# WIE2003 Group Assignment Report - Predict Your Grade

Group 6

* Lim Wei Yeong (WIE190021 17206011/1)
* Gan Jia Soon (WIE190012 17206343/1)
* Ng Zi Xiang (WIE190035 17205360/1)
* Ng Phoon Ken (WIE190034 17204748/1)

## Introduction

Academic performance of students is always a concern not only to themselves, but also to their parents, teacher and society once they enter working environment. There is a lot of factors that are actually affecting the students in concentrating to their academic subjects, cause a bad result overally. This project is done to investigate on them.

## Problem Statement/ Questions proposed

1. What are the factors that can affect grade of students?
2. How much do these factors affect grade of the students?
3. What is the relationship between these factors?

## Possible stakeholder

Our product is created on the purpose helping predicting the grade of students under conditions and show for various factor affecting academic performance of students. We believe that this product could be useful to users like **teachers**, **parents** and even **students**.

## Data Acquisition

In completing this assignment, R being used. We focus on data extraction, cleaning, analysis and finally visualisation with prediction using Shiny application mainly. We successfully obtain our data set through online where there 2 data sets inside, where one recording for some variables with result of the students for 3 semester in Mathematics, while another with Portuguese. We selected the data set with Mathematics only, where we had a total of **395 rows and 33 columns** at first. Some variables included inside the data set are *age*, *health status*, *family relationship* and more.

## Data Extraction and Cleaning

First, we set the working directory, and load the data, check for dimension, existence of null value and such for the data set chosen. “student-mat.csv” is the raw data set that is unprocessed where we extract it from the source <https://archive.ics.uci.edu/ml/datasets/student+performance>

setwd("D:/WIE2003 Intro to DS/Assignment/Group/student/Grade")  
d1=read.table("student-mat.csv",sep=";",header=TRUE)  
# Display for number of rows and columns  
dim(d1)

## [1] 395 33

# Check for class of the data   
class(d1)

## [1] "data.frame"

# Check for names of columns with its respective type  
str(d1)

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

# Check for existence of null value inside the data  
sum(is.na(d1))

## [1] 0

# Remove duplicate data  
d1 <- unique(d1)  
# Check for summary for whole data  
summary(d1)

## school sex age address   
## Length:395 Length:395 Min. :15.0 Length:395   
## Class :character Class :character 1st Qu.:16.0 Class :character   
## Mode :character Mode :character Median :17.0 Mode :character   
## Mean :16.7   
## 3rd Qu.:18.0   
## Max. :22.0   
## famsize Pstatus Medu Fedu   
## Length:395 Length:395 Min. :0.000 Min. :0.000   
## Class :character Class :character 1st Qu.:2.000 1st Qu.:2.000   
## Mode :character Mode :character Median :3.000 Median :2.000   
## Mean :2.749 Mean :2.522   
## 3rd Qu.:4.000 3rd Qu.:3.000   
## Max. :4.000 Max. :4.000   
## Mjob Fjob reason guardian   
## Length:395 Length:395 Length:395 Length:395   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## traveltime studytime failures schoolsup   
## Min. :1.000 Min. :1.000 Min. :0.0000 Length:395   
## 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:0.0000 Class :character   
## Median :1.000 Median :2.000 Median :0.0000 Mode :character   
## Mean :1.448 Mean :2.035 Mean :0.3342   
## 3rd Qu.:2.000 3rd Qu.:2.000 3rd Qu.:0.0000   
## Max. :4.000 Max. :4.000 Max. :3.0000   
## famsup paid activities nursery   
## Length:395 Length:395 Length:395 Length:395   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## higher internet romantic famrel   
## Length:395 Length:395 Length:395 Min. :1.000   
## Class :character Class :character Class :character 1st Qu.:4.000   
## Mode :character Mode :character Mode :character Median :4.000   
## Mean :3.944   
## 3rd Qu.:5.000   
## Max. :5.000   
## freetime goout Dalc Walc   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:2.000 1st Qu.:1.000 1st Qu.:1.000   
## Median :3.000 Median :3.000 Median :1.000 Median :2.000   
## Mean :3.235 Mean :3.109 Mean :1.481 Mean :2.291   
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:2.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## health absences G1 G2   
## Min. :1.000 Min. : 0.000 Min. : 3.00 Min. : 0.00   
## 1st Qu.:3.000 1st Qu.: 0.000 1st Qu.: 8.00 1st Qu.: 9.00   
## Median :4.000 Median : 4.000 Median :11.00 Median :11.00   
## Mean :3.554 Mean : 5.709 Mean :10.91 Mean :10.71   
## 3rd Qu.:5.000 3rd Qu.: 8.000 3rd Qu.:13.00 3rd Qu.:13.00   
## Max. :5.000 Max. :75.000 Max. :19.00 Max. :19.00   
## G3   
## Min. : 0.00   
## 1st Qu.: 8.00   
## Median :11.00   
## Mean :10.42   
## 3rd Qu.:14.00   
## Max. :20.00

Let us take a look for preview of the data that is not yet processed.

head(d1)

## school sex age address famsize Pstatus Medu Fedu Mjob Fjob reason  
## 1 GP F 18 U GT3 A 4 4 at\_home teacher course  
## 2 GP F 17 U GT3 T 1 1 at\_home other course  
## 3 GP F 15 U LE3 T 1 1 at\_home other other  
## 4 GP F 15 U GT3 T 4 2 health services home  
## 5 GP F 16 U GT3 T 3 3 other other home  
## 6 GP M 16 U LE3 T 4 3 services other reputation  
## guardian traveltime studytime failures schoolsup famsup paid activities  
## 1 mother 2 2 0 yes no no no  
## 2 father 1 2 0 no yes no no  
## 3 mother 1 2 3 yes no yes no  
## 4 mother 1 3 0 no yes yes yes  
## 5 father 1 2 0 no yes yes no  
## 6 mother 1 2 0 no yes yes yes  
## nursery higher internet romantic famrel freetime goout Dalc Walc health  
## 1 yes yes no no 4 3 4 1 1 3  
## 2 no yes yes no 5 3 3 1 1 3  
## 3 yes yes yes no 4 3 2 2 3 3  
## 4 yes yes yes yes 3 2 2 1 1 5  
## 5 yes yes no no 4 3 2 1 2 5  
## 6 yes yes yes no 5 4 2 1 2 5  
## absences G1 G2 G3  
## 1 6 5 6 6  
## 2 4 5 5 6  
## 3 10 7 8 10  
## 4 2 15 14 15  
## 5 4 6 10 10  
## 6 10 15 15 15

Some variables such as *school where the student come from, reason choosing the school* being removed since these are not applicable as these are not suitable factors to be considered in affecting grade since we treat this to be a more universal problem.

Columns like *daily and weekly alcohol consumption* being removed as well since proportion of alcohol consumption among students is not significant enough in some countries.

d1 <- subset(d1, select=-c(Dalc, Walc, school, reason))

Remove the grade for the 3 semester where they would be sum up and processed to form a new column “Average.result”

total <- d1$G1 + d1$G2 + d1$G3  
Result <- total / 3  
d1$Average.result <- Result  
d1 <- subset(d1, select=-c(G1, G2, G3))

Set yes/no to TRUE/FALSE (levels for factor)

d1[d1 == "no"] <- "FALSE"  
d1[d1 == "yes"] <- "TRUE"

Renaming columns to be a more appropriate names

names(d1)[names(d1) == "sex"] <- "Sex"  
names(d1)[names(d1) == "age"] <- "Age"  
names(d1)[names(d1) == "address"] <- "Address"  
names(d1)[names(d1) == "famsize"] <- "Family.size"  
names(d1)[names(d1) == "Pstatus"] <- "Parent.s.cohabition.status"  
names(d1)[names(d1) == "Medu"] <- "Mother.s.education"  
names(d1)[names(d1) == "Fedu"] <- "Father.s.education"  
names(d1)[names(d1) == "Mjob"] <- "Mother.s.job"  
names(d1)[names(d1) == "Fjob"] <- "Father.s.job"  
names(d1)[names(d1) == "guardian"] <- "Guardian"  
names(d1)[names(d1) == "traveltime"] <- "Time.from.home.to.school"  
names(d1)[names(d1) == "studytime"] <- "Weekly.study.time"  
names(d1)[names(d1) == "failures"] <- "Number.of.past.class.failures"  
names(d1)[names(d1) == "schoolsup"] <- "Extra.educational.support"  
names(d1)[names(d1) == "famsup"] <- "Family.educational.support"  
names(d1)[names(d1) == "paid"] <- "Extra.paid.class"  
names(d1)[names(d1) == "activities"] <- "Extra.curricular.activity"  
names(d1)[names(d1) == "nursery"] <- "Attended.nursery.school"  
names(d1)[names(d1) == "higher"] <- "Aims.for.higher.education"  
names(d1)[names(d1) == "internet"] <- "Internet.access"  
names(d1)[names(d1) == "romantic"] <- "In.romantic.relationship"  
names(d1)[names(d1) == "famrel"] <- "Evaluation.for.family.relationship"  
names(d1)[names(d1) == "freetime"] <- "Evaluation.for.free.time.after.school"  
names(d1)[names(d1) == "goout"] <- "Frequent.going.out.with.friends"  
names(d1)[names(d1) == "health"] <- "Current.health.status"  
names(d1)[names(d1) == "absences"] <- "Number.of.school.absences"

Make columns except **Age**, **Number of school absences** and **“Average result”** to be factor

col\_names <- names(d1)  
col\_names <- col\_names [-c(2, 26, 27)]  
d1[col\_names] <- lapply(d1[col\_names], factor)

## Explotary Data Analysis

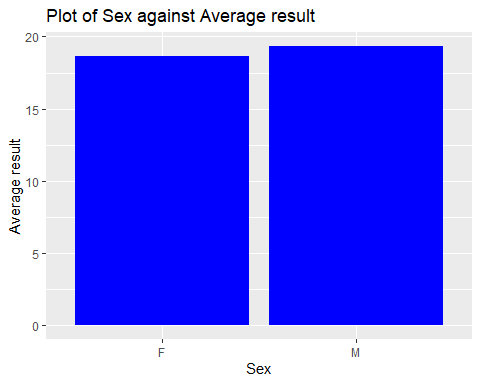
Some graphs would be plotted and then remove for some columns based on reasons. But first, load the suitable library for the visualization

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.5

First graph showing for **Sex** and **“Average result”**

data1 <- data.frame(Sex = d1$Sex, Average.result = d1$Average.result)  
p1 <- ggplot(data1) + geom\_col(aes(x=Sex, y=Average.result), position="dodge", fill ="blue")   
p1 <- p1 + labs(title = "Plot of Sex against Average result", x = "Sex", y = "Average result")  
p1



∴ There is insignificant difference for the two **Sex** attributes with the **“Average result”**, remove **Sex** column is suggested.

To plot following graphs, details below need to be obtained, *count for the factors* of the selected columns need to be known

summary(d1$Guardian)

## father mother other   
## 90 273 32

summary(d1$Mother.s.job)

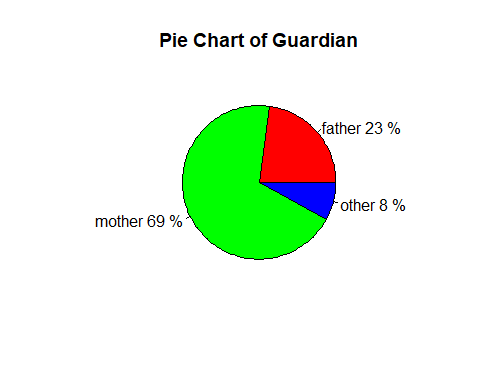
## at\_home health other services teacher   
## 59 34 141 103 58

summary(d1$Father.s.job)

## at\_home health other services teacher   
## 20 18 217 111 29

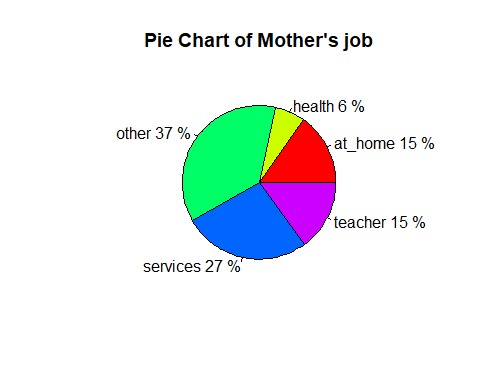
Pie Chart of **Guardian**

countG <- c(90, 273, 32)  
labG <- c("father", "mother", "other")  
percentG <- round(countG / sum(countG)\*100)  
labG <- paste(labG, percentG)  
labG <- paste(labG, "%", sep = " ")  
pG <- pie(countG, labels = labG, col=rainbow(length(labG)), main = "Pie Chart of Guardian")



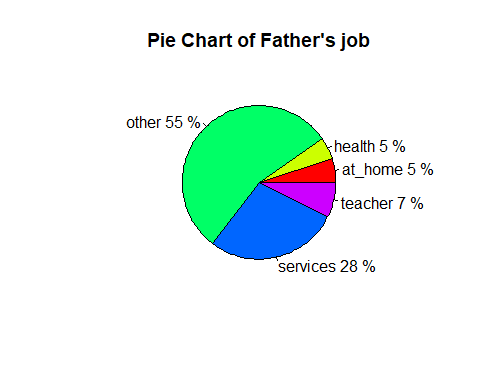
Pie Chart of **Mother’s job**

countM <- c(59, 24, 141, 103, 58)  
labM <- c("at\_home", "health", "other", "services", "teacher")  
percentM <- round(countM / sum(countM)\*100)  
labM <- paste(labM, percentM)  
labM <- paste(labM, "%", sep = " ")  
pM <- pie(countM, labels = labM, col=rainbow(length(labM)), main = "Pie Chart of Mother's job")



Pie Chart of **Father’s job**

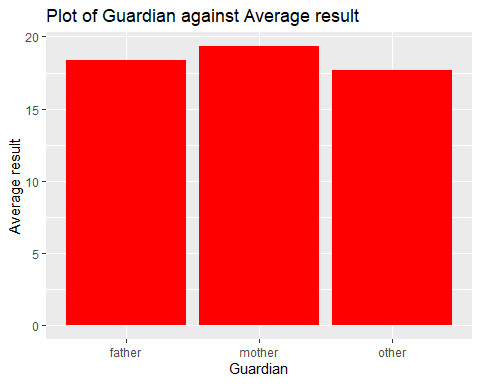
countF <- c(20, 18, 217, 111, 29)  
labF <- c("at\_home", "health", "other", "services", "teacher")  
percentF <- round(countF / sum(countF)\*100)  
labF <- paste(labF, percentF)  
labF <- paste(labF, "%", sep = " ")  
pF <- pie(countF, labels = labF, col=rainbow(length(labF)), main = "Pie Chart of Father's job")



∴ Proportion for “Other and services” are too high for column **Mother’s job** and **Father’s job**, causing the factor to be considered with parent’s job is meaningless. Hence the both columns should be removed.

Make same graph for **Guardian** which is same what did with **Sex** column before

data2 <- data.frame(Guardian = d1$Guardian, Average.result = d1$Average.result)  
p2 <- ggplot(data2) + geom\_col(aes(x=Guardian, y=Average.result), position="dodge", fill = "red")   
p2 <- p2 + labs(title="Plot of Guardian against Average result", x="Guardian", y="Average result")  
p2



∴ There is insignificant difference for the three **Guardian** attributes with the **“Average result”**, removing **Guardian** column is required

Removing columns **Sex**, **Guardian**, **Mother’s job**, and **Father’s job**

d1 <- d1[-c(1, 8, 9, 10)]

Final checking on data processed

summary(d1)

## Age Address Family.size Parent.s.cohabition.status  
## Min. :15.0 R: 88 GT3:281 A: 41   
## 1st Qu.:16.0 U:307 LE3:114 T:354   
## Median :17.0   
## Mean :16.7   
## 3rd Qu.:18.0   
## Max. :22.0   
## Mother.s.education Father.s.education Time.from.home.to.school  
## 0: 3 0: 2 1:257   
## 1: 59 1: 82 2:107   
## 2:103 2:115 3: 23   
## 3: 99 3:100 4: 8   
## 4:131 4: 96   
##   
## Weekly.study.time Number.of.past.class.failures Extra.educational.support  
## 1:105 0:312 FALSE:344   
## 2:198 1: 50 TRUE : 51   
## 3: 65 2: 17   
## 4: 27 3: 16   
##   
##   
## Family.educational.support Extra.paid.class Extra.curricular.activity  
## FALSE:153 FALSE:214 FALSE:194   
## TRUE :242 TRUE :181 TRUE :201   
##   
##   
##   
##   
## Attended.nursery.school Aims.for.higher.education Internet.access  
## FALSE: 81 FALSE: 20 FALSE: 66   
## TRUE :314 TRUE :375 TRUE :329   
##   
##   
##   
##   
## In.romantic.relationship Evaluation.for.family.relationship  
## FALSE:263 1: 8   
## TRUE :132 2: 18   
## 3: 68   
## 4:195   
## 5:106   
##   
## Evaluation.for.free.time.after.school Frequent.going.out.with.friends  
## 1: 19 1: 23   
## 2: 64 2:103   
## 3:157 3:130   
## 4:115 4: 86   
## 5: 40 5: 53   
##   
## Current.health.status Number.of.school.absences Average.result   
## 1: 47 Min. : 0.000 Min. : 1.333   
## 2: 45 1st Qu.: 0.000 1st Qu.: 8.333   
## 3: 91 Median : 4.000 Median :10.667   
## 4: 66 Mean : 5.709 Mean :10.679   
## 5:146 3rd Qu.: 8.000 3rd Qu.:13.333   
## Max. :75.000 Max. :19.333

Write the data set where index row won’t be recorded to be stored as another csv, *“Cleaned.csv”*

write.csv(d1, "Cleaned.csv", row.names = FALSE)

## Feature Extraction

Obtain the features that is needed to train the **Random Forest Model**. **as.factor()** is used to convert the variables to be **categorical/ factor variables** which is the same as the data set processed before

ds <- read.csv("Cleaned.csv", header=TRUE)  
  
ds.model <- data.frame(  
 Age = ds$Age,  
 Time.from.home.to.school = as.factor(ds$Time.from.home.to.school),  
 Number.of.past.class.failures = as.factor(ds$Number.of.past.class.failures),  
 Extra.educational.support = as.factor(ds$Extra.educational.support),  
 Family.educational.support = as.factor(ds$Family.educational.support),  
 Extra.paid.class = as.factor(ds$Extra.paid.class),  
 Extra.curricular.activity = as.factor(ds$Extra.curricular.activity),  
 Attended.nursery.school = as.factor(ds$Attended.nursery.school),  
 Aims.for.higher.education = as.factor(ds$Aims.for.higher.education),  
 Internet.access = as.factor(ds$Internet.access),  
 In.romantic.relationship = as.factor(ds$In.romantic.relationship),  
 Average.result = ds$Average.result)

## Building Prediction Model

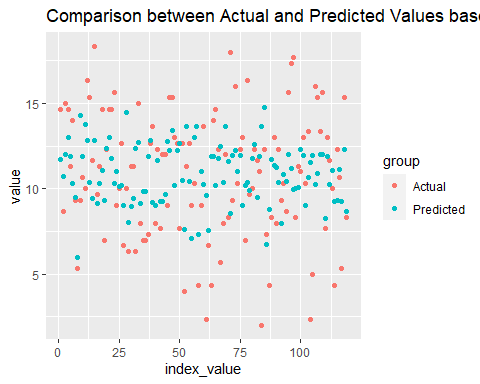
Since the average result is a type of **continuous variable**, therefore it is clearly to use **supervised learning regression model**. There are several regression model can be used for our prediction. We choose **Random Forest Regression model** to predict the average result. Random Forest Regression model is an ensemble learning with multiple decision tree. It has the advantages on dealing with non-linearity.

The data set obtained from the feature extraction is split into two, which used for *training (70%) and testing (30%)*. Based on our testing set result, the prediction model has a **root mean square error (rmse) between 3 to 4**, which is relatively moderate.

# Load dplyr library  
library(dplyr)  
  
# Load randomForest library to use the model  
library(randomForest)  
  
# Split the ds.model to train data frame and test data frame  
dt <- sort(sample(nrow(ds.model), nrow(ds.model)\*0.7))  
train <- ds.model[dt,]  
test <- ds.model[-dt,]  
  
# Remove the average result from the test data frame  
test <- select(test, -Average.result)  
  
# Train the Random Forest model  
model <- randomForest(formula=Average.result ~., data=train)  
  
# Finding root mean square error  
actual <- select(ds.model[-dt,], Average.result)  
names(actual) <- "actual"  
predicted <- predict(model, test)  
cal\_df <- cbind(actual, predicted)  
cal\_df$error <- cal\_df$actual - cal\_df$predicted  
rmse <- sqrt(mean(cal\_df$error^2))  
rmse

## [1] 3.509935

# Plot to see the difference between the actual and predicted based on test set  
value <- select(cal\_df, actual)  
names(value) <- "value"  
index\_value <- c(1:nrow(value))  
group <- "Actual"  
actual\_df <- cbind(value, index\_value, group)  
value <- select(cal\_df, predicted)  
names(value) <- "value"  
index\_value <- c(1:nrow(value))  
group <- "Predicted"  
predicted\_df <- cbind(value, index\_value, group)  
plot\_df <- rbind(actual\_df, predicted\_df)  
ggplot(plot\_df, aes(index\_value, value, colour=group)) + geom\_point() + ggtitle("Comparison between Actual and Predicted Values based on Test Dataset")



## Application development

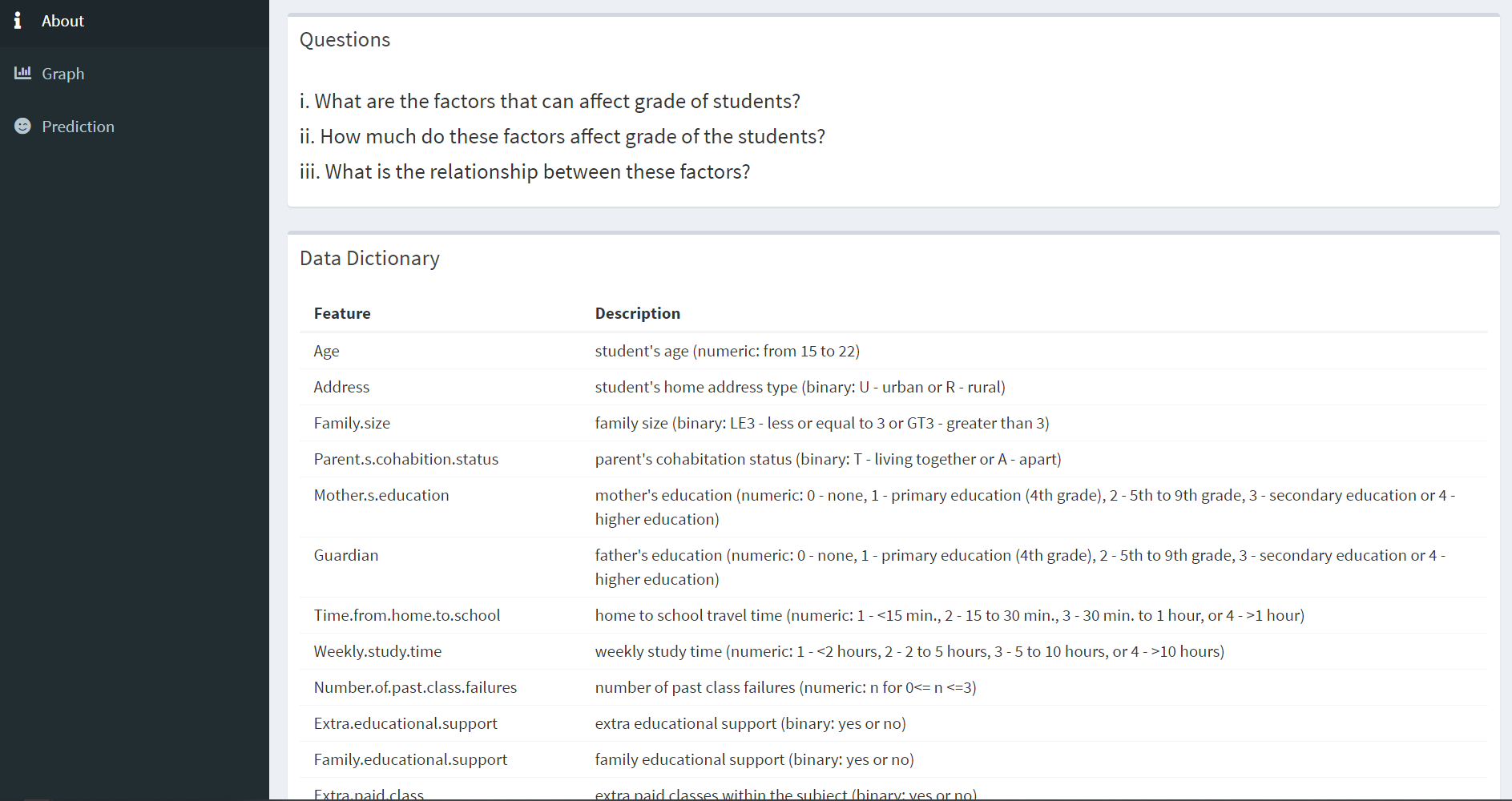
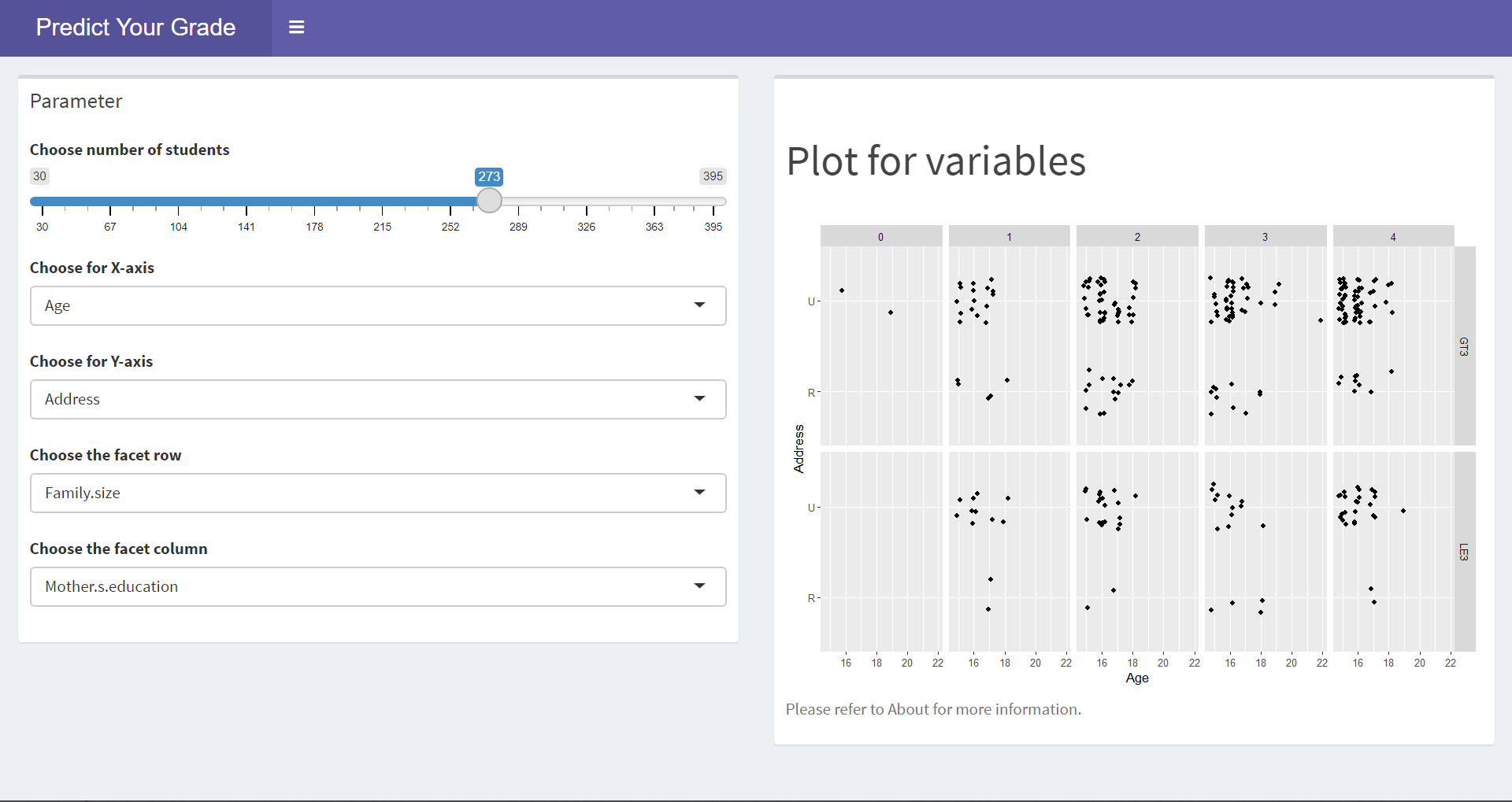
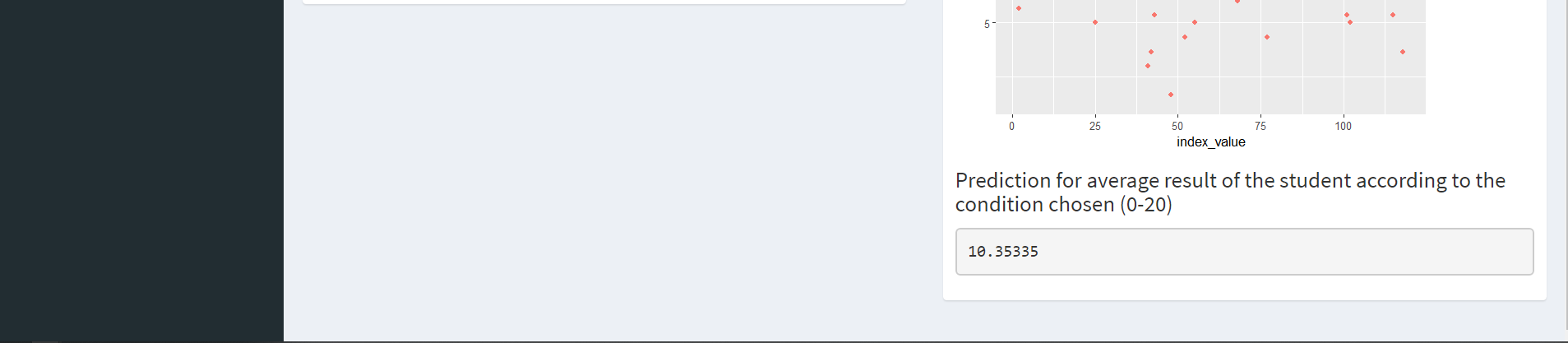
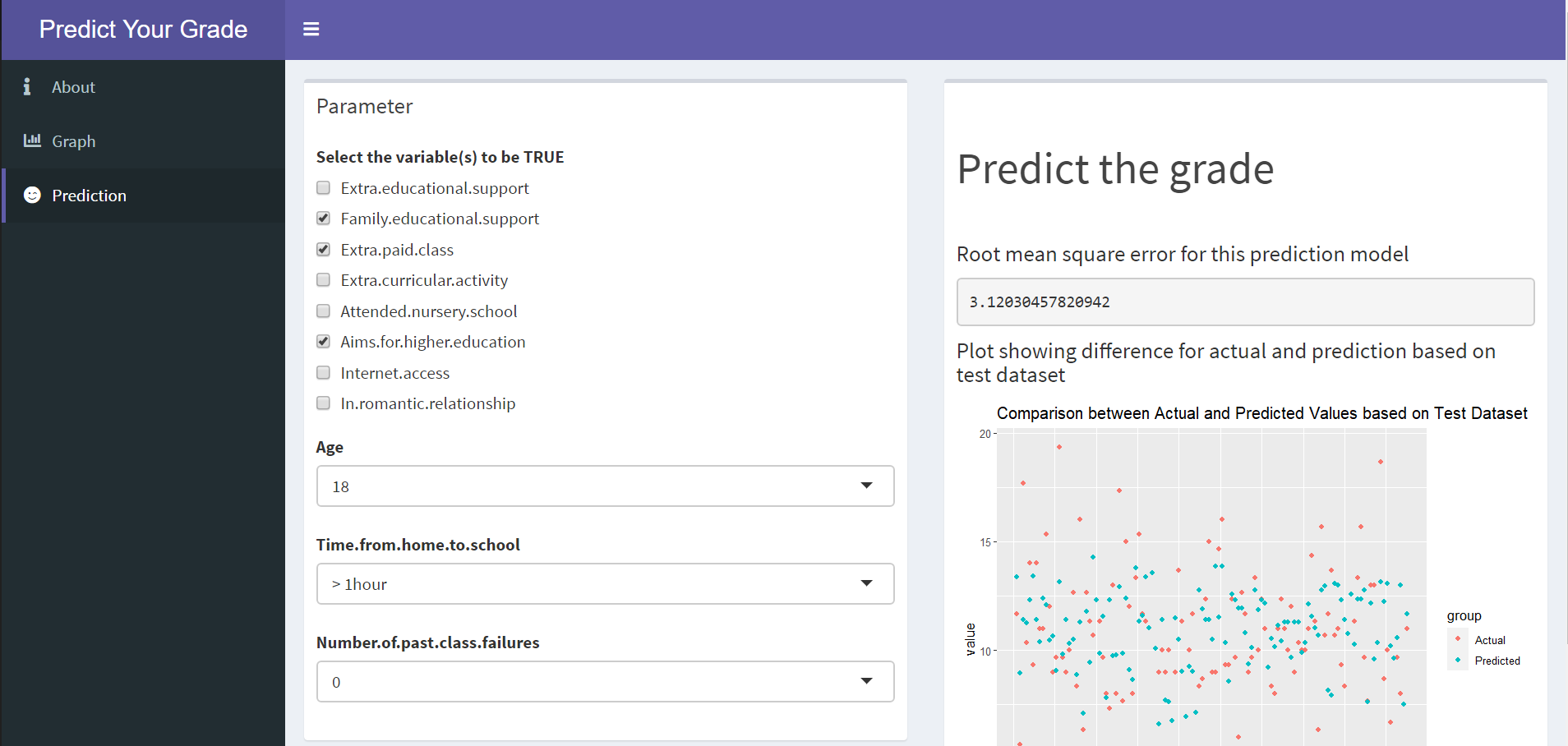
app.R is split into 2 parts, which are ui, in charge of visualizing the dashboard, and server, in charge of manipulation of input and displaying output. There is a total of *3 tabs* for the shiny app we have created, first to be **“About”**, where descriptions of the variable stored and questions proposed would be listed there. Second tab is **“Graph”**, where a jitter/ scatter graph would be shown, based on input by users on X-axis, Y-axis, facet row and also facet column. The last tab is **“Prediction”**, where users can manipulate for conditions according to their likes to predict average result of the student under the selected conditions. Root mean square error for the prediction and a graph showing the difference between prediction and actual results would be shown as well. *Shinydashboard* library is used in beautify the visualization of the product.

Link to the web application: <https://limweiyeong99.shinyapps.io/grade/>

Below show for the source code of app.R

library(dplyr)  
library(ggplot2)  
library(randomForest)  
library(rsconnect)  
library(shiny)  
library(shinydashboard)  
  
# Read data set that is processed and cleaned  
ds <- read.csv("Cleaned.csv", header=TRUE)  
  
# Data dictionary  
Feature <- names(ds)  
Description <- c(  
 "student's age (numeric: from 15 to 22)",  
 "student's home address type (binary: U - urban or R - rural)",  
 "family size (binary: LE3 - less or equal to 3 or GT3 - greater than 3)",  
 "parent's cohabitation status (binary: T - living together or A - apart)",  
 "mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)",  
 "father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)",  
 "home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)",  
 "weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)",  
 "number of past class failures (numeric: n for 0<= n <=3)",  
 "extra educational support (binary: yes or no)",  
 "family educational support (binary: yes or no)",  
 "extra paid classes within the subject (binary: yes or no)",  
 "extra-curricular activities (binary: yes or no)",  
 "attended nursery school (binary: yes or no)",  
 "wants to take higher education (binary: yes or no)",  
 "Internet access at home (binary: yes or no)",  
 "in a romantic relationship (binary: yes or no)",  
 "quality of family relationships (numeric: from 1 - very bad to 5 - excellent)",  
 "free time after school (numeric: from 1 - very low to 5 - very high)",  
 "going out with friends (numeric: from 1 - very low to 5 - very high)",  
 "current health status (numeric: from 1 - very bad to 5 - very good)",  
 "number of school absences (numeric: from 0 to 93)",  
 "average result for the subject within 3 period (continuos: 0 to 20)"  
)  
dict <- cbind(Feature, Description)  
  
# Obtain the features that is needed to train the Random Forest Model  
ds.model <- data.frame(  
 Age = ds$Age,  
 Time.from.home.to.school = as.factor(ds$Time.from.home.to.school),  
 Number.of.past.class.failures = as.factor(ds$Number.of.past.class.failures),  
 Extra.educational.support = as.factor(ds$Extra.educational.support),  
 Family.educational.support = as.factor(ds$Family.educational.support),  
 Extra.paid.class = as.factor(ds$Extra.paid.class),  
 Extra.curricular.activity = as.factor(ds$Extra.curricular.activity),  
 Attended.nursery.school = as.factor(ds$Attended.nursery.school),  
 Aims.for.higher.education = as.factor(ds$Aims.for.higher.education),  
 Internet.access = as.factor(ds$Internet.access),  
 In.romantic.relationship = as.factor(ds$In.romantic.relationship),  
 Average.result = ds$Average.result)  
  
# Split the ds.model to train dataframe and test dataframe  
dt <- sort(sample(nrow(ds.model), nrow(ds.model)\*0.7))  
train <- ds.model[dt,]  
test <- ds.model[-dt,]  
  
# Remove the average result from the test data frame  
test <- select(test, -Average.result)  
  
# Train the Random Forest model  
model <- randomForest(formula=Average.result ~., data=train)  
  
# Finding root mean square error  
actual <- select(ds.model[-dt,], Average.result)  
names(actual) <- "actual"  
predicted <- predict(model, test)  
cal\_df <- cbind(actual, predicted)  
cal\_df$error <- cal\_df$actual - cal\_df$predicted  
rmse <- sqrt(mean(cal\_df$error^2))  
  
# Plot to see the difference between the actual and predicted based on test set  
value <- select(cal\_df, actual)  
names(value) <- "value"  
index\_value <- c(1:nrow(value))  
group <- "Actual"  
actual\_df <- cbind(value, index\_value, group)  
value <- select(cal\_df, predicted)  
names(value) <- "value"  
index\_value <- c(1:nrow(value))  
group <- "Predicted"  
predicted\_df <- cbind(value, index\_value, group)  
plot\_df <- rbind(actual\_df, predicted\_df)  
  
ui <- dashboardPage(  
 title = "Predict your grade",  
 skin = "purple",  
 dashboardHeader(title = "Predict Your Grade"),  
 dashboardSidebar(  
 sidebarMenu(  
 menuItem("About", tabName = "about", icon = icon("info")),  
 menuItem("Graph", tabName = "graph", icon = icon("chart-bar")),  
 menuItem("Prediction", tabName = "prediction", icon = icon("smile-wink"))  
 )  
 ),  
 dashboardBody(  
 tabItems(  
 # First tab content  
 tabItem(tabName = "about",  
 fluidRow(  
 box(  
 width = 12,  
 title = "Questions",  
 h4("i. What are the factors that can affect grade of students?"),  
 h4("ii. How much do these factors affect grade of the students?"),   
 h4("iii. What is the relationship between these factors?")  
 )  
 ),  
 fluidRow(  
 box(  
 width = 12,  
 title = "Data Dictionary",  
 tableOutput("details")  
 )  
 )  
 ),  
 tabItem(tabName = "graph",  
 fluidRow(  
 box(  
 title = "Parameter",  
 sliderInput("Population", "Choose number of students", min = 30, max = 395, value = 200),  
 # Below are inputs for graph  
 selectInput("x\_axis", "Choose for X-axis", choices=names(ds), selected=names(ds)[1]),  
 selectInput("y\_axis", "Choose for Y-axis", choices=names(ds), selected=names(ds)[2]),  
 # None would be represented with "." in syntax, where there will be no chosen variable   
 selectInput('facet\_row', 'Choose the facet row', choices=c(None='.', names(ds))),   
 selectInput('facet\_col', 'Choose the facet column', choices=c(None='.', names(ds)))  
 ),  
 box(  
 title = h1("Plot for variables"),  
 plotOutput("graph"),  
 helpText("Please refer to About for more information.")  
 )  
 )  
   
 ),  
 tabItem(tabName = "prediction",  
 fluidRow(  
 box(  
 title = "Parameter",  
 # Below are inputs for prediction  
 checkboxGroupInput('check\_box', 'Select the variable(s) to be TRUE',  
 choiceNames=c('Extra.educational.support','Family.educational.support','Extra.paid.class',  
 'Extra.curricular.activity', 'Attended.nursery.school', 'Aims.for.higher.education',  
 'Internet.access', 'In.romantic.relationship'),  
 choiceValues=c('Extra.educational.support','Family.educational.support','Extra.paid.class',  
 'Extra.curricular.activity', 'Attended.nursery.school', 'Aims.for.higher.education',  
 'Internet.access', 'In.romantic.relationship')  
 ),  
 selectInput('age', 'Age', choices = as.integer(levels(factor(ds$Age)))),  
 selectInput('time', 'Time.from.home.to.school', choices = c("< 15min","15 - 30min","30min - 1hour", "> 1hour")),  
 selectInput('fail', 'Number.of.past.class.failures', choices = levels(factor(ds$Number.of.past.class.failures)))  
 ),  
 box(  
 title = h1("Predict the grade"),  
 h4("Root mean square error for this prediction model"), verbatimTextOutput("rmse"),  
 h4("Plot showing difference for actual and prediction based on test dataset"), plotOutput("difference"),  
 h4("Prediction for average result of the student according to the condition chosen (0-20)"), verbatimTextOutput("predict")  
 )  
 )  
 )  
 )  
   
 )  
)  
  
server <- function(input, output){  
   
 data\_shown <- reactive({  
 ds[1:input$Population,]  
 })  
   
 # Prevent facet row and column having same variables chosen   
 facet\_row <- reactive({  
 validate(  
 need((input$facet\_row != input$facet\_col) || input$facet\_row == '.',   
 "Please choose another variable for facet row")  
 )  
 input$facet\_row  
 })  
 facet\_column <- reactive({  
 validate(  
 need((input$facet\_row != input$facet\_col) || input$facet\_col == '.',  
 "Please choose another variable for facet row")  
 )  
 input$facet\_col  
 })  
   
 # For graph tab  
 output$graph <- renderPlot({  
 # Jitter graph or scatter graph is drawn to show distribution for the variables  
 p <- ggplot(data\_shown(), aes\_string(x=input$x\_axis, y=input$y\_axis)) +  
 geom\_jitter(width = 0.25, height = 0.25)  
 # Declare for facet row and facet column to be shown in graph in case not a null value  
 facets <- paste(facet\_row(), '~', facet\_column())  
 if (facets != '. ~ .')  
 p <- p + facet\_grid(facets)  
 # Output the graph  
 p  
 })  
   
 # For prediction tab menu  
 output$predict <- renderText({  
 # Code the user input  
 if (input$time == "< 15min") {  
 time = 1  
 }  
 else if (input$time == "15 - 30min") {  
 time = 2  
 }  
 else if (input$time == "30min - 1hour") {  
 time = 3  
 }  
 else if (input$time == "> 1hour"){  
 time = 4  
 }  
 # Let those boolean variable to be in form of factor and FALSE at neutral  
 input\_data <- data.frame(  
 Age = as.integer(input$age),  
 Time.from.home.to.school = factor(time, levels=levels(ds.model$Time.from.home.to.school)),  
 Number.of.past.class.failures = factor(input$fail, levels=levels(ds.model$Number.of.past.class.failures)),  
 Extra.educational.support = factor(FALSE, levels=levels(ds.model$Extra.educational.support)),  
 Family.educational.support = factor(FALSE, levels=levels(ds.model$Family.educational.support)),  
 Extra.paid.class = factor(FALSE, levels=levels(ds.model$Extra.paid.class)),  
 Extra.curricular.activity = factor(FALSE, levels=levels(ds.model$Extra.curricular.activity)),  
 Attended.nursery.school = factor(FALSE, levels=levels(ds.model$Attended.nursery.school)),  
 Aims.for.higher.education = factor(FALSE, levels=levels(ds.model$Aims.for.higher.education)),  
 Internet.access = factor(FALSE, levels = levels(ds.model$Internet.access)),  
 In.romantic.relationship = factor(FALSE, levels=levels(ds.model$In.romantic.relationship))  
 )  
 # If checkbox tick for the boolean variable, set as true for those applicable  
 if(length(input$check\_box) > 0){  
 if("Extra.educational.support" %in% input$check\_box){  
 input\_data$Extra.educational.support = TRUE  
 }  
 if("Family.educational.support" %in% input$check\_box){  
 input\_data$Family.educational.support = TRUE  
 }  
 if("Extra.paid.class" %in% input$check\_box){  
 input\_data$Extra.paid.class = TRUE  
 }  
 if("Extra.curricular.activity" %in% input$check\_box){  
 input\_data$Extra.curricular.activity = TRUE  
 }  
 if("Attended.nursery.school" %in% input$check\_box){  
 input\_data$Attended.nursery.school = TRUE  
 }  
 if("Aims.for.higher.education" %in% input$check\_box){  
 input\_data$Aims.for.higher.education = TRUE  
 }  
 if("Internet.access" %in% input$check\_box){  
 input\_data$Internet.access = TRUE  
 }  
 if("In.romantic.relationship" %in% input$check\_box){  
 input\_data$In.romantic.relationship = TRUE  
 }  
 }  
 test <- rbind(test, input\_data)  
 testrow <- test[nrow(test), ]  
 test <- test[1:nrow(test) - 1, ]  
 predict(model, testrow)  
 })  
 # Render rsme  
 output$rmse <- renderText({  
 paste(sqrt(mean(cal\_df$error^2)))  
 })  
   
 # Render prediction plot to visualise the accuracy  
 output$difference <- renderPlot({  
 ggplot(plot\_df, aes(index\_value, value, colour=group)) + geom\_point() + ggtitle("Comparison between Actual and Predicted Values based on Test Dataset")   
 })  
   
 # Render the data dictionary table  
 output$details <- renderTable({  
 dict  
 })  
   
}  
  
# Run the application   
shinyApp(ui = ui, server = server)

Below show for images taken based on our product.

1. This is **“About”** tab showing details regarding variables and questions proposed 
2. This is **“Graph”** tab showing for the scatter graph 
3. This is **“Predict”** tab showing for the prediction 

We have listed the factors that affect the grade of students in **“About”** tab, making some analysis and visualization for determining how factors affecting the grade as well as relationship between the factors in **“Graph”** tab. Further checking on extend factors affecting grade of students can be determined at **“Predict”** tab.

## Project experience

We learned lot on developing the data product, yet there are still some challenges faced by us in completing the assignment, and they are listed as shown:

1. Defining a good question for the project and finding and appropriate data set regarding the question
2. Inexperience in operation related to prediction and choosing appropriate prediction model
3. Communication with teammates is halted because of the online learning environment
4. Lack of experience in members in R programming language

## Conclusion

There are need for ensuring academic performance of the student to be improved, where this required more effort in determining factors affecting the students and solving them one by one. Suitable helps such as financial support or meeting with parents for further discussion on student’s daily life should be determined. This require attention from not only students, but also parents, school, government and even society since these students who will be leading the world in future and academic performance stands for capability of them.