Student Catagorised GPA Prediction Capstone Project

Rawin Chanpitak

09/01/2020

1.Introduction

This report is another one of the compulsory assignment for **Capstones Projects** from **HarvardX Online Course**. The objective of this capstone is to show that the learner of this couse capable to complete data science project by themselves.

The dataset that is used here is from kaggle website. It is call "student" dataset which contain 33 variables about students who studies in Math course. The goal here is to predict the range of GPA of the students. This can be really helpful as a manager of universities or schools who tries to improve the standard of the student. As the data contain information such as familysize and mid-term score, the final grade could be predicted. The dataset was devided into three sets namely: training set, validation set and test set.

The result of this project is that it is possible to predict the final grade from this data. The final grade is classified as lower than average, around average, and upper than average. The best perform algorithm is tuned knn with 0.738 accuracy for identifying which range a student is in.

2.Data Explanation

In this dataset most of the data is factor or binary. Eventhough the notepad that is given with the pack of data indicate numeric, the numeric, that is given, does not indicate value as it seems. Therefore, they were factorized. The explanation of each data is provided below.

1. school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)
2. sex - student's sex (binary: "F" - female or "M" - male)
3. age - student's age (factor: from 15 to 22)
4. address - student's home address type (binary: "U" - urban or "R" - rural)
5. famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)
6. Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)
7. Medu - mother's education (factor: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
8. Fedu - father's education (factor: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
9. Mjob - mother's job (factor: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
10. Fjob - father's job (factorl: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
11. reason - reason to choose this school (factor: close to "home", school "reputation", "course" preference or "other")
12. guardian - student's guardian (factor: "mother", "father" or "other")
13. traveltime - home to school travel time (factor: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14. studytime - weekly study time (factor: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. failures - number of past class failures (factor: n if 1<=n<3, else 4)
16. schoolsup - extra educational support (binary: yes or no)

17. famsup - family educational support (binary: yes or no)
18. paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19. activities - extra-curricular activities (binary: yes or no)
20. nursery - attended nursery school (binary: yes or no)
21. higher - wants to take higher education (binary: yes or no)
22. internet - Internet access at home (binary: yes or no)
23. romantic - with a romantic relationship (binary: yes or no)
24. famrel - quality of family relationships (factor: from 1 - very bad to 5 - excellent)
25. freetime - free time after school (factor: from 1 - very low to 5 - very high)
26. goout - going out with friends (factor: from 1 - very low to 5 - very high)
27. Dalc - workday alcohol consumption (factor: from 1 - very low to 5 - very high)
28. Walc - weekend alcohol consumption (factor: from 1 - very low to 5 - very high)
29. health - current health status (factor: from 1 - very bad to 5 - very good)
30. absences - number of school absences (factor: from 0 to 93) 31 G1 - first period grade (numeric: from 0 to 20) 31 G2 - second period grade (numeric: from 0 to 20) 32 G3 - final grade (numeric: from 0 to 20, output target)

All of the variables in dataset were used in machine learning except the second period grade. This is because by the time that the second period grade is out, the final grade will be also out if and only if there is no grade for any project or homework.

3. Data Preparation and Exploration

This section provides the method of preparation and some exploration of this data.

3.1 Preparing The Dataset

The data can be download directly from my depository. The data files is comma-spreads value format. However, inside that file, the only line that is seperated by command is header. The data point is seperated by semicolon. Thus, the data frame has one colume. The string operation helps to deal with this situation.

```
# Loading data
student <- data.frame(read.csv("student-mat.csv"))
coltemp <- strsplit(as.character(names(student)), "\\.")[[1]]
colnames(student) <- "temp"
student <- student %>% separate(temp, into = coltemp, sep = ";")
```

After this, all of the column's class were character. Hence, they were changed into factor and numeric as they should be. The variables G2 was removed as mentioned earlier

```
# Change the data from character to factor and numeric
for (i in seq(1:30)) {
  student[,i] <- as.factor(student[,i])
}
student$absences <- as.numeric(as.character(student$absences))
student$G1 <- as.numeric(as.character(student$G1))
student$G3 <- as.numeric(as.character(student$G3))
student <- student %>% select(-G2)
```

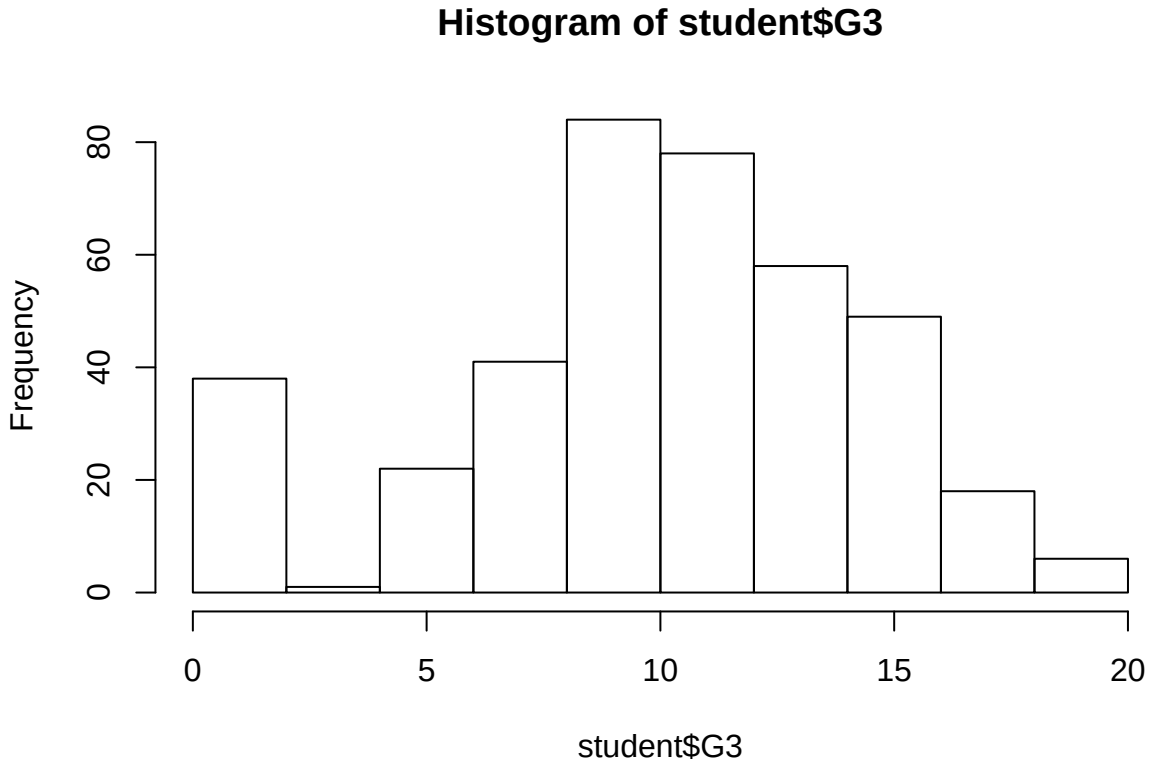
From this, the variables were checked for what they were assign to be

```
sapply(student,class)
```

```
##     school        sex        age    address    famsize    Pstatus       Medu
##   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"
##       Fedu       Mjob       Fjob     reason   guardian traveltime  studytime
##   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"
##   failures  schoolsup     famsup       paid activities    nursery     higher
##   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"
##   internet   romantic     famrel   freetime      goout       Dalc       Walc
##   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"   "factor"
##     health   absences         G1         G3
##   "factor"  "numeric"  "numeric"  "numeric"
```

The distribution of output vectors can be seen below

```
hist(student$G3)
```



The final grade variable was factorized. This is because this project is to predict the range of GPA. The GPA was split equally into 3 part by using quatile function. The student who percentile is below 0.33 were assigned as "lower GPA". The student who percentile is above 0.66 were assigned as "higher GPA". In the middle they were called "average GPA"

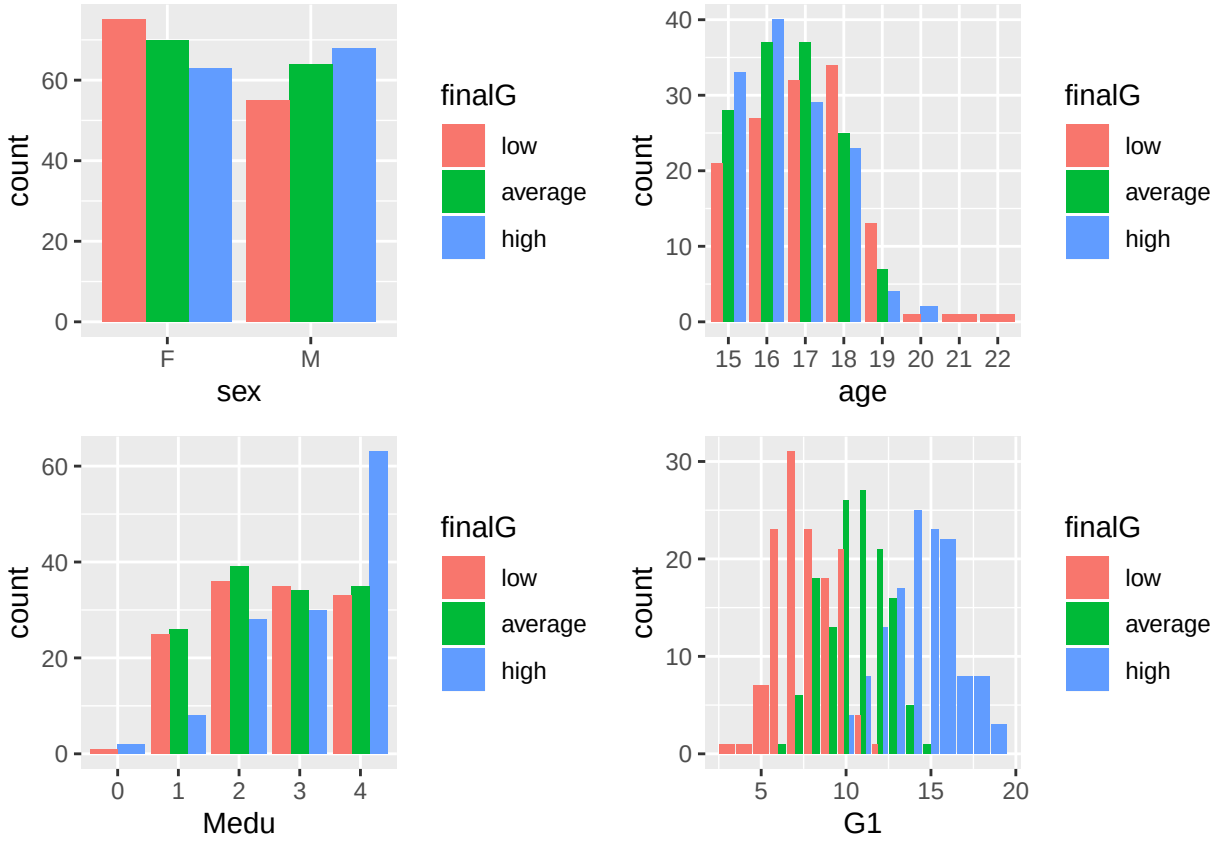
```
per_value <- quantile(student$G3,probs = c(0.33,0.67))
student <- student %>% mutate(finalG = ifelse(G3<per_value[1],0,
                                              ifelse(G3<per_value[2],1,2)))
```

```
student$finalG <- factor(student$finalG, labels = c("low","average","high"))
student <- student %>% select(-G3)
```

3.2 Exploring the Data

The variables in dataset were plot to exploring any relationship between different catagory of Final GPA to see overview of the relationship. It is not nessesary for showing every variables here. therefor some of them should be enough to damonstrate the relationship.

```
plot1 <- student %>% ggplot(aes(sex, fill = finalG)) + geom_bar(position = "dodge")
plot2 <- student %>% ggplot(aes(age, fill = finalG)) + geom_bar(position = "dodge")
plot3 <- student %>% ggplot(aes(Medu, fill = finalG)) + geom_bar(position = "dodge")
plot4 <- student %>% ggplot(aes(G1, fill = finalG)) + geom_bar(position = "dodge")
plot_grid(plot1,plot2,plot3,plot4,nrow = 2)
```



From the graft, The connection can be seen from the data and it give general ideas of realationship between those and the final grade. Firstly, there are roughly the same amount of female and male in this dataset. Male student tend to have more higher grade proportion from theyselves and the opposite goes from female student. Moving on to the age, it is seems that the younger the student the higher grade they have. Thirdly, mothers education tend to give a positive relation with children GPA. Finally, the mid-term score can be a sufficient predictor for final score since it seperates student's GPA significantly.

After data exploration, the output vector was transform the labels from low, average, and high to 1, 2, and 3. This is because it is much easier to construct a ensemble model.

```r
student$finalG <- factor(student$finalG, labels = c("1","2","3"))
```

4. Training The dataset

Before training, the dataset was split into training set, validation set, and testing set with. The testing set had 0.15 proportion of total data and validation set had 0.15 proportion of the rest of remained data

```r
set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
test_index <- createDataPartition(y = student$finalG, times = 1, p = 0.15, list = FALSE)
f_train_stu <- student[-test_index,]
test_stu <- student[test_index,]

v_index <- createDataPartition(y = f_train_stu$finalG, times = 1, p = 0.15, list = FALSE)
train_stu <- f_train_stu[-v_index,]
val_stu <- f_train_stu[v_index,]
```

In this project, the algorithms that were used are Naive Bayes "naive_bayes", support vector machine with linear function "svmLinear", K-nearest-neighbor "knn", generalized additive model with local weighted regression "gamLoess", Penalized Multinomial Regression "multinom", and Random Forest "rf". The algoriths were used to predict the final grade of student. Then, they were compared with each other. Noted that the dataset also not has enough data points to use some of other classification algorithm such as "lda" and "qda". Therefore, they were leaved out from this model

Every training will involve cross-validation technique from train control setting

```r
acc_val_result
```

```
## # A tibble: 6 x 2
##   method      Acc
##   <chr>       <chr>
## 1 naive_bayes 0.431372549019608
## 2 svmLinear   0.588235294117647
## 3 knn         0.588235294117647
## 4 gamLoess    0.411764705882353
## 5 multinom    0.529411764705882
## 6 rf          0.568627450980392
```

This report also adopted ensemble technique to find the best predictor. the ensemble model was constructed by average the predicted out come of the top three model. After receive the accuracy of the model, the top performer was used to predict the data in test set.

The top three accurate models are shown below

```r
acc_val_result %>% arrange(desc(Acc)) %>% select(method) %>% slice(1:3)
```

```
## # A tibble: 3 x 1
##   method
##   <chr>
## 1 svmLinear
## 2 knn
## 3 rf
```

The ensemble model is a bit tedious to construct because output vector is in factor class. Therefore, a lot of data transformations were used in this method

```
# Constructing the ensemble model

target <- acc_val_result %>% arrange(desc(Acc)) %>% select(Acc) %>% slice(3)
top_index <- which(acc_val_result$Acc>(as.numeric(target)-0.001))
top_models <- models[top_index]
top_fits <- fits[top_index]

y_hat_top <- sapply(seq(1:3),function(i){
  y_hat <- predict(top_fits[[i]],val_stu)
})
colnames(y_hat_top) <- models[top_index]
y_hat_top <- as_data_frame(y_hat_top)
```

```
## Warning: `as_data_frame()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

```
# Calculate average of top three model

en_train <- sapply(seq(1:nrow(y_hat_top)),function(i){
  mean(as.numeric(y_hat_top[i,]))
})

# Calculating accuracy for ensemble model

en_train <- as.factor(round(en_train,digits = 0))
acc <- confusionMatrix(data = en_train, reference = val_stu$finalG)$overall["Accuracy"]
acc_val_result <- bind_rows(acc_val_result,
                            data_frame(method = "Ensemble",Acc = as.character(acc)))
acc_val_result
```

```
## # A tibble: 7 x 2
##   method      Acc
##   <chr>       <chr>
## 1 naive_bayes 0.431372549019608
## 2 svmLinear   0.588235294117647
## 3 knn         0.588235294117647
## 4 gamLoess    0.411764705882353
## 5 multinom    0.529411764705882
## 6 rf          0.568627450980392
## 7 Ensemble    0.588235294117647
```

It can be seen that there is no need for ensemble model due to the fact that normal knn and svmLinear perform as same as ensemble model. It should be reasonable to use tuned parameter to yeild better accuracy

from top model. However, there is a problem about tuning svmLinear that I cannot solve and always give error. As a result, the tuned knn and random forest were explore.

```
#tuned knn

knnFit <- train(finalG ~ .,
                data = train_stu,
                method = "knn",
                trControl = ctrl,
                tuneLength = 20)

y_hat <- predict(knnFit,val_stu)
acc <- confusionMatrix(data = y_hat, reference = val_stu$finalG)$overall["Accuracy"]
acc_val_result <- bind_rows(acc_val_result,
                            data_frame(method = "Tuned KNN",Acc = as.character(acc)))
acc_val_result
```

```
## # A tibble: 8 x 2
##   method      Acc
##   <chr>       <chr>
## 1 naive_bayes 0.431372549019608
## 2 svmLinear   0.588235294117647
## 3 knn         0.588235294117647
## 4 gamLoess    0.411764705882353
## 5 multinom    0.529411764705882
## 6 rf          0.568627450980392
## 7 Ensemble    0.588235294117647
## 8 Tuned KNN   0.607843137254902
```

It turned out that tuned knn perform better than plain knn itself and other algorithm. Thus, tuned knn was used in testing data

5. Testing The Model

The plain knn model was trained again with the training set and validation set combined together. After that, the final model was adopt the testing set to see the accuracy on testing set.

```
# tuned knn model

ctrl <- trainControl(method="repeatedcv",
                     repeats=5)
knnFit <-  train(finalG ~ .,
                 method = "knn",
                 data = f_train_stu,
                 trControl=ctrl,
                 tuneLength = 20)

y_hat <- predict(knnFit,test_stu)
acc <- confusionMatrix(data = y_hat, reference = test_stu$finalG)$overall["Accuracy"]
acc
```

```
##  Accuracy
## 0.7377049
```

The accuracy from testing set is higher than expected. This is because the training set is larger than the previous training set. It can be seen that the accuracy of validation set is sufficiently difference with accuracy of testing set. This can be because the data set is not large enough for training set to capture the connection between the variable and final grade.

6. Conclusion

This dataset give information about student who study in Maths. The dataset contain 33 variables. The goal of this project is to classify final grade of the student. The data is analyed in the prediction fashion to identify student who might get good grade at the end of semester. This can be useful in terms of improving the quality of the school by exploring the data. From all of the algorithm that is used, tuned KNN give the best accuracy in validation set. In testing set, the accuracy of tuned knn is 0.738 which is totally better than randomly assign since the output target have three level namely: lower than average, around average, and upper than average.

There are some problem to this project. Firstly, the dataset is not large enough for algorithm to capture relationship between variables. Secondly, eventhough there is cross validation method, the training algorithms are not all tune. Lastly, more algorithm should be used.