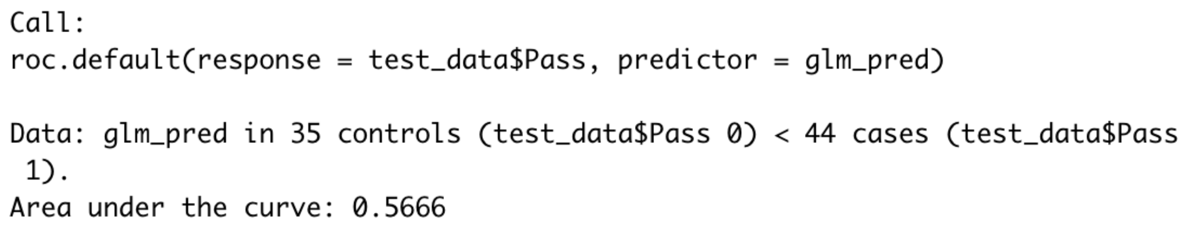
Tuning the hyperparameter to improve accuracy when alpha = 0, which is the ridge regression and type.measure is mean squared error (mse)

```
> model1 <-cv.glmnet(stu_mat, stu_res, alpha = 0, family = "binomial", type.measure
="mse" )
> lambda_min <- model1$lambda.min

> coef(model1, s = lambda_min)
```

```
> glm_prob <- predict(model1, newx = stu_mat_test, s = lambda_min, type = "response")

> glm_pred <- ifelse(glm_prob>=0.5, 1, 0)
> glm_confu_mat <- table(pred = glm_pred, actual = test_data$Pass)
> glm_confu_mat

    actual
pred  0  1
   0 15 13
   1 20 31
```

From the result above, for the failed row, 15 were correctly predicted as zero while 13 were incorrectly predicted, while for the passed row, 20 were incorrectly predicted while 31 were correctly predicted.

```
> auc = roc(test_data$Pass, glm_pred)

> print(auc)
```



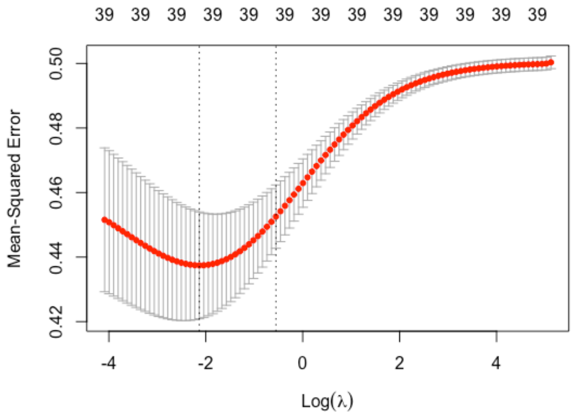
```
Call:
roc.default(response = test_data$Pass, predictor = glm_pred)

Data: glm_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass
 1).
Area under the curve: 0.5666
```

The area under the curve score is 0.5666

```
> mean(glm_pred == test_data$Pass)

[1] 0.5822785
```

Accuracy is 58%

```
> plot(model1)
```



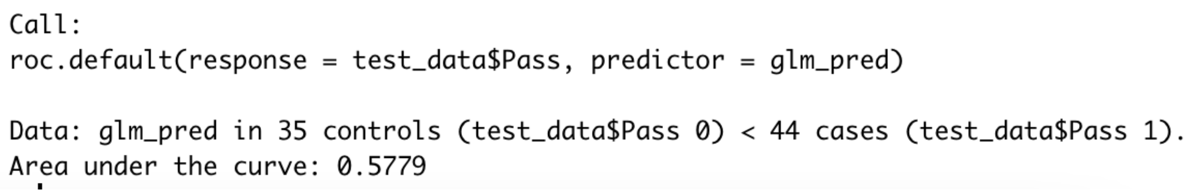
Changing the type.measure to the area under the curve (AUC) and alpha is 1

```
> model1 <-cv.glmnet(stu_mat, stu_res, alpha = 1, family = "binomial", type.measure
="auc")
```

```
> lambda_min <- model1$lambda.min

> stu_mat_test <- model.matrix(Pass~. , test_data)

> glm_prob <- predict(model1, newx = stu_mat_test, s = lambda_min, type = "response")

> glm_pred <- ifelse(glm_prob>=0.5, 1, 0)

> glm_confu_mat <- table(pred = glm_pred, actual = test_data$Pass)

> glm_confu_mat

    actual
pred  0  1
   0 16 12
   1 19 32

> mean(glm_pred == test_data$Pass)

[1] 0.6075949
```

Accuracy is 61%

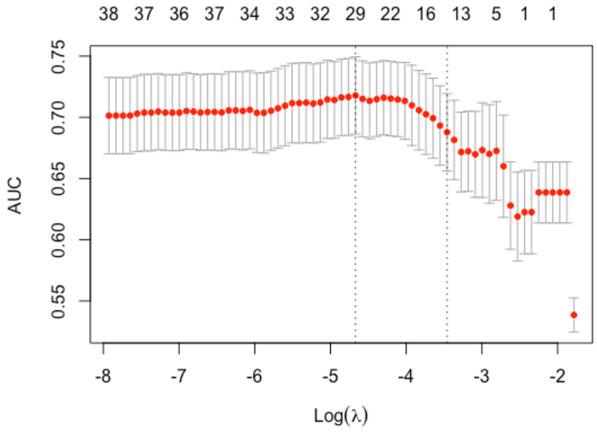
```
> auc = roc(test_data$Pass, glm_pred)

> auc
```



```
Call:
roc.default(response = test_data$Pass, predictor = glm_pred)

Data: glm_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.5779
```

The area under the curve score is 0.5779

```
> plot(model1)
```

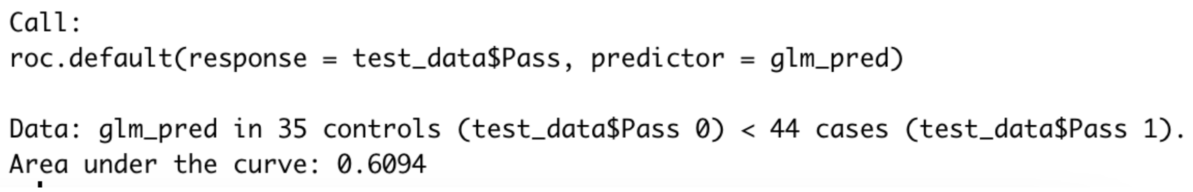


Changing the type.measure to mean absolute error(MAE) and alpha is 1

```
> model1 <-cv.glmnet(stu_mat, stu_res, alpha = 1, family = "binomial", type.measure
="mae")
> lambda_min <- model1$lambda.min

> stu_mat_test <- model.matrix(Pass~. , test_data)

> glm_prob <- predict(model1, newx = stu_mat_test, s = lambda_min, type = "response")

> glm_pred <- ifelse(glm_prob>=0.5, 1, 0)

> glm_confu_mat <- table(pred = glm_pred, actual = test_data$Pass)

> glm_confu_mat

    actual
pred  0  1
   0 18 13
   1 17 31

> mean(glm_pred == test_data$Pass)

[1] 0.6202532
```

Accuracy is 62%

```
> auc = roc(test_data$Pass, glm_pred)
```

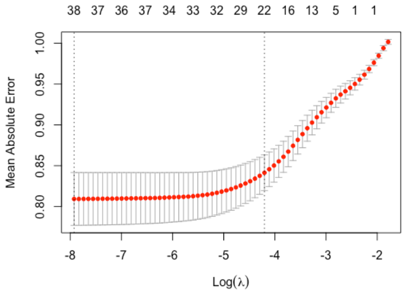


```
Call:
roc.default(response = test_data$Pass, predictor = glm_pred)

Data: glm_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.6094
```

The area under the curve score is 0.6094

```
> rmse <- caret::RMSE(test_data$Pass, glm_pred)

[1] 0.616236
```

```
> plot(model1)
```



Fitting using Xgboost model

It is a simple, easily interpretable and robust classifier. It is an intuitive concept where each "leaf" node represents a feature within the dataset, with each branch being the possible values (or ranges of values) associated with each feature. The Extreme Gradient Boosting model is a scalable, distributed gradient-boosted decision tree. It provides a parallel tree boosting and is the top machine learning library for regression, classification, and ranking problems. It is represented as follows

xgboost(data = NULL, label = NULL, missing = NULL, params = list(), nrounds, verbose = 1, print.every.n = 1L, early.stop.round = NULL, maximize = NULL, ...)

```
> install.packages("xgboost")

> library(xgboost)

> train_label <- train_data$Pass
```

Using the model with the number of epochs of 100, the max depth is 8

```
> model2 <- xgboost(data = stu_mat, label = train_label, eta = 0.2, nrounds = 100, max_depth
= 8, objective = "binary:logistic", verbose = 0)

[17:41:33] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0, the
default evaluation metric used with the objective 'binary:logistic' was changed from 'error'
to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
> xgb_prob <- predict(model2, stu_mat_test, type = "response")

> xgb_pred <- ifelse(xgb_prob>=0.5, 1, 0)

> xgb_cm1 <- table(pred = xgb_pred, actual = test_data$Pass)

> xgb_cm1

   actual
pred  0  1
   0 17 15
   1 18 29
```

From the result above, for the failed row, 17 were correctly predicted as zero while 15 were incorrectly predicted, while for the passed row, 18 were incorrectly predicted while 29 were correctly predicted.

Evaluating the error result of the xgboost model with mse, mae and rmse

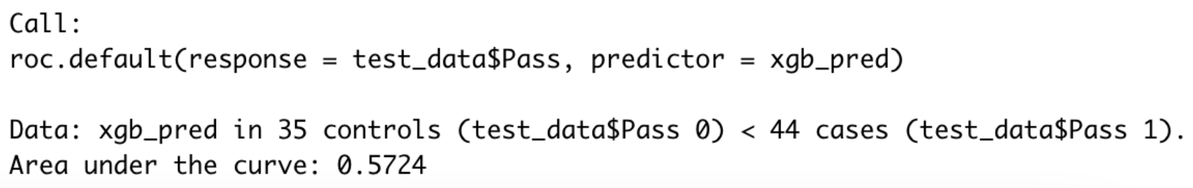
```
> mse <- mean((test_data$Pass - xgb_pred)^2)

[1] 0.4177215
```

```
> mae <- caret::MAE(test_data$Pass, xgb_pred)

[1] 0.4177215

> rmse <- caret::RMSE(test_data$Pass, xgb_pred)

[1] 0.6463138
```

```
> mean(xgb_pred == test_data$Pass)

[1] 0.5822785
```

Accuracy is 58%

```
> auc = roc(test_data$Pass, xgb_pred)
```



```
Call:
roc.default(response = test_data$Pass, predictor = xgb_pred)

Data: xgb_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.5724
```

The area under the curve score is 0.5724

Tuning the hyperparameter where the nrounds is reduced to 50 and max_depth is 4

```
> model2 <- xgboost(data = stu_mat, label = train_label, eta = 0.2, nrounds = 50, max_depth = 4, objective = "binary:logistic", verbose = 0)

[18:05:47] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

> xgb_prob <- predict(model2, stu_mat_test, type = "response")

> xgb_pred <- ifelse(xgb_prob>=0.5, 1, 0)

> xgb_cm1 <- table(pred = xgb_pred, actual = test_data$Pass)

> xgb_cm1
    actual
pred  0  1
   0 19 14
   1 16 30
```

From the result above, for the failed row, 19 were correctly predicted as zero while 14 were incorrectly predicted, while for the passed row, 16 were incorrectly predicted while 30 were correctly predicted.

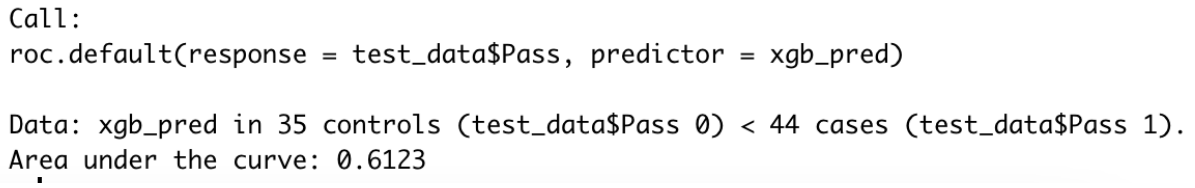
Evaluating the error result of the xgboost model with mse, mae and rmse

```
> mse <- mean((test_data$Pass - xgb_pred)^2)

[1] 0.3797468

> mae <- caret::MAE(test_data$Pass, xgb_pred)

[1] 0.3797468

> rmse <- caret::RMSE(test_data$Pass, xgb_pred)

[1] 0.616236
```

```
> mean(xgb_pred == test_data$Pass)

[1] 0.6202532
```

Accuracy is 62%

```
> auc = roc(test_data$Pass, xgb_pred)
```



```
Call:
roc.default(response = test_data$Pass, predictor = xgb_pred)

Data: xgb_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.6123
```

The area under the curve score is 0.6123

Tuning the hyperparameter where the nrounds is 25 and max_depth is 2

```
> model2 <- xgboost(data = stu_mat, label = train_label, eta = 0.2, nrounds = 25, max_depth
= 2, objective = "binary:logistic", verbose = 0)

[18:08:17] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0, the
default evaluation metric used with the objective 'binary:logistic' was changed from 'error'
to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

> xgb_prob <- predict(model2, stu_mat_test, type = "response")

> xgb_pred <- ifelse(xgb_prob>=0.5, 1, 0)

> xgb_cm1 <- table(pred = xgb_pred, actual = test_data$Pass)

> xgb_cm1

    actual
pred  0  1
```

```
  0 13 10
  1 22 34
```

From the result above, for the failed row, 13 were correctly predicted as zero while 10 were incorrectly predicted, while for the passed row, 22 were incorrectly predicted while 34 were correctly predicted.

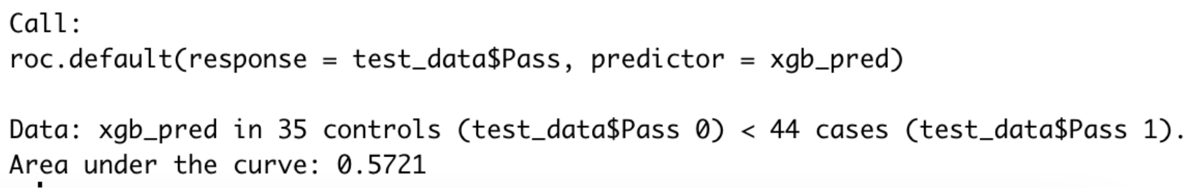
Evaluating the error result of the xgboost model with mse, mae and rmse

```
> mse <- mean((test_data$Pass - xgb_pred)^2)

[1] 0.4050633

> mae <- caret::MAE(test_data$Pass, xgb_pred)

[1] 0.4050633

> rmse <- caret::RMSE(test_data$Pass, xgb_pred)

[1] 0.6364458
```

```
> mean(xgb_pred == test_data$Pass)

[1] 0.5949367
```

Accuracy is 59%

```
> auc = roc(test_data$Pass, xgb_pred)
```



```
Call:
roc.default(response = test_data$Pass, predictor = xgb_pred)

Data: xgb_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.5721
```

The area under the curve score is 0.5721

Tuning the hyperparameter where the nrounds is 10 and max_depth is 4

```
> model2 <- xgboost(data = stu_mat, label = train_label, eta = 0.2, nrounds = 10, max_depth
= 4, objective = "binary:logistic", verbose = 0)

[18:10:13] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0, the
default evaluation metric used with the objective 'binary:logistic' was changed from 'error'
to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

> xgb_prob <- predict(model2, stu_mat_test, type = "response")
> xgb_pred <- ifelse(xgb_prob>=0.5, 1, 0)

> xgb_cm1 <- table(pred = xgb_pred, actual = test_data$Pass)

> xgb_cm1

    actual
pred  0  1
   0 12 13
   1 23 31
```

From the result above, for the failed row, 12 were correctly predicted as zero while 13 were incorrectly predicted, while for the passed row, 23 were incorrectly predicted while 31 were correctly predicted.

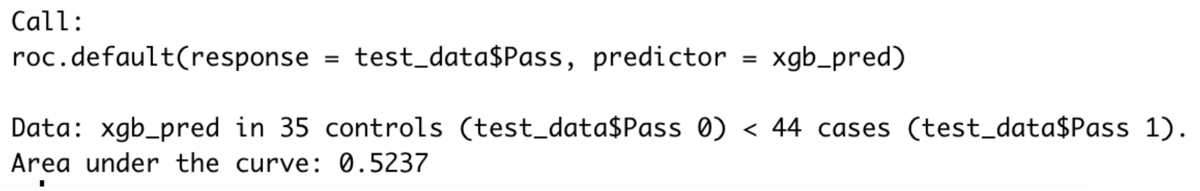
Evaluating the error result of the xgboost model with mse, mae and rmse

```
> mse <- mean((test_data$Pass - xgb_pred)^2)

[1] 0.4556962

> mae <- caret::MAE(test_data$Pass, xgb_pred)

[1] 0.4556962

> rmse <- caret::RMSE(test_data$Pass, xgb_pred)

[1] 0.6750527
```

```
> mean(xgb_pred == test_data$Pass)

[1] 0.5443038
```

Accuracy is 54%

```
> auc = roc(test_data$Pass, xgb_pred)
```



```
Call:
roc.default(response = test_data$Pass, predictor = xgb_pred)

Data: xgb_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.5237
```

The area under the curve score is 0.5237

By tuning the hyperparameters where the nrounds is 100 and max_depth is 4

```
> model2 <- xgboost(data = stu_mat, label = train_label, eta = 0.2, nrounds = 100, max_depth
= 4, objective = "binary:logistic", verbose = 0)
```

```
[18:12:53] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

> xgb_prob <- predict(model2, stu_mat_test, type = "response")

> xgb_pred <- ifelse(xgb_prob>=0.5, 1, 0)

> xgb_cm1 <- table(pred = xgb_pred, actual = test_data$Pass)

> xgb_cm1

    actual
pred  0  1
   0 19 14
   1 16 30
```

From the result above, for the failed row, 19 were correctly predicted as zero while 14 were incorrectly predicted, while for the passed row, 16 were incorrectly predicted while 30 were correctly predicted.

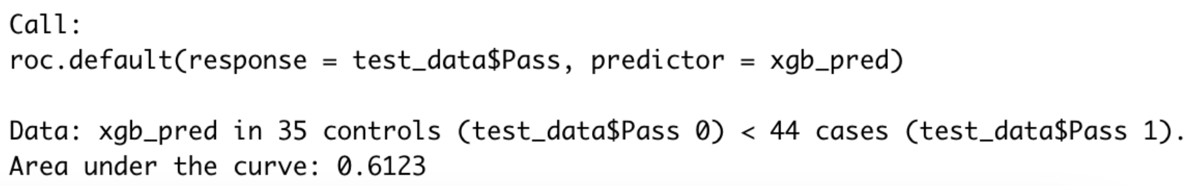
Evaluating the error result of the xgboost model with mse, mae and rmse

```
> mse <- mean((test_data$Pass - xgb_pred)^2)

[1] 0.3797468

> mae <- caret::MAE(test_data$Pass, xgb_pred)

[1] 0.3797468

> rmse <- caret::RMSE(test_data$Pass, xgb_pred)

[1] 0.616236
```

```
> mean(xgb_pred == test_data$Pass)
[1] 0.6202532
```

Accuracy is 62%

```
> auc = roc(test_data$Pass, xgb_pred)
```



```
Call:
roc.default(response = test_data$Pass, predictor = xgb_pred)

Data: xgb_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.6123
```

The area under the curve score is 0.6123

By tuning the hyperparameter where the nrounds is 100 and max_depth is increased to 6

```
> model2 <- xgboost(data = stu_mat, label = train_label, eta = 0.2, nrounds = 100, max_depth = 6, objective = "binary:logistic", verbose = 0)

[18:15:16] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

> xgb_prob <- predict(model2, stu_mat_test, type = "response")

> xgb_pred <- ifelse(xgb_prob>=0.5, 1, 0)

> xgb_cm1 <- table(pred = xgb_pred, actual = test_data$Pass)

> xgb_cm1
    actual
pred  0  1
   0 18 14
   1 17 30
```

From the result above, for the failed row, 18 were correctly predicted as zero while 14 were incorrectly predicted, while for the passed row, 17 were incorrectly predicted while 30 were correctly predicted.

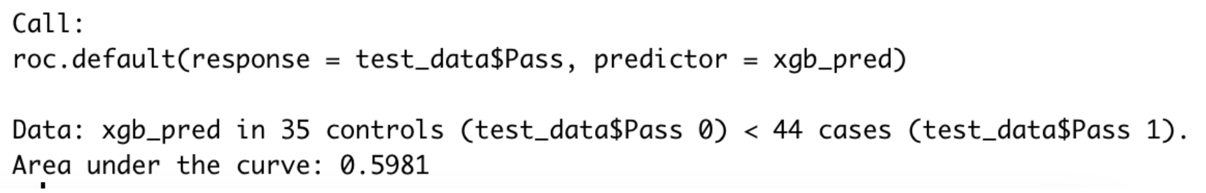
Evaluating the error result of the xgboost model with mse, mae and rmse

```
> mse <- mean((test_data$Pass - xgb_pred)^2)

[1] 0.3924051

> mae <- caret::MAE(test_data$Pass, xgb_pred)

[1] 0.3924051

> rmse <- caret::RMSE(test_data$Pass, xgb_pred)

[1] 0.6264224
```

```
> mean(xgb_pred == test_data$Pass)

[1] 0.6075949
```

Accuracy is 61%

```
> auc = roc(test_data$Pass, xgb_pred)
```



```
Call:
roc.default(response = test_data$Pass, predictor = xgb_pred)

Data: xgb_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.5981
```

The area under the curve score is 0.5981

Fitting using the Regression tree

Since the response variable has continuous values, regression is used and also because of its high accuracy, stability and easy interpretation. It is represented as follows

rpart(formula, data, weights, subset, na.action = na.rpart, method, model = FALSE, x = FALSE, y = TRUE, parms, control, cost, …)

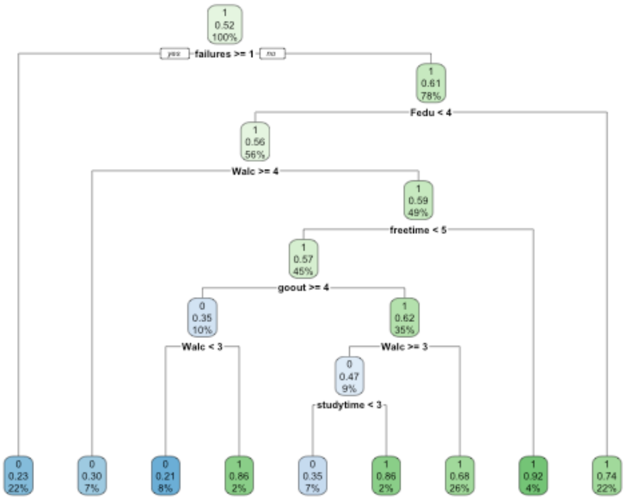
```
> install.packages("fancyRpartPlot")

> library(fancyRpartPlot)

> install.packages("rpart")

> library(rpart)

> install.packages("rpart.plot")

>library(rpart.plot)
```

```
> model3 <- rpart(Pass ~ age + Medu + Fedu + traveltime + studytime + failures + famrel
+ freetime + goout + Dalc + Walc + health + absences, data=train_data, method = 'class')

> summary(model3)
```

Plotting a rpart decision tree using the coloured rpart plotter

```
> rpart.plot(model3, extra = 106)
```

The value 106, which is the "extra", is used to display the binary outcomes associated with each completed branch.



- Observations, where the number of failures is more, are classified as 1 (i.e., will not pass).
- Students with more failures but whose father's education level is less than 4 are also unlikely to pass.
- The student's subset that is most likely to pass are those with less number of failures, a father's education is less than 4, i.e. higher education, have a high weekend alcohol consumption, low free time, low going out with friends rate, and whose weekly study time is less than 3 hours.

This approach has the ability to gather information from the tree that allows for decisions to be made more effectively. For the model to be tested effectively, the response variable must be removed from the testing set.

To generate the predicted values, testing data is applied to the model. This can be done using the predict function, which has three elements: the model to be used in the predictions, the testing data to be used, and the type of prediction to be made.

```
> tree_pred <- predict(model3, newdata = test_data, type = "class")
```

To understand the performance of the model, a confusion matrix can be used; it shows the values predicted by the model in comparison with the actual label values of the training data. There are four possible combinations:

- If both values are 0 (no), then this is a true negative or TN.
- If both values are 1 (yes), then this is a true positive, or TP.
- If the predicted value is 0, but the actual value is 1, this is a false negative or FN.
- If the predicted value is 1, but the actual value is 0, this is a false positive, or FP.

```
> table(predicted = tree_pred, actual = test_data$Pass)
```

```
          actual
predicted  0  1
         0 16 14
         1 19 30
```

The accuracy of a model is the total quantity of TP and TN results over the total number of observations within the testing data.

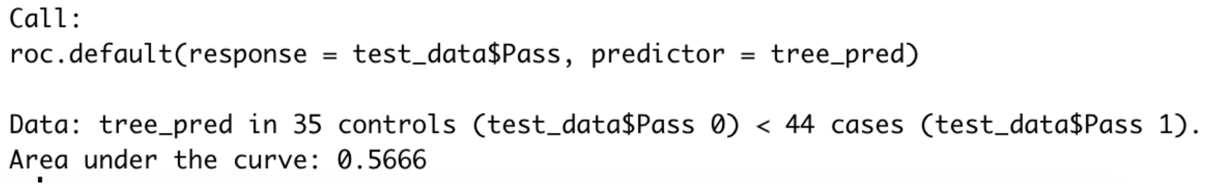
Evaluating the error result with mae and rmse

```
> tree_pred <- as.numeric(as.character(tree_pred))

> mae <- caret::MAE(test_data$Pass, tree_pred)

[1] 0.4177215

> rmse <- caret::RMSE(test_data$Pass, tree_pred)

[1] 0.6463138
```

```
> mean(tree_pred==test_data$Pass)

[1] 0.5822785
```

The model can predict the correct class in 58% of cases.

```
> auc = roc(test_data$Pass, tree_pred)
```

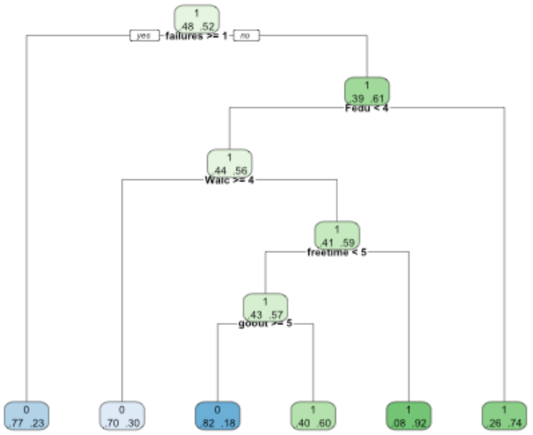


```
Call:
roc.default(response = test_data$Pass, predictor = tree_pred)

Data: tree_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.5666
```

The area under the curve score is 0.5666

Tuning the hyperparameter by including parms, control

```
> model3 <- rpart(Pass ~ age + Medu + Fedu + traveltime + studytime + failures + famrel +
freetime + goout + Dalc + Walc + health + absences, data=train_data, method = 'class', parms
= list(split = "information"), control = rpart.control(minsplit = 5, minbucket = 5, maxdepth
= 10, cp = .02))

> rpart.plot(model3, extra = 4, cex = .5)
```



From the plot, the student's subset that is most likely to pass are those with a smaller number of failures, a father's education is less than 4, i.e., higher education, have a high weekend alcohol consumption, low free time and a low going out with friend’s rate.

```
> tree_pred <- predict(model3, newdata = test_data, type = "class")

> table(predicted = tree_pred, actual = test_data$Pass)

        actual
predicted  0  1
          0 15 13
          1 20 31
```

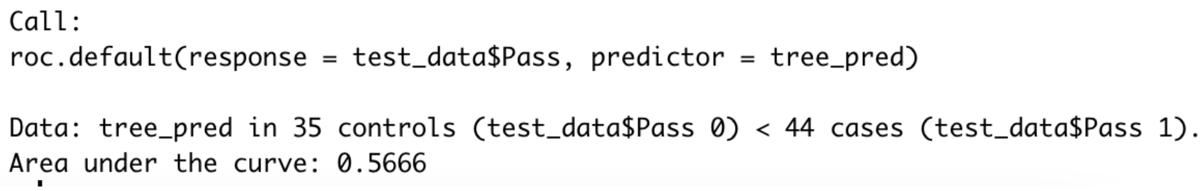
Evaluating the error result with mae and rmse

```
> tree_pred <- as.numeric(as.character(tree_pred))

> mse <- mean((test_data$Pass - tree_pred)^2)

[1] 0.4177215

> mae <- caret::MAE(test_data$Pass, tree_pred)

[1] 0.4177215

> rmse <- caret::RMSE(test_data$Pass, tree_pred)

[1] 0.6463138
```

```
> mean(tree_pred==test_data$Pass)

[1] 0.5822785
```

Accuracy is 58%

```
> auc = roc(test_data$Pass, tree_pred)
```



```
Call:
roc.default(response = test_data$Pass, predictor = tree_pred)

Data: tree_pred in 35 controls (test_data$Pass 0) < 44 cases (test_data$Pass 1).
Area under the curve: 0.5666
```

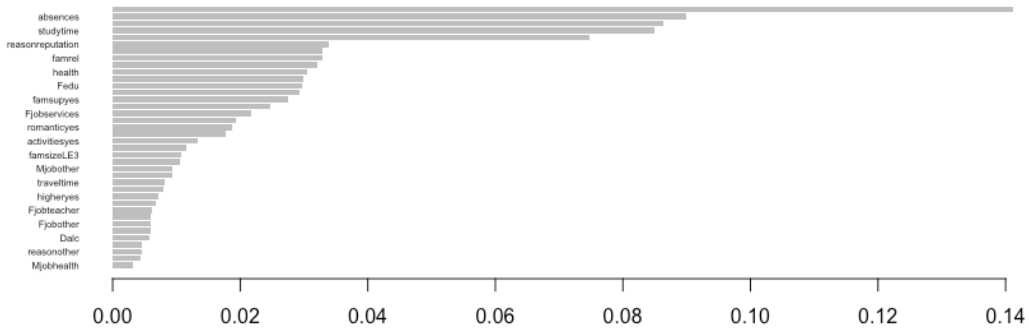
The area under the curve score is 0.5666

Variable Importance

To decide the most important features associated with a student passing or failing the assessment in the models

For Model 2

```
> importance_matrix <- xgb.importance(feature_names = colnames(stu_mat_test), model = model2)

> xgb.plot.importance(importance_matrix)
```



From the plot for Model 2, the following features absences, studytime, famrel, health, and Fedu are the most important features associated with students passing or failing the assessment.

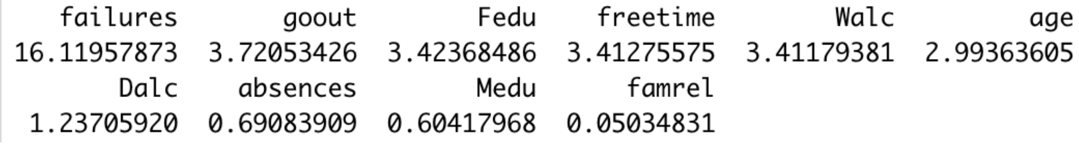
For Model 3

```
> split = summary(model3)$splits
```

```
> split
          count ncat    improve index        adj
failures    316    1 16.1195787   0.5 0.00000000
goout       316    1  6.2618497   3.5 0.00000000
Walc        316    1  5.3522501   3.5 0.00000000
Fedu        316   -1  4.5283518   3.5 0.00000000
studytime   316   -1  4.4339877   2.5 0.00000000
age           0    1  0.8196203  18.5 0.18571429
absences      0    1  0.7879747  14.5 0.04285714
```

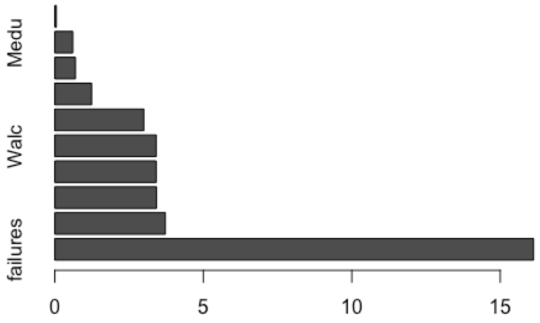
```
Fedu        246   -1  3.4236849   3.5 0.00000000
famrel      246    1  2.5406124   1.5 0.00000000
freetime    246   -1  2.4925230   1.5 0.00000000
age         246    1  1.9222813  17.5 0.00000000
goout       246    1  1.8628403   3.5 0.00000000
Medu          0   -1  0.7723577   3.5 0.17647059
famrel        0    1  0.7276423   1.5 0.01470588
Dalc          0   -1  0.7276423   4.5 0.01470588
Walc        178    1  3.4117938   3.5 0.00000000
freetime    178   -1  3.1415276   4.5 0.00000000
goout       178    1  2.9929868   3.5 0.00000000
age         178    1  2.5398468  17.5 0.00000000
famrel      178    1  1.9821684   2.5 0.00000000
Dalc          0    1  0.9157303   2.5 0.34782609
freetime    155   -1  3.4127557   4.5 0.00000000
age         155    1  2.5783564  17.5 0.00000000
goout       155    1  2.5783564   3.5 0.00000000
absences    155   -1  1.1756938   0.5 0.00000000
studytime   155   -1  1.0843706   2.5 0.00000000
goout       143    1  3.7205343   4.5 0.00000000
freetime    143    1  2.9380125   2.5 0.00000000
age         143    1  2.5993481  17.5 0.00000000
absences    143   -1  1.2786675   0.5 0.00000000
studytime   143   -1  1.1268024   2.5 0.00000000
```

```
> model3$variable.importance
```



```
   failures       goout        Fedu    freetime        Walc         age 
16.11957873  3.72053426  3.42368486  3.41275575  3.41179381  2.99363605 
       Dalc    absences        Medu      famrel 
 1.23705920  0.69083909  0.60417968  0.05034831 
```

```
> barplot(t(model3$variable.importance),horiz=TRUE)
```



From the plot for Model 3, failure by 16%, goout had a 6% gain, and Walc had a 5% gain, which makes them essential features for the decision tree model.

Evaluation

The objective is to build a classification model capable of predicting student grades. The most utilised method for predicting the outcome with respect to the dataset is supervised learning because it gives accurate results. The data were subjected to 3 method, which includes Logistic regression, Xgboost and Regression tree. The performance was determined by preprocessing and splitting into train and test data. From the analysis as shown in table 3, Models 1 and 2 outperformed others with an accuracy of above 60 %. Several features within the student grades dataset provided some interesting insights; the age feature showed that most students above 17 years failed the assessment, unlike their younger peers, possibly due to added responsibilities. Students who performed well-had parents with higher education, which in turn gave them extra tutoring. Furthermore, students who took less than 15 mins to arrive at school performed much better than their counterparts who took a longer time to arrive. It can be inferred that students who have a study time of more than 3 to 10 hours and have not failed in the past performed better. Ironically, students who do not have any family relationships performed better, which could be due to some view that as a distraction hence more time to study. Reduced going out with friends and low Weekend alcohol consumption also contribute to better performance. Also, students with terrible health performed better than others; it can be assumed these students had more study time since they were away from school.

Table 3: Models used and best performing hyperparameter tunning

Model	Model type	Initial result	hyper-parameter tuning
Model 1	Glmnet	59%	62%
Model 2	Xgboost	58%	62%
Model 3	rpart	58%	58%

Model 1 and Model 2 improved their accuracy during hyperparameter tuning, Model 1 improving to 62% from an initial 59%, while Model 2 increased to 62% from 58%. Model 3 did not show any improvement after hyperparameter tuning.

Table 4: Evaluative metrics for the best performing model accuracy

Model	mse	mae	rmse
Model 1	0.5949367	0.6202532	0.616236
Model 2	0.3797468	0.3797468	0.616236
Model 3	0.4177215	0.4177215	0.6463138

When the models were optimised using the evaluative metrics, Model 1 accuracy improved when a mean absolute error (MAE) was included. Model 3 accuracy improved when root mean square error (RMSE) was computed. Based on the result in Table 4, Model 2 performed better than the other models with a lower mean absolute error.

Model 1 and Model 2 were found to be the most effective method, with high accuracy of above 60%, which is substantial for a human-centric problem. When comparing the model performance with various metrics such as accuracy, area under the curve (AUC) and Matthew's correlation coefficient (MCC), Models 1 and 2 had the highest scores.

Table 5: Model comparison using accuracy, area under the curve and Matthew’s correlation coefficient

Model	Accuracy	AUC	MCC
Model 1	62%	0.6094	0.2226

Model 2	62%	0.6123	0.2263
Model 3	58%	0.5666	0.1382

Also, findings identified important features associated with passing or failing in an assessment, as shown in table 6.
Using Matthews correlation coefficient to evaluate the result, It ranges in the interval [−1,+1], with extreme values –1 and +1 reached in case of perfect misclassification and perfect classification, respectively. At the same time, MCC=0 is the expected value for the classifier.

For model 1

```
> install.packages("mccr")

> library(mccr)

> mccr(glm_pred, test_data$Pass)

[1] 0.222622
```

For Model 2

```
> mccr(xgb_pred, test_data$Pass)

[1] 0.2262976
```

For Model 3

```
> mccr(tree_pred, test_data$Pass)

[1] 0.1382386
```

Table 6: Feature importance

Models	Important Features
Model 2	Absences, Study time, family relationship, health and Father's education
Model 3	Past failures, Weekend alcohol consumption and Mother's education

Recently schools have intensified efforts to bolster their rating by trying to improve student performance by predicting grades; understanding student grades in each course is necessary for assisting academically challenged students in surmounting obstacles and helping them excel in the learning process. However, such predictions are difficult to make, especially for new schools, because of not enough data to analyse. Findings have shown that it is possible to predict grades with reasonable accuracy with limited variables.

References

Alturki, S. and Alturki, N. (2021) ‘Using Educational Data Mining to Predict Students’ Academic Performance for Applying Early Interventions’. *J. Inf. Technol. Educ*. Innov. Pract., 20, 121–137

Arun, K. et al. (2021) ‘Student academic performance prediction using educational data mining. *In Proceedings of the 2021’, International Conference on Computer Communication and Informatics (ICCCI),* Coimbatore, India, 27–29; pp. 1–9. 14.

Francis, K. and Babu, S. (2019) ‘Predicting Academic Performance of Students Using a Hybrid Data Mining Approach’. *J. Med. Syst*., 43, 162.

Li, F. et al. (2019) 'Which Factors Have the Greatest Impact on Student’s Performance’. *J. Phys. Conf. Ser.* 2019, 1288, 012077.

Romero, C. and Ventura, S. (2010) ‘Educational Data Mining: A Review of the State of the Art’, *IEEE Trans. Syst. Man Cybern*. Part. C (Appl. Rev.), 40, 601–618.

Siddique, A. et al. (2021) *Predicting Academic Performance Using an Efficient Model Based on Fusion of Classifiers.* Available at: https://doi.org/10.3390/app112411845 (Downloaded: 12 April 2022).

Trautwein, U. et al. (2006) 'Tracking, grading, and student motivation: Using group composition and status to predict self-concept and interest in ninth-grade mathematics’. *J. Educ. Psychol*. 2006, 98, 788–806.