Student Performance

Amin Fesharaki

12/6/2021

```
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':
##
##     Recall
```

```
library(C50)
library(nnet)
library(NeuralNetTools)
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'tibble'

## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'
```

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:MLmetrics':
##
##     MAE, RMSE
```

```
library(plyr)
library(rpart)
library(e1071)
library(rpart)
library(rpart.plot)
library(ggplot2)
library(tidyverse)
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'hms'

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
```

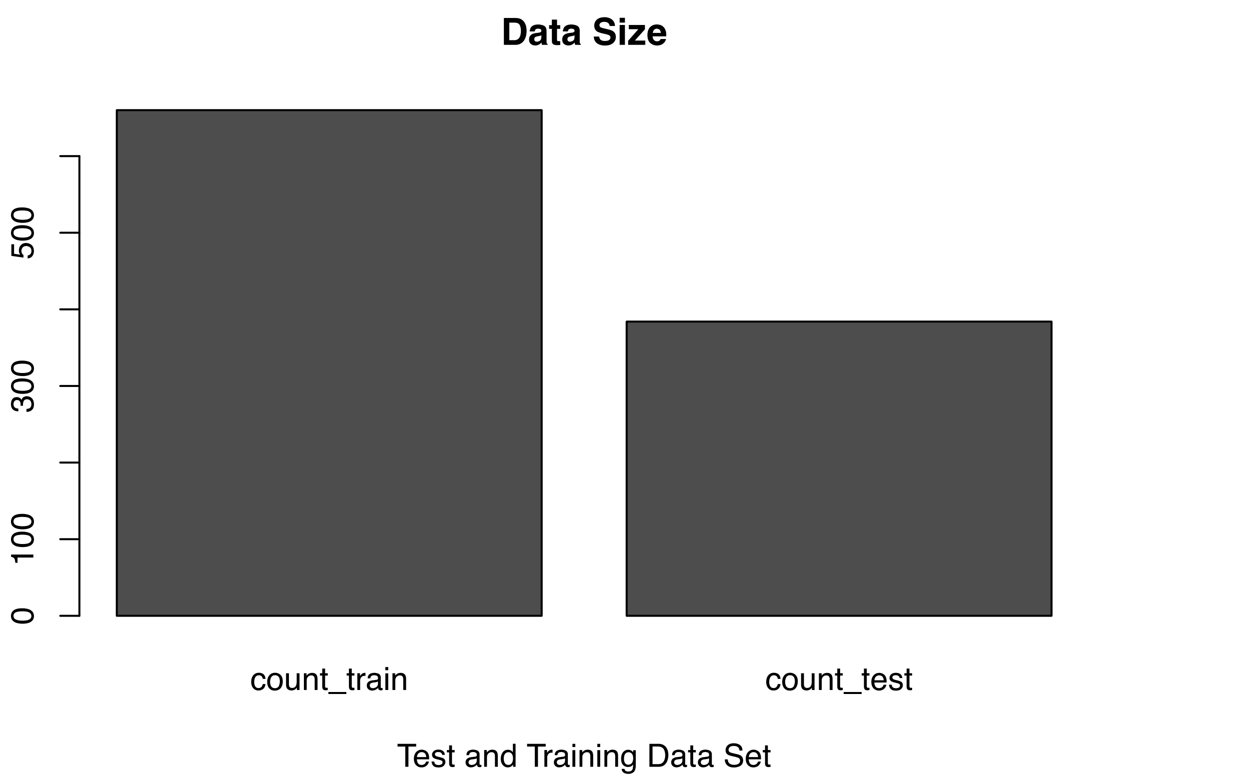
```
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x purrr::lift()      masks caret::lift()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
student_mat <- read.csv('/Users/datascience/Desktop/Project/student-mat.csv', header=TRUE, sep = ",")
student_mat$subject <- "math"
student_por <- read.csv('/Users/datascience/Desktop/Project/student-por.csv', header=TRUE, sep = ",")
student_por$subject <- "portuguese"

student <- rbind(student_mat, student_por)

head(student)
```

```
##   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 subject
## 1        6  5  6  6    math
## 2        4  5  5  6    math
## 3       10  7  8 10    math
## 4        2 15 14 15    math
## 5        4  6 10 10    math
## 6       10 15 15 15    math
```

Data Preparation and Exploratory Data Analysis

```
#Partition Data into Test and Training Datasets

set.seed(7)

n <- dim(student_mat)[1]


train_ind <- runif(n) < 0.67

student_train <- student[ train_ind, ]
student_test <- student[ !train_ind, ]


count_train <- nrow(student_train)
count_test <- nrow(student_test)

counts <- cbind(count_train, count_test)

barplot(counts, main="Data Size",
   xlab="Test and Training Data Set")
```



Find Size of the Data Set

```
print("The size of the training dataset is")
```

```
## [1] "The size of the training dataset is"
```

```
print(dim(student_train))
```

```
## [1] 660  34
```

```r
print("The size of the test dataset is")
```

```
## [1] "The size of the test dataset is"
```

```r
print(dim(student_test))
```

```
## [1] 384  34
```

Checking if Data Values are Balanced

```r
# Purpose of this check is to find if there is a balanced number of values in the data set for variable
print("Check for distrubution of school")
```

```
## [1] "Check for distrubution of school"
```

```r
table(student$school)
```

```
##
##  GP  MS
## 772 272
```

```r
table(student_train$school)
```

```
##
##  GP  MS
## 492 168
```

```r
table(student_test$school)
```

```
##
##  GP  MS
## 280 104
```

```r
print("Check for distrubution of age")
```

```
## [1] "Check for distrubution of age"
```

```r
table(student$age)
```

```
##
##  15  16  17  18  19  20  21  22
## 194 281 277 222  56   9   3   2
```

```r
table(student_train$age)
```

```
##
##  15  16  17  18  19  20  21  22
## 120 183 167 141  39   8   1   1
```

```r
table(student_test$age)
```

```
##
##  15  16  17  18  19  20  21  22
##  74  98 110  81  17   1   2   1
```

```r
print("Check for distrubution of Final Grade")
```

```
## [1] "Check for distrubution of Final Grade"
```

```r
table(student$G3)
```

```
##
##   0   1   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
```

```
## 53    1    1    8   18   19   67   63  153  151  103  113   90   82   52   35   27    7    1
```

```r
table(student_train$G3)
```

```
##
##   0    1    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
##  33    1    4   12   12   41   40  101   92   58   68   56   57   34   26   20    5
```

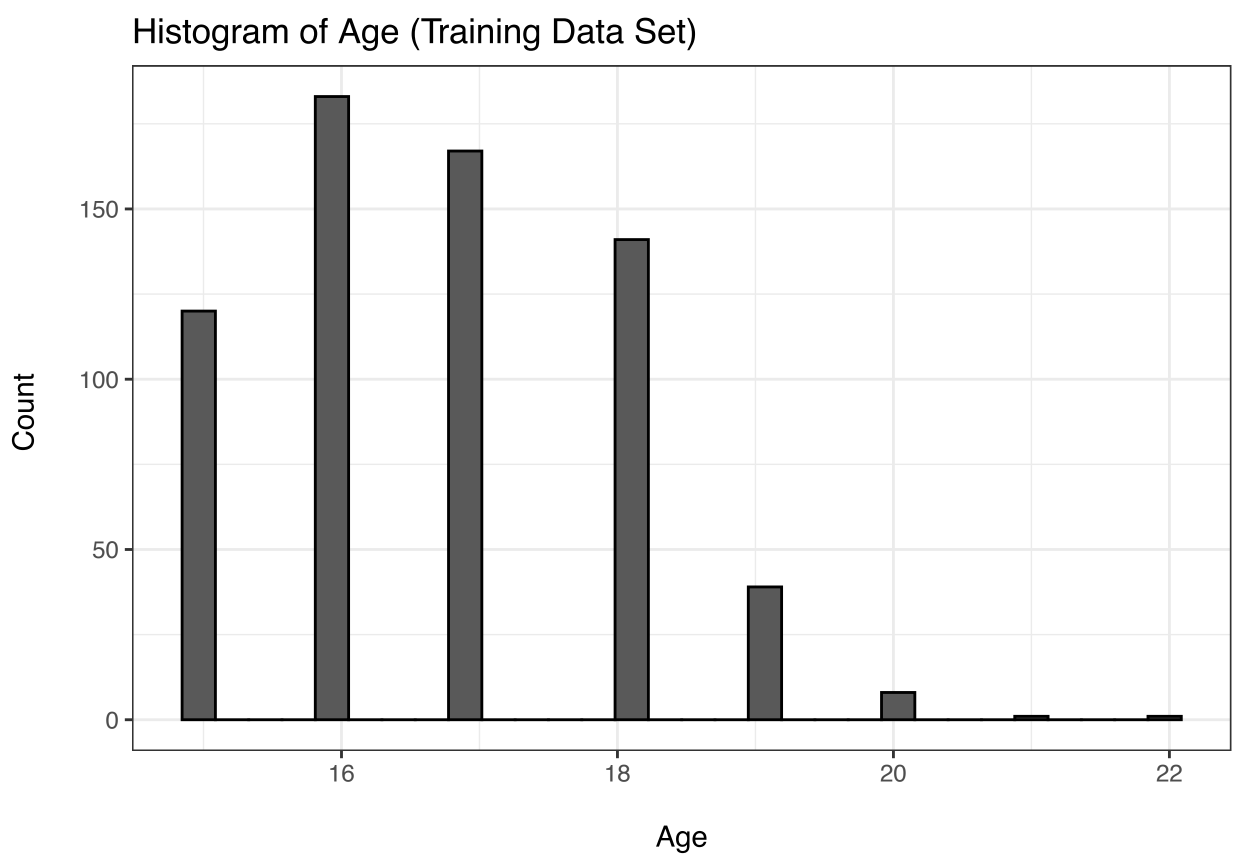
```r
table(student_test$G3)
```

```
##
##   0   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##  20   1   4   6   7  26  23  52  59  45  45  34  25  18   9   7   2   1
```

Visualizing the Data Distribution

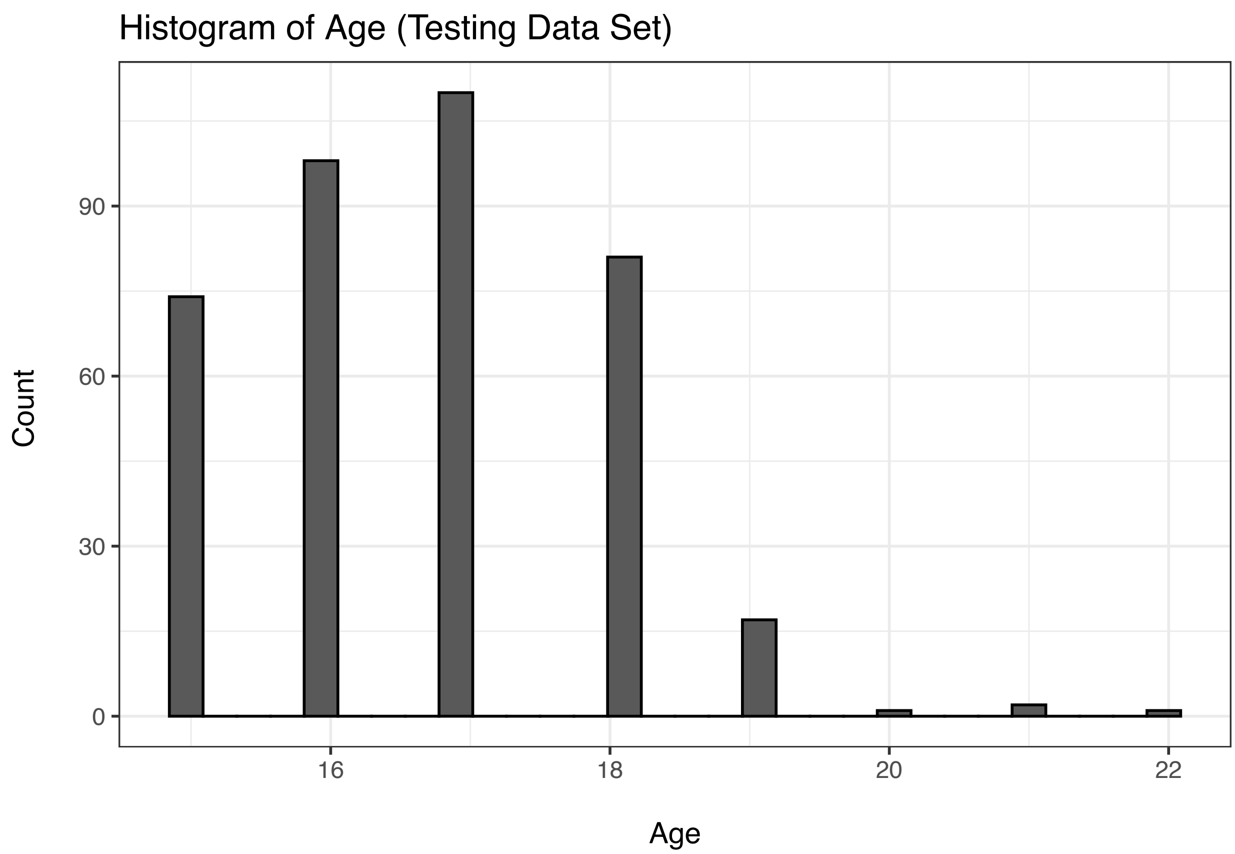
```r
# Purpose of this check is to find if there is a balanced number of values in the data set for variable

#Histogram of Age (Training)
ggplot(student_train, aes(age)) + geom_histogram(color="black")+
labs(x = "\nAge", y = "Count \n")+
ggtitle("Histogram of Age (Training Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



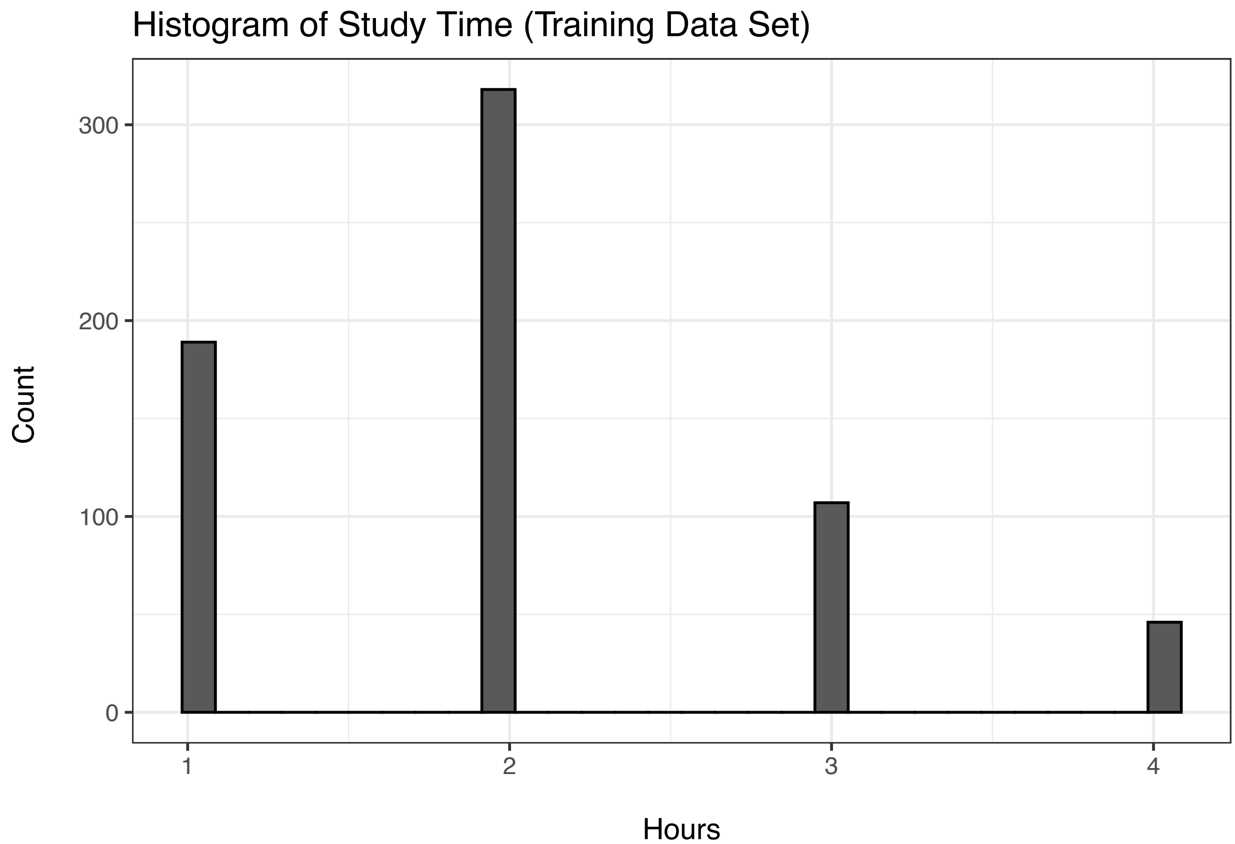

```r
#Histogram of Age (Testing)
ggplot(student_test, aes(age)) + geom_histogram(color="black")+
labs(x = "\nAge", y = "Count \n")+
ggtitle("Histogram of Age (Testing Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



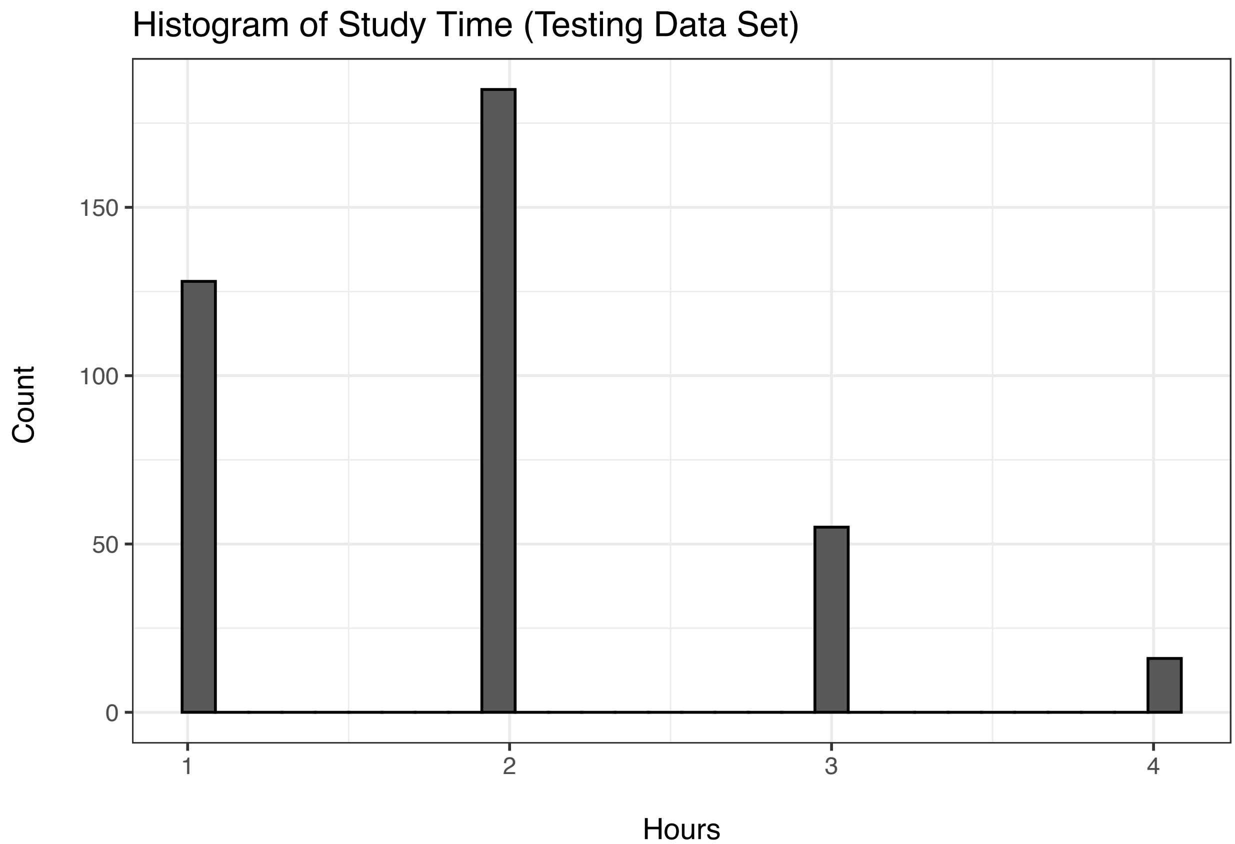

```r
#Histogram of Study Time (Training)
ggplot(student_train, aes(studytime)) + geom_histogram(color="black")+
labs(x = "\nHours", y = "Count \n")+
ggtitle("Histogram of Study Time (Training Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



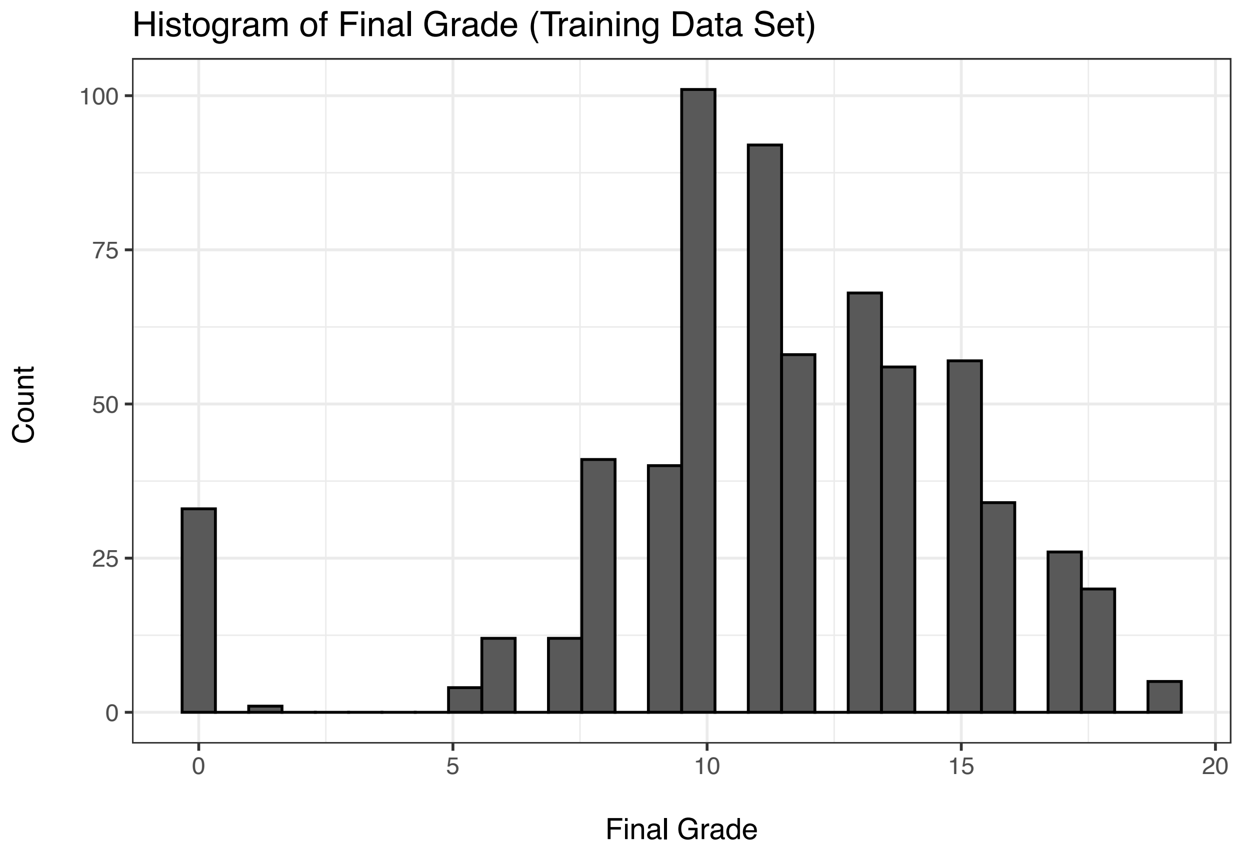

```
#Histogram of Study (Testing)
ggplot(student_test, aes(studytime)) + geom_histogram(color="black")+
labs(x = "\nHours", y = "Count \n")+
ggtitle("Histogram of Study Time (Testing Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



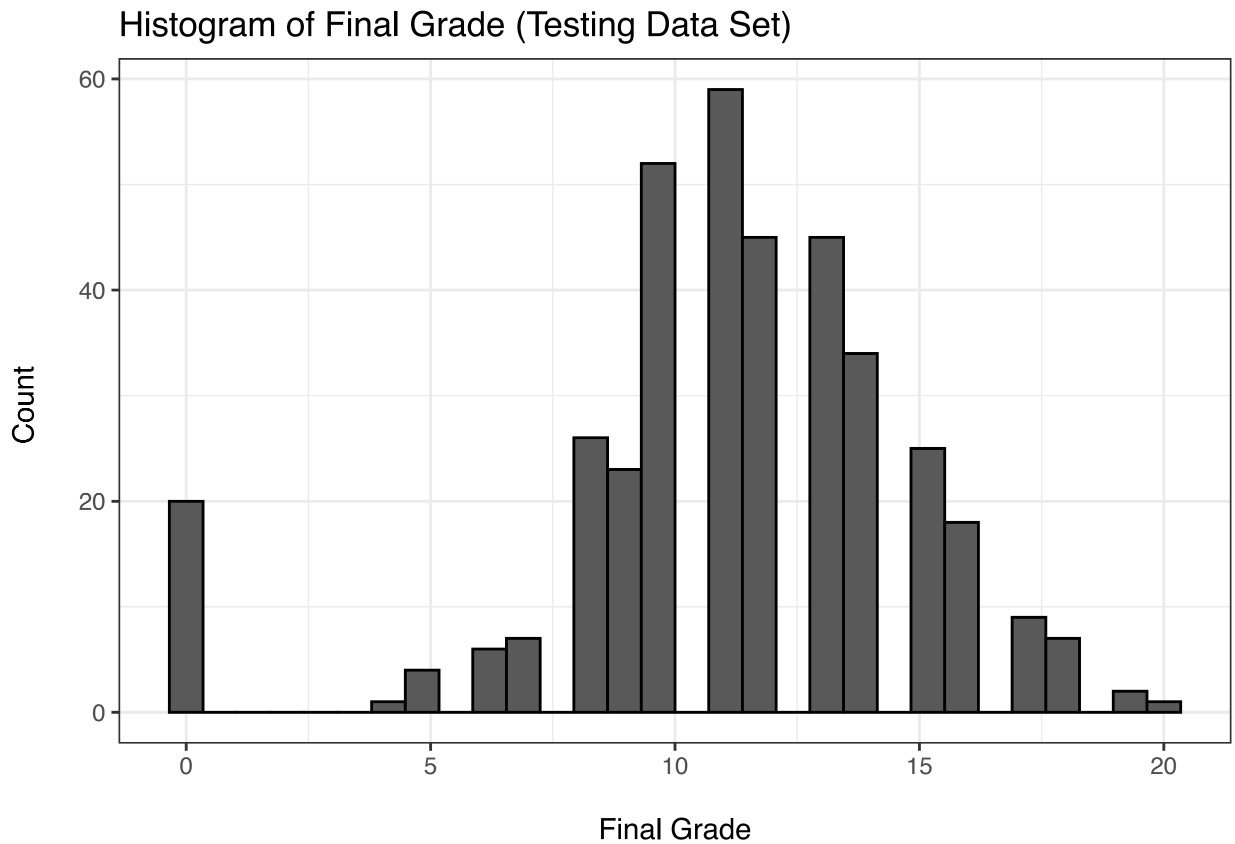

```r
#Histogram of Final Grade (Training)
ggplot(student_train, aes(G3)) + geom_histogram(color="black")+
labs(x = "\nFinal Grade", y = "Count \n")+
ggtitle("Histogram of Final Grade (Training Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



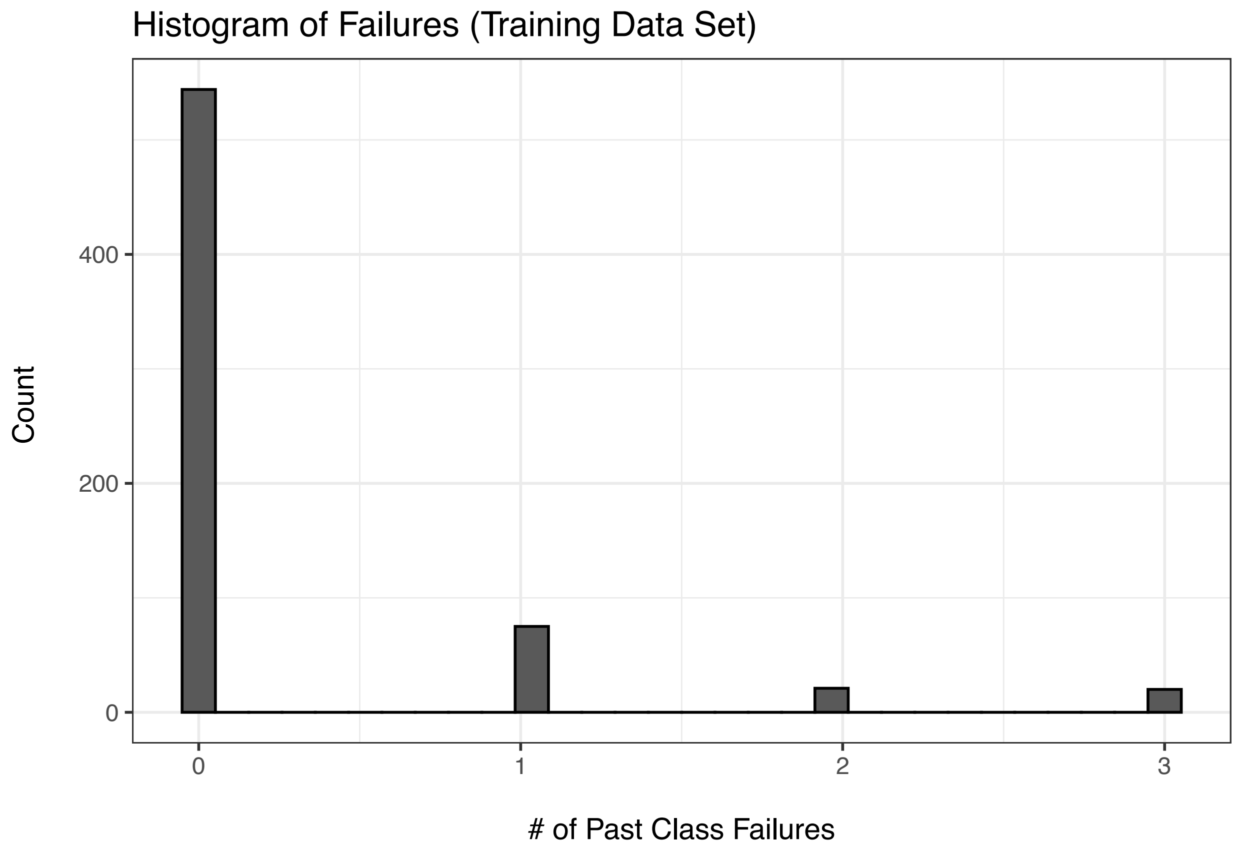

```
#Histogram of Final Grade (Testing)
ggplot(student_test, aes(G3)) + geom_histogram(color="black")+
labs(x = "\nFinal Grade", y = "Count \n")+
ggtitle("Histogram of Final Grade (Testing Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



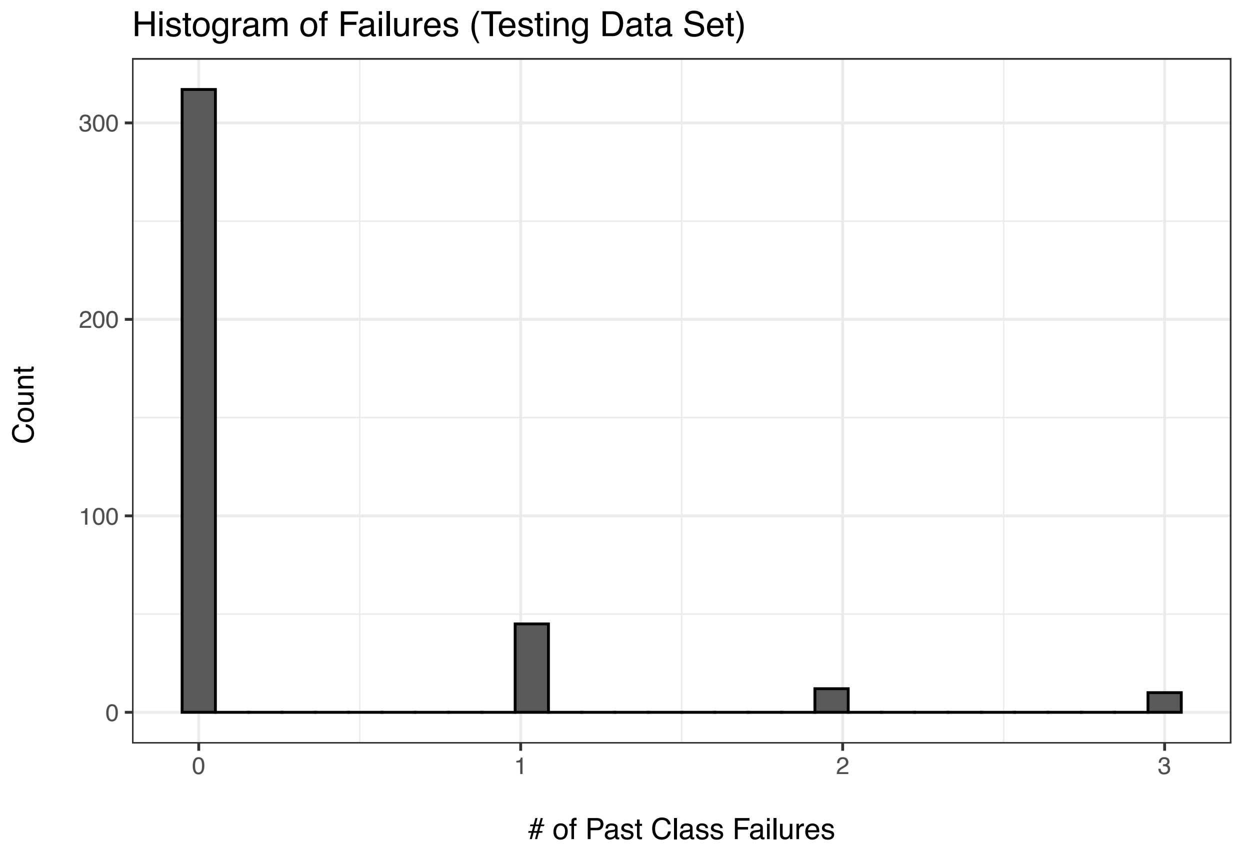

```r
#Histogram of failures (Training)
ggplot(student_train, aes(failures)) + geom_histogram(color="black")+
labs(x = "\n# of Past Class Failures", y = "Count \n")+
ggtitle("Histogram of Failures (Training Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Histogram of failures (Testing)
ggplot(student_test, aes(failures)) + geom_histogram(color="black")+
labs(x = "\n# of Past Class Failures", y = "Count \n")+
ggtitle("Histogram of Failures (Testing Data Set)") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Linear Regression

Creating a Linear Regression Model

```
#subset the training dataset for only variables to be used in regression

student_train_linear_subset <- subset(student_train, select = c("age", "traveltime", "studytime", "failu



# Now, we standardize both predictor variables and save the output as a data
# frame. Data frame format is required for running the kmeans() command
student_train_linear_subset_z <- as.data.frame(scale(student_train_linear_subset))



model01 <- lm(formula = G3 ~ age + traveltime + studytime + failures + famrel + freetime + Dalc + absenc
              data = student_train_linear_subset_z)

summary(model01)
```

```
##
## Call:
## lm(formula = G3 ~ age + traveltime + studytime + failures + famrel +
##     freetime + Dalc + absences + G1 + G2, data = student_train_linear_subset_z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.40707 -0.10841  0.02476  0.19578  1.45103
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.273e-16  1.575e-02   0.000 1.000000
## age         -7.432e-04  1.682e-02  -0.044 0.964768
## traveltime   4.127e-02  1.599e-02   2.580 0.010092 *
## studytime   -1.123e-02  1.639e-02  -0.685 0.493386
## failures    -6.075e-02  1.813e-02  -3.350 0.000853 ***
## famrel       1.767e-02  1.632e-02   1.083 0.279251
## freetime     2.134e-02  1.650e-02   1.294 0.196243
## Dalc        -3.112e-02  1.654e-02  -1.881 0.060361 .
## absences     4.962e-02  1.613e-02   3.076 0.002185 **
## G1           1.024e-01  3.398e-02   3.013 0.002686 **
## G2           8.025e-01  3.405e-02  23.570  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4047 on 649 degrees of freedom
## Multiple R-squared:  0.8387, Adjusted R-squared:  0.8362
## F-statistic: 337.4 on 10 and 649 DF,  p-value: < 2.2e-16
```

From the summary of the regression model, we find that a number of the predictor variables are not significant to the model. The variables that do not show significance, or show a very low significance compared to our threshold of a 0.05 significance level include age, traveltime, studytime, famrel, freetime, and Dalc.

Creating an improved model without the insignificant predictor variables.

```
model02 <- lm(formula = G3 ~ traveltime + failures + absences + G1 + G2,
              data = student_train_linear_subset_z)

summary(model02)
```

```
##
## Call:
## lm(formula = G3 ~ traveltime + failures + absences + G1 + G2,
##     data = student_train_linear_subset_z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.43157 -0.10058  0.02146  0.20575  1.40937
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.028e-16  1.578e-02   0.000 1.000000
## traveltime   3.881e-02  1.590e-02   2.441 0.014922 *
## failures    -6.056e-02  1.727e-02  -3.507 0.000483 ***
## absences     4.443e-02  1.589e-02   2.796 0.005329 **
## G1           1.034e-01  3.385e-02   3.053 0.002357 **
## G2           8.034e-01  3.393e-02  23.678  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4054 on 654 degrees of freedom
## Multiple R-squared:  0.8369, Adjusted R-squared:  0.8356
## F-statistic:   671 on 5 and 654 DF,  p-value: < 2.2e-16
```

All the variables in the new model are significant with p-values less than (0.05).

Next we will use the model to predict G3 scores from the test data.

```
#subset the training dataset for only variables to be used in regression

student_test_linear_subset <- subset(student_test, select = c("age", "traveltime", "studytime", "failur



# Now, we standardize both predictor variables and save the output as a data
# frame. Data frame format is required for running the kmeans() command
student_test_linear_subset_z <- as.data.frame(scale(student_test_linear_subset))



predictions <- predict(object=model02, newdata=student_test_linear_subset_z)


print("MAE Regression is:")
```

```
## [1] "MAE Regression is:"
```

```
MAE(student_test_linear_subset_z$G3, predictions)
```

```
## [1] 0.2426327
```

```
average_y = mean(student_test_linear_subset_z$G3)

print("MAE Baseline is:")
```

```
## [1] "MAE Baseline is:"
```

```
MAE(average_y, predictions)
```

```
## [1] 0.6826324
```

The mean average error of the regression prediction results are lower than the baseline which means that the model's results are better than the baseline model.

Next we chose to explore if using only highly significant variables, variables with p-value less than 0.01, would lead to an even more accurate model. Therefore we removed the feature 'traveltime' from the mode.

Create a new Model using only highly significant variables

```
model03 <- lm(formula = G3 ~ failures + absences + G1 + G2,
              data = student_train_linear_subset_z)

summary(model03)
```

```
##
## Call:
## lm(formula = G3 ~ failures + absences + G1 + G2, data = student_train_linear_subset_z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.46232 -0.09473  0.01351  0.20614  1.54422
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  2.882e-16  1.584e-02   0.000 1.000000
## failures    -5.806e-02  1.730e-02  -3.356 0.000837 ***
## absences     4.468e-02  1.595e-02   2.801 0.005244 **
## G1           1.016e-01  3.397e-02   2.990 0.002893 **
## G2           8.023e-01  3.406e-02  23.559  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.407 on 655 degrees of freedom
## Multiple R-squared:  0.8354, Adjusted R-squared:  0.8344
## F-statistic:   831 on 4 and 655 DF,  p-value: < 2.2e-16
```

After creating the model we find the MAE and compare to the earlier model

```r
# Create predictions using new model
predictions <- predict(object=model03, newdata=student_test_linear_subset_z)


print("MAE Regression is:")
```

```
## [1] "MAE Regression is:"
```

```r
#MAE(y_pred=predictions, y_true=student_test_linear_subset_z$G3)
MAE(student_test_linear_subset_z$G3, predictions)
```

```
## [1] 0.2413168
```

```r
average_y = mean(student_test_linear_subset_z$G3)

print("MAE Baseline is:")
```

```
## [1] "MAE Baseline is:"
```

```r
#MAE(y_pred=predictions, y_true=average_y)
MAE(average_y, predictions)
```

```
## [1] 0.6864503
```

The MAE is a small amount lower compared to the 2nd model but it does not show to be a large difference compared to the previous model.

We again create a 4th model but using features of only the highest level of significance.

```r
model04 <- lm(formula = G3 ~ failures + G2,
              data = student_train_linear_subset_z)

summary(model04)
```

```
##
## Call:
## lm(formula = G3 ~ failures + G2, data = student_train_linear_subset_z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.49942 -0.08432 -0.02376  0.20183  1.50134
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.185e-16  1.601e-02   0.000 1.000000
```

```
## failures    -6.191e-02  1.741e-02  -3.556 0.000404 ***
## G2           8.858e-01  1.741e-02  50.876  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4114 on 657 degrees of freedom
## Multiple R-squared:  0.8313, Adjusted R-squared:  0.8307
## F-statistic:  1618 on 2 and 657 DF,  p-value: < 2.2e-16
```

```
# Create predictions using new model
predictions <- predict(object=model04, newdata=student_test_linear_subset_z)


print("MAE Regression is:")
```

```
## [1] "MAE Regression is:"
```

```
MAE(student_test_linear_subset_z$G3, predictions)
```

```
## [1] 0.2426601
```

```
average_y = mean(student_test_linear_subset_z$G3)

print("MAE Baseline is:")
```

```
## [1] "MAE Baseline is:"
```

```
MAE(average_y, predictions)
```

```
## [1] 0.6785072
```

Compared to model 3, the MAE actually increased when including only features that had the highest level of signifiance. This showed us that removing features from the data set potentially lowered the performance of the model rather than improved it.

Knowing this, the team would recommend to use the 2nd model when attempting to estimate a student's final score and theorize that the variables of traveltime, failures, absences, G1 and G2 are the most important in estimating a student's final exam performance.

Classification

Are there significant differences in the grades that students are receiving based off of the school that they attend? Using student attributes can we predict the school a student attends?

```
# For train for classification
student_train_class <- subset(student_train, select = c("school", "sex", "address", "famsize", "Pstatus

student_train_class$school <- factor(student_train_class$school)
student_train_class$sex <- factor(student_train_class$sex)
student_train_class$address <- factor(student_train_class$address)
student_train_class$famsize <- factor(student_train_class$famsize)
student_train_class$Pstatus <- factor(student_train_class$Pstatus)
student_train_class$schoolsup <- factor(student_train_class$schoolsup)
student_train_class$famsup <- factor(student_train_class$famsup)
student_train_class$activities <- factor(student_train_class$activities)
student_train_class$higher <- factor(student_train_class$higher)



# min - max Standardization
```

```
student_train_class$age.mm <- (student_train_class$age - min(student_train_class$age)) / (max(student_tr

student_train_class$traveltime.mm <- (student_train_class$traveltime - min(student_train_class$travelti

student_train_class$studytime.mm <- (student_train_class$studytime - min(student_train_class$studytime)

student_train_class$failures.mm <- (student_train_class$failures - min(student_train_class$failures)) /

student_train_class$famrel.mm <- (student_train_class$famrel - min(student_train_class$famrel)) / (max(s

student_train_class$freetime.mm <- (student_train_class$freetime - min(student_train_class$freetime)) /

student_train_class$Dalc.mm <- (student_train_class$Dalc - min(student_train_class$Dalc)) / (max(student

student_train_class$absences.mm <- (student_train_class$absences - min(student_train_class$absences)) /

student_train_class$G1.mm <- (student_train_class$G1 - min(student_train_class$G1)) / (max(student_train

student_train_class$G2.mm <- (student_train_class$G2 - min(student_train_class$G2)) / (max(student_train

student_train_class$G3.mm <- (student_train_class$G3 - min(student_train_class$G3)) / (max(student_train

#Add new column where Final passing grade (14+/20) = 0 and final failing grade (13-/20) = 1
student_train_class$G3.p[which(student_train_class$G3<13)] <- 1
student_train_class$G3.p[which(student_train_class$G3>=13)] <- 0

student_train_class$G3.pp[which(student_train_class$G3<13)] <- "Fail"
student_train_class$G3.pp[which(student_train_class$G3>=13)] <- "Pass"
student_train_class$G3.pp <- factor(student_train_class$G3.pp)


# For test for classification
student_test_class <- subset(student_test, select = c("school", "sex", "address", "famsize", "Pstatus",

student_test_class$school <- factor(student_test_class$school)
student_test_class$sex <- factor(student_test_class$sex)
 student_test_class$address <- factor(student_test_class$address)
student_test_class$famsize <- factor(student_test_class$famsize)
student_test_class$Pstatus <- factor(student_test_class$Pstatus)
student_test_class$schoolsup <- factor(student_test_class$schoolsup)
student_test_class$famsup <- factor(student_test_class$famsup)
student_test_class$activities <- factor(student_test_class$activities)
student_test_class$higher <- factor(student_test_class$higher)


# min - max Standardization
student_test_class$age.mm <- (student_test_class$age - min(student_test_class$age)) / (max(student_test_

student_test_class$traveltime.mm <- (student_test_class$traveltime - min(student_test_class$traveltime)

student_test_class$studytime.mm <- (student_test_class$studytime - min(student_test_class$studytime)) /

student_test_class$failures.mm <- (student_test_class$failures - min(student_test_class$failures)) / (ma
```

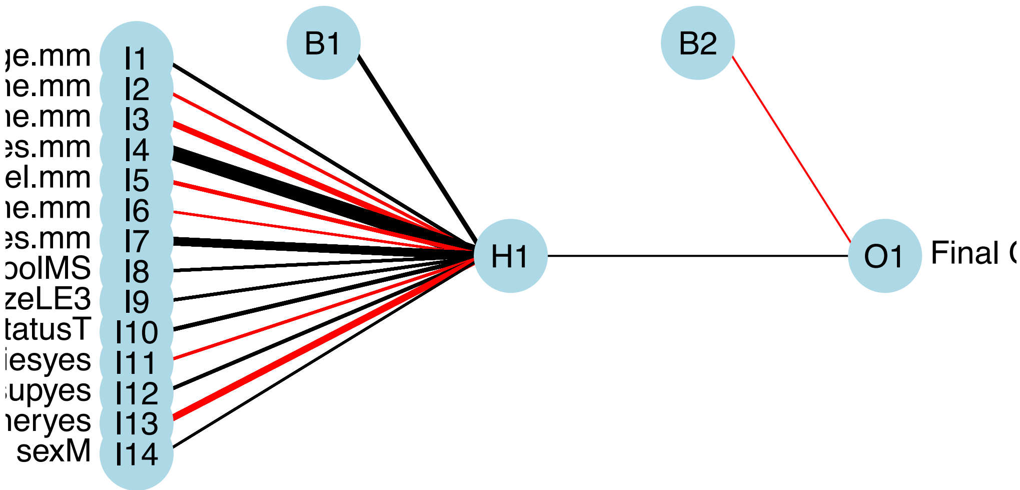
```r
student_test_class$famrel.mm <- (student_test_class$famrel - min(student_test_class$famrel)) / (max(stu

student_test_class$freetime.mm <- (student_test_class$freetime - min(student_test_class$freetime)) / (ma

student_test_class$Dalc.mm <- (student_test_class$Dalc - min(student_test_class$Dalc)) / (max(student_te

student_test_class$absences.mm <- (student_test_class$absences - min(student_test_class$absences)) / (ma

student_test_class$G1.mm <- (student_test_class$G1 - min(student_test_class$G1)) / (max(student_test_cla

student_test_class$G2.mm <- (student_test_class$G2 - min(student_test_class$G2)) / (max(student_test_cla

student_test_class$G3.mm <- (student_test_class$G3 - min(student_test_class$G3)) / (max(student_test_cla

#Add new column where Final passing grade (14+/20) = 0 and final failing grade (13-/20) = 1
student_test_class$G3.p[which(student_test_class$G3<13)] <- 1
student_test_class$G3.p[which(student_test_class$G3>=13)] <- 0

student_test_class$G3.pp[which(student_test_class$G3<13)] <- "Fail"
student_test_class$G3.pp[which(student_test_class$G3>=13)] <- "Pass"
student_test_class$G3.pp <- factor(student_test_class$G3.pp)
```

Create and Plot Neural Network

```r
# Creating Neural
nnet01 <- nnet (G3.p ~ age.mm + traveltime.mm + studytime.mm + failures.mm + famrel.mm + freetime.mm + a
```

```
## # weights:  17
## initial  value 179.724986
## iter  10 value 133.385582
## iter  20 value 128.231824
## iter  30 value 127.878964
## iter  40 value 127.090811
## iter  50 value 126.838306
## iter  60 value 126.561512
## iter  70 value 124.662922
## iter  80 value 124.387390
## iter  90 value 124.160978
## iter 100 value 124.042855
## final  value 124.042855
## stopped after 100 iterations
```

```r
# Plot the neural network.

plotnet(nnet01, neg_col = "red",  y_names = "Final Grade (")
```




```r
# make predictions (returns probabilities)
student_train_class$pred_prob <- predict(object = nnet01, newdata = student_train_class)

#Plot
#chr string indicating color of positive connection weights,  'black'
#chr string indicating color of negative connection weights, 'red'
```

Output the Neural Network Weights

```r
neuralweights(nnet01)
```

```
## $struct
## [1] 14  1  1
##
## $wts
## $wts$`hidden 1 1`
##  [1]  184.17330   90.33874  -57.88420 -241.76452  772.55858 -131.62590
##  [7]  -19.03313  416.10166   70.22659   70.77254  128.35685  -58.67979
## [13]   97.33782 -280.53110   31.38826
##
## $wts$`out 1`
## [1] -0.4325168  2.2087788
```

```r
nnet01$wts
```

```
##  [1]  184.1733021   90.3387392  -57.8842027 -241.7645153  772.5585847
##  [6] -131.6258987  -19.0331321  416.1016610   70.2265947   70.7725415
## [11]  128.3568526  -58.6797895   97.3378177 -280.5310977   31.3882585
## [16]   -0.4325168    2.2087788
```

Evaluate Neural Network

```r
#Evaluate the neural network model using the test dataset.Construct a contingency table to compare the

# make predictions (returns probabilities)
student_test_class$pred_prob_test <- predict(object = nnet01, newdata = student_test_class)
# convert to classes
student_test_class$pred_test <- (student_test_class$pred_prob_test > 0.5)*1

# performance metrics / Confusion Matrix
student_test_class[c('G3.p', 'pred_test')] <- lapply(student_test_class[c('G3.p', 'pred_test')], as.fac
```

```r
confusionMatrix(student_test_class$pred_test, student_test_class$G3.p, positive='1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  99 102
##          1  42 141
##
##                Accuracy : 0.625
##                  95% CI : (0.5745, 0.6736)
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : 0.646
##
##                   Kappa : 0.2592
##
##  Mcnemar's Test P-Value : 8.803e-07
##
##             Sensitivity : 0.5802
##             Specificity : 0.7021
##          Pos Pred Value : 0.7705
##          Neg Pred Value : 0.4925
##              Prevalence : 0.6328
##          Detection Rate : 0.3672
##    Detection Prevalence : 0.4766
##       Balanced Accuracy : 0.6412
##
##        'Positive' Class : 1
##
```

```r
cm
```

```
## function (x)
## 2.54 * x
## <bytecode: 0x7f94017a8378>
## <environment: namespace:grDevices>
```

Contingency Table for Neural Network

```r
 #Contingency Table
c.pred <- table(student_test_class$G3.p, student_test_class$pred_test)
rownames(c.pred) <- c("Actual: No", "Actual: Yes")
colnames(c.pred) <- c("Predicted: No", "Predicted: Yes")
addmargins(A = c.pred, FUN = list(Total=sum), quiet = TRUE)
```

```
##
##               Predicted: No Predicted: Yes Total
##   Actual: No             99             42   141
##   Actual: Yes           102            141   243
##   Total                 201            183   384
```

```r
TN0 <- c.pred[1,1]
FN0 <- c.pred[2,1]
FP0 <- c.pred[1,2]
TP0 <- c.pred[2,2]
```

Decision Trees

```r
# Setting up Predictions for cart, c5.0 and NB with same predictors as Neural Network
X = data.frame(age.mm = student_test_class$age.mm, traveltime.mm = student_test_class$traveltime.mm, fa
```

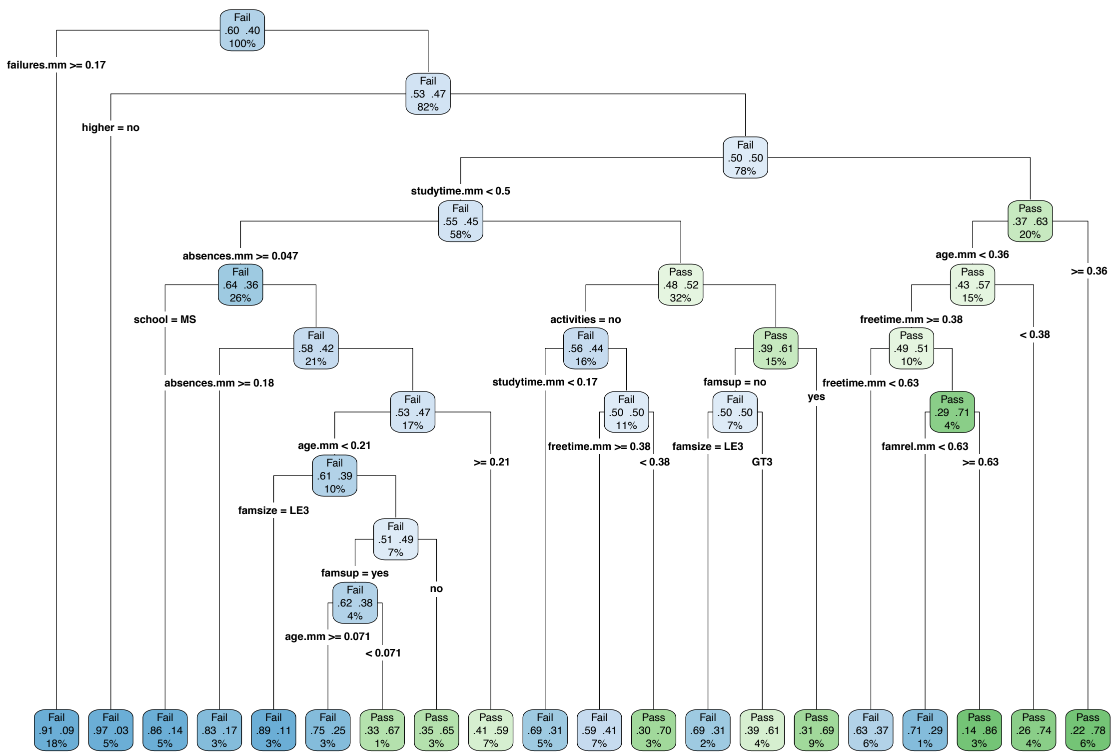
#(a) Cart

```r
# Cart Model trained by training data set
cart  <- rpart(formula = G3.pp ~ age.mm + traveltime.mm + studytime.mm + failures.mm + famrel.mm + free

student_test_class$pred_cart <- predict(object = cart, newdata = X)

# Predictions Test Data Set
Pred_cart = predict(object = cart, newdata = X, type = "class")
head(Pred_cart)
```

```
##    1    2    3    4    5    6
## Pass Fail Pass Pass Fail Fail
## Levels: Fail Pass
```

Cart Visual

```r
rpart.plot(cart,type = 4, extra =104, tweak = 1.5)
```




```r
#Type 4 - Like 3 but label all nodes, not just leaves. Similar to text.rpart's fancy=TRUE. See also cli

#Extra 4 - Class models: probability per class of observations in the node (conditioned on the node, su
```

Cart Evaluation

```r
# Evaluation Metrics for Cart
cart.pred <- table(student_test_class$G3.p, Pred_cart)
rownames(cart.pred) <- c("Actual: No", "Actual: Yes")
colnames(cart.pred) <- c("Predicted: No", "Predicted: Yes")
addmargins(A = cart.pred, FUN = list(Total=sum), quiet = TRUE)
```

```
##              Pred_cart
##               Predicted: No Predicted: Yes Total
##   Actual: No             77            64   141
##   Actual: Yes           168            75   243
##   Total                 245           139   384
```

```
# Assigning General Form of Table to matrix values for Cart
TN1 <- cart.pred[1,1]
FN1 <- cart.pred[2,1]
FP1 <- cart.pred[1,2]
TP1 <- cart.pred[2,2]
```

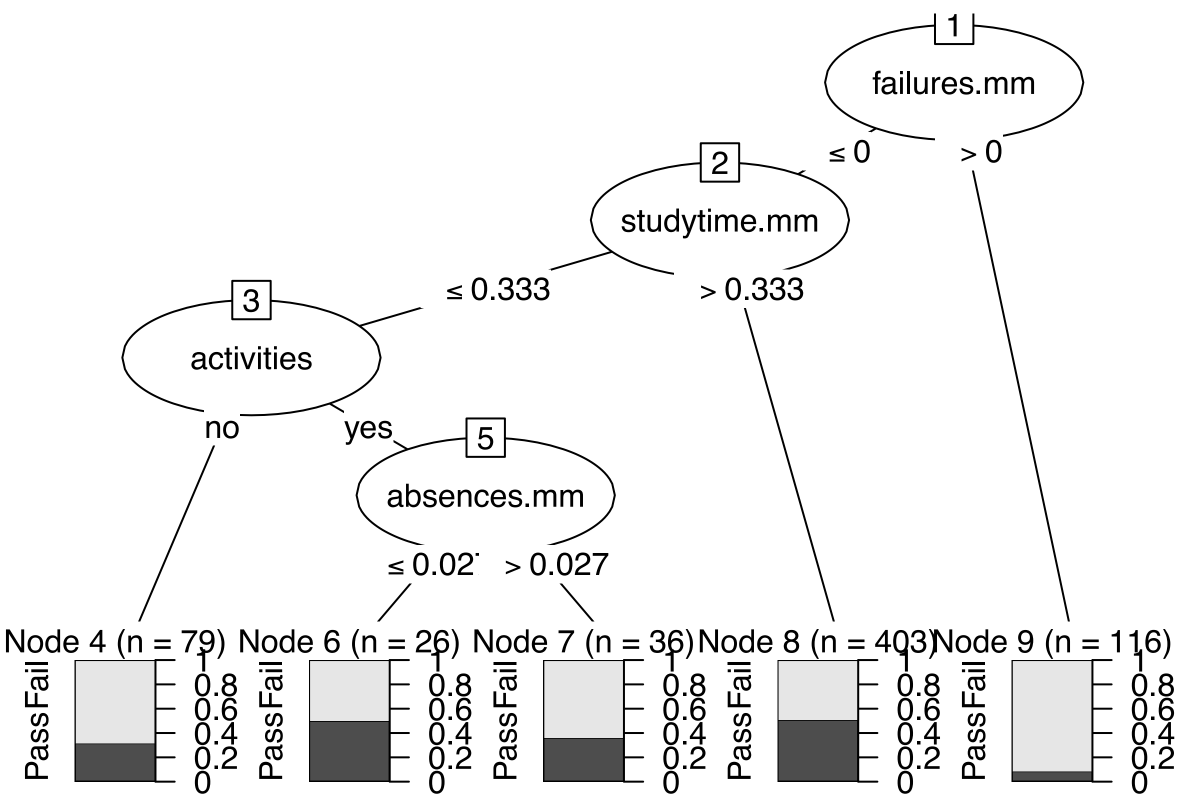
(b) C5.0

```
# C5 model trained by training data set

c5 <- C5.0(formula = G3.pp ~ age.mm + traveltime.mm + studytime.mm + failures.mm + famrel.mm + freetime

# Predictions Test Data Set
Pred_c5 = predict(object = c5, newdata = X)
head(Pred_c5)
```

```
## [1] Fail Fail Fail Fail Fail Fail
## Levels: Fail Pass
```

C5 Visual

```
plot(c5)
```



C5 Evaluation

```
# Evaluation Metrics for C5.0
c5.pred <- table(student_test_class$G3.pp, Pred_c5)
rownames(c5.pred) <- c("Actual: No", "Actual: Yes")
colnames(c5.pred) <- c("Predicted: No", "Predicted: Yes")
addmargins(A = c5.pred, FUN = list(Total=sum), quiet = TRUE)
```

```
##              Pred_c5
##               Predicted: No Predicted: Yes Total
##   Actual: No            183             60   243
##   Actual: Yes            93             48   141
##   Total                 276            108   384
```

C5 Table

```
# Assigning General Form of Table to matrix values for C5.0
TN2 <- c5.pred[1,1]
FN2 <- c5.pred[2,1]
FP2 <- c5.pred[1,2]
TP2 <- c5.pred[2,2]
```

(C) Naives Bayes

```
# Naives Bayes model trained by training data set
nb01 <- naiveBayes(formula = G3.pp ~ age.mm + traveltime.mm + studytime.mm + failures.mm + famrel.mm + 

# Predictions Test data set
Pred_NB <- predict(object = nb01, newdata = X)
head(Pred_NB)
```

```
## [1] Pass Fail Pass Pass Fail Pass
## Levels: Fail Pass
```

NB Evaluation

```
# Evaluation Metrics for Naives Bayes

nb.pred <- table(student_test_class$G3.pp, Pred_NB)
rownames(nb.pred) <- c("Actual: No", "Actual: Yes")
colnames(nb.pred) <- c("Predicted: No", "Predicted: Yes")
addmargins(A = nb.pred, FUN = list(Total=sum), quiet = TRUE)
```

```
##              Pred_NB
##               Predicted: No Predicted: Yes Total
##   Actual: No            115            128   243
##   Actual: Yes            24            117   141
##   Total                 139            245   384
```

NB Table

```
# Assigning General Form of Table to matrix values for NB
TN3 <- nb.pred[1,1]
FN3 <- nb.pred[2,1]
FP3 <- nb.pred[1,2]
TP3 <- nb.pred[2,2]
```

```r
#Baseline Model -
BaselineT <-table(student_test_class$G3.p)
AccN <- BaselineT[1] / (BaselineT[1] + BaselineT[2]) #Accuracy - All Negative model
AccP <- BaselineT[2] / (BaselineT[1] + BaselineT[2]) #Accuracy - All Positive model
cat ("---All Negative Baseline Model----", "\nAccuracy = ", AccN)
```

```
## ---All Negative Baseline Model----
## Accuracy =  0.3671875
```

```r
cat ("\n---All Positive Baseline Model----", "\nAccuracy = ", AccP)
```

```
##
## ---All Positive Baseline Model----
## Accuracy =  0.6328125
```

(A) Accuracy (B) Sensitivity (C) Specificity (D) Error (C) Precision

```r
# Neural Network
Acc0 <- (TN0 + TP0) / (TN0 + FN0 + FP0 + TP0) # Accuracy
Sens0 <- (TP0) / (FN0 + TP0) #Sensitivity
Spec0 <- (TN0) / (TN0 + FP0) # Specificity
Error0 <- 1 - Acc0 #Error Rate
Prec0 <- (TP0) / (FP0 + FP0) #Precision


# Cart Model
Acc1 <- (TN1 + TP1) / (TN1 + FN1 + FP1 + TP1) # Accuracy
Sens1 <- (TP1) / (FN1 + TP1) #Sensitivity
Spec1 <- (TN1) / (TN1 + FP1) # Specificity
Error1 <- 1 - Acc1 #Error Rate
Prec1 <- (TP1) / (FP1 + FP1) #Precision

# C5.0 Model
Acc2 <- (TN2 + TP2) / (TN2 + FN2 + FP2 + TP2) # Accuracy
Sens2 <- (TP2) / (FN2 + TP2) #Sensitivity
Spec2 <- (TN2) / (TN2 + FP2) # Specificity
Error2 <- 1 - Acc2 #Error Rate
Prec2 <- (TP2) / (FP2 + FP2) #Precision

# Naives Bayes
Acc3 <- (TN3 + TP3) / (TN3 + FN3 + FP3 + TP3) # Accuracy
Sens3 <- (TP3) / (FN3 + TP3) #Sensitivity
Spec3 <- (TN3) / (TN3 + FP3) # Specificity
Error3 <- 1 - Acc3 #Error Rate
Prec3 <- (TP3) / (FP3 + FP3) #Precision


cat ("---Neural Network----", "\nAccuracy = ", Acc0, "\nSensitivity = ", Sens0, "\nSpecificty=", Spec0,
```

```
## ---Neural Network----
## Accuracy =  0.625
## Sensitivity =  0.5802469
## Specificty= 0.7021277
## Error Rate 0.375
## Precision 1.678571
```

```r
cat ("\n---Cart Model---", "\nAccuracy = ", Acc1, "\nSensitivity = ", Sens1, "\nSpecificty=", Spec1, "\n
```

```
##
## ---Cart Model---
## Accuracy =  0.3958333
## Sensitivity =  0.308642
## Specificty= 0.5460993
## Error Rate 0.6041667
## Precision 0.5859375
```

```r
cat ("\n---C5.0 Model---", "\nAccuracy = ", Acc2, "\nSensitivity = ", Sens2, "\nSpecificty=", Spec2, "\n
```

```
##
## ---C5.0 Model---
## Accuracy =  0.6015625
## Sensitivity =  0.3404255
## Specificty= 0.7530864
## Error Rate 0.3984375
## Precision 0.4
```

```r
cat ("\n---Naives Bayes---", "\nAccuracy = ", Acc3, "\nSensitivity = ", Sens3, "\nSpecificty=", Spec3,
```

```
##
## ---Naives Bayes---
## Accuracy =  0.6041667
## Sensitivity =  0.8297872
## Specificty= 0.473251
## Error Rate 0.3958333
## Precision 0.4570312
```