Contents:

Introduction:

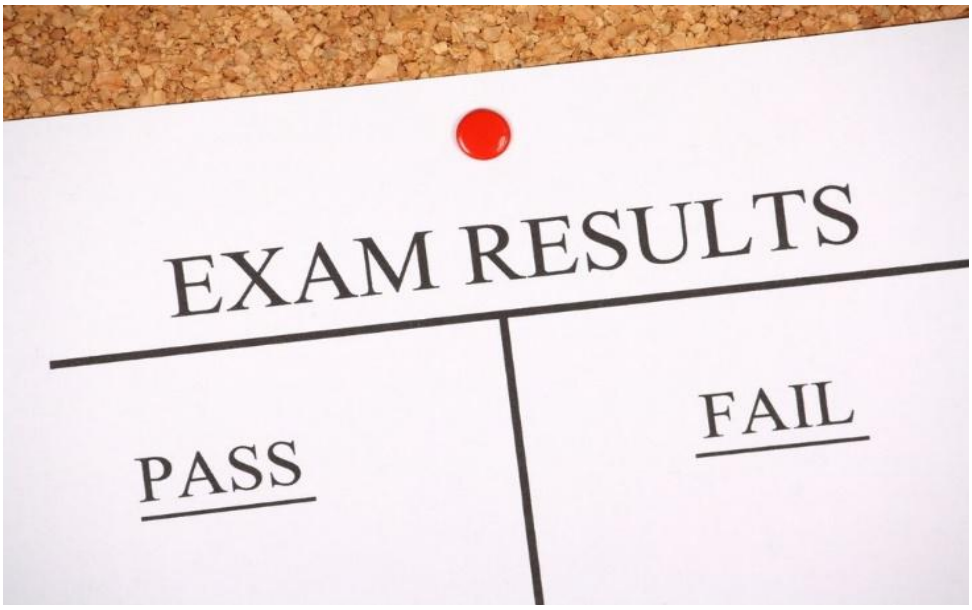
Language used: R Language

Software: R Studio

Data Set : UCI Repository

Model Techniques Used:

1) Logistic Regression
2) Decision Tree using C50 and CART
3) Random Forest
4) SVM (Linear and Radial Kernel)



Problem Statement :

To develop a model which predicts whether student who are pursuing their secondary education in 2 different schools Pass or Fail in their examination.

Group Members:

1) Anirudh R N

2) Kiran C M

Data Set Attribute Information

This data contains student achievement in secondary education of two schools. The data attributes include student grades, demographic, social and school related features, etc) and it was collected by using school reports and questionnaires. Datasets are provided regarding the performance in Mathematics (mat) and Portuguese(por).

Attribute Information:

Number of Attributes: 32

Attributes for student-mat.csv & student-por.csv(Math & Portuguese Course)

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 (numeric: 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 (numeric: 0 - none, 1 - primary education (4th grade), 2 – (5th to 9th grade), 3 - secondary education or 4 - higher education)
8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – (5th to 9th grade), 3 - secondary education or 4 - higher education)
9 Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10 Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11 reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12 guardian - student's guardian (nominal: 'mother', 'father' or 'other')
13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15 failures - number of past class failures (numeric: 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 other subjects) (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 (numeric: from 1 - very bad to 5 - excellent)
25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)
26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)
27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29 health - current health status (numeric: from 1 - very bad to 5 - very good)
30 absences - number of school absences (numeric: from 0 to 93)

these grades are related with the course subject, Math

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)

```
> str(rawDataSet)
'data.frame':   1044 obs. of  34 variables:
 $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
 $ sex       : Factor w/ 2 levels "F","M": 1 1 1 1 1 2 2 1 2 2 ...
 $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
 $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 2 2 2 2 2 ...
 $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 2 1 1 2 2 1 2 1 ...
 $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
 $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
 $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
 $ Mjob      : Factor w/ 5 levels "at_home","health",..: 1 1 1 2 3 4 3 3 4 3 ...
 $ Fjob      : Factor w/ 5 levels "at_home","health",..: 5 3 3 4 3 3 3 5 3 3 ...
 $ reason    : Factor w/ 4 levels "course","home",..: 1 1 3 2 2 4 2 2 2 2 ...
 $ guardian  : Factor w/ 3 levels "father","mother",..: 2 1 2 2 1 2 2 2 2 2 ...
 $ traveltime: 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 : Factor w/ 2 levels "no","yes": 2 1 2 1 1 1 1 2 1 1 ...
 $ famsup    : Factor w/ 2 levels "no","yes": 1 2 1 2 2 2 1 2 2 2 ...
 $ paid      : Factor w/ 2 levels "no","yes": 1 1 2 2 2 2 1 1 2 2 ...
 $ activities: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 1 2 ...
 $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
 $ higher    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ internet  : Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 2 ...
 $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
 $ famrel    : 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 ...
 $ grade     : Factor w/ 2 levels "FAIL","PASS": 1 1 1 2 1 2 2 1 2 2 ...
>
```

Show 10 entries   Search:

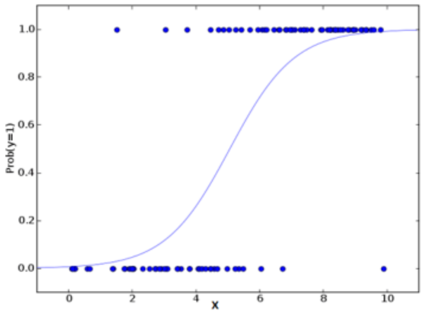
	school	sex	age	Pstatus	Medu	Fedu	Mjob	Fjob	guardian	traveltime
1	GP	F	18	A	3.35	4	at_home	teacher	mother	1.8
2	GP	F	17	T	1	1	at_home	other	father	1
3	GP	F	15	T	1	1	at_home	other	mother	1
4	GP	F	15	T	4	2	health	services	mother	1
5	GP	F	16	T	3	2.542	other	other	father	1
6	GP	M	16	T	4	3	services	other	mother	1

Showing 1 to 6 of 6 entries   Previous 1 Next

Model Selection

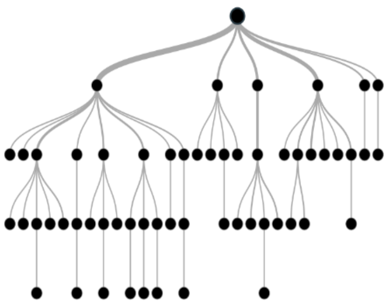
1) Logistic Regression:

Logistic Regression is a classification supervised algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. To represent binary / categorical outcome, we use dummy variables. You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical



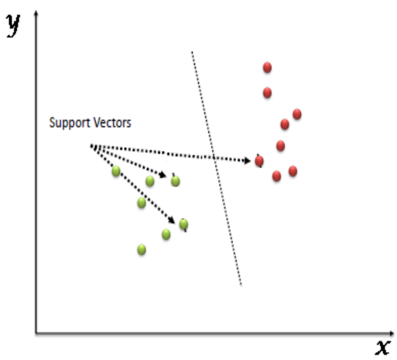
2) Decision Tree:

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables



3) Support Vector Machine (SVM)

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well



Feature Engineering:

In machine learning, feature engineering is the process of selecting or creating features (variables) in a data set to improve machine learning results The first use of feature engineering used here is the selection of the relevant variables.

1) Removing the Column Address,famsize,reason,nursery,Walc in the Data Set, since it does not provide relevant information to predict the result of the student.

2) We will average the previous result(G1+G2) as Previous Grade results(PrevResult)

3) We will introduce a new factor as “finalGrade” to predict whether the student would pass/fail. This variable would be defined from G3.

```
studentDataset<-subset(studentDataset,select=-c(address,famsize,reason,
                                  nursery,Walc,G1,G2))
colnames(studentDataset)[colnames(studentDataset)=="G3"] <- "finalScore"

View(studentDataset)
studentDataset$FinalGrade<-NULL
studentDataset$FinalGrade<-
factor(ifelse(studentDataset$finalScore>=median(studentDataset$finalScore),"PASS","FAIL"))
```

```
$ finalScore: num  6 6 10 15.2 10 ...
$ prevScore : num  5.5 0 0 14.5 8 15 12 5.5 17 14.5 ...
$ FinalGrade: Factor w/ 2 levels "FAIL","PASS": 1 1 1 2 1 2 2 1 2 2 ...
```

Sample Model Development using Logistic Regression.

Sample Code and Analysis:

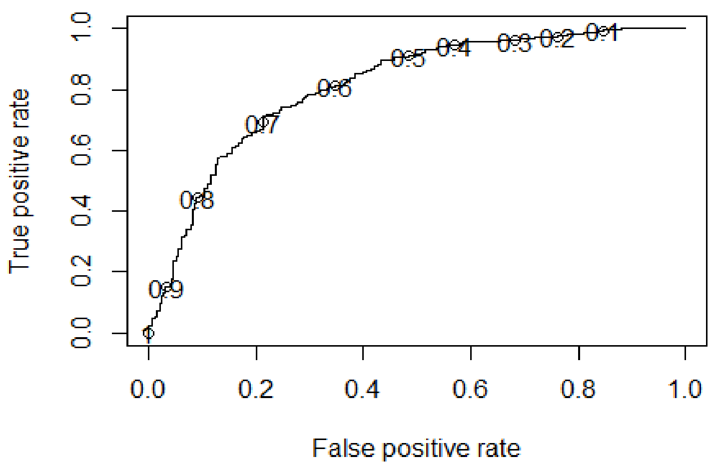
#Model Development using Train Data
LogisticModel_1 <- glm(FinalGrade ~ . -FinalGrade,studentDataset_train[-c(27,26)],family = "binomial")

#Predict Our Model using Test Data
LogisticPredict_1 <- predict(LogisticModel_1,studentDataset_test[-c(27,26)],type = "response")

#Cross Matrix to predict the Actual/Predicted TRUE and FALSE
table(Actualvalue = studentDataset_test$FinalGrade,PredictedValue=LogisticPredict_1 >0.5)

#Plotting ROCR to find the threshold of best true positive rate and negative false rate
library(ROCR)
ROCRPrediction <- prediction(predict_train,studentDataset_train$FinalGrade)
ROCRPerformance <- performance(ROCRPrediction,"tpr","fpr")
plot(ROCRPerformance,print.cutoffs.at=seq(0.1,by=0.1))

From the below ROCR plot, we have considered 0.4 threshold which had less false positive rate and more accuracy.




```
Call:

glm(formula = grade ~ . - grade, family = "binomial", data = 
rawDataSet_subset[-c(28,
    27, 26)])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3645  -0.8307   0.5267   0.8006   3.0152

    Null deviance: 1372.4  on 1043  degrees of freedom
Residual deviance: 1066.6  on 1009  degrees of freedom
AIC: 1136.6

Number of Fisher Scoring iterations: 5
```

Sample Model Development using Decision Tree

Sample Code and Analysis:

Part-1:C50

#Model Development using Train Data

C50model<-C5.0(studentDataset_train[-c(26,27,28)],studentDataset_train$FinalGrade)

#Predict Our Model using Test Data

C50predict<-predict(C50model,studentDataset_test[-c(26,27,28)])

#Cross Matrix to predict the Actual/Predicted TRUE and FALSE

CrossTable(C50predict,studentDataset_test$FinalGrade)

#Accuracy of the model

sum(c50predict == studentDataset_test$FinalGrade) / length(studentDataset_test$FinalGrade)

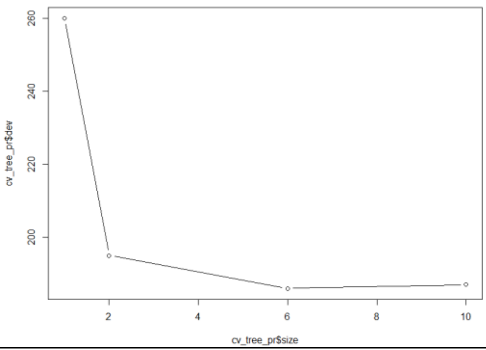
#Tune the Model using trials method(boost the iterations)

C50model_tuned<-C5.0(………,trials=10)

#Prune the Model(reducing the Tree Size from 25 to 10) to get better Accuracy

prune_model_1 <- prune.misclass(tree_model_1,best = 10)

cv_tree_pr <- cv.tree(prune_model_1, FUN = prune.misclass)




```
Call:
C5.0.default(x = studentDataset_train[-c(26, 27, 28)], y =
studentDataset_train$FinalGrade)

Classification Tree
Number of samples: 700
Number of predictors: 25

Tree size: 41
Evaluation on training data (700 cases):

           Decision Tree
         ----------------
         Size      Errors

           56    74(10.6%)   <<
```

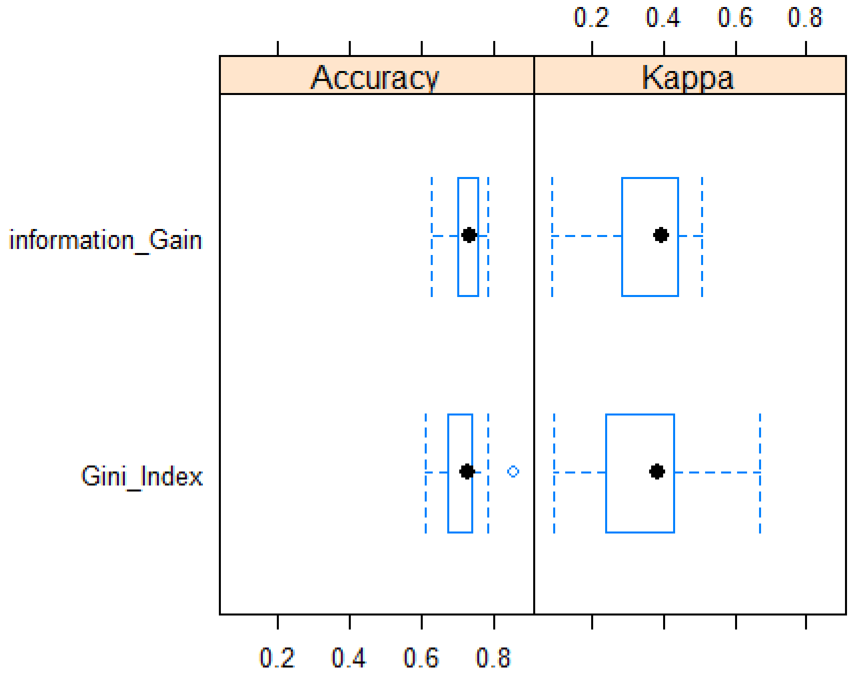
Part 2: CART

#Model Development using Train Data-[information gain as criteria]
cartModelIgain<- train(studentDataset_train[-c(26,27,28)],studentDataset_train$FinalGrade,trControl = trctrl,method = "rpart",parms = list(split = "**information**"))

#Model Development using Train Data-[gini index as criteria]
cartModelGini <- train(studentDataset_train[-c(26,27,28)],studentDataset_train$FinalGrade,method = "rpart",parms = list(split = "**gini**"),trControl=trctrl,tuneLength=10)

#Predict Our Model using Test Data
cartIgainPrediction <- predict(cartModelIgain,studentDataset_test)
cartGiniPrediction <- predict(cartModelGini, newdata = studentDataset_test)

#Confusion Matrix to predict the Actual/Predicted TRUE and FALSE
confusionMatrix(cartGiniPrediction,studentDataset_test$FinalGrade)



CART

```
700 samples
 25 predictor
  2 classes: 'FAIL', 'PASS'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 630, 630, 630, 630, 630, 630, ...
```

Sample Model Development using SVM

Sample Code and Analysis:

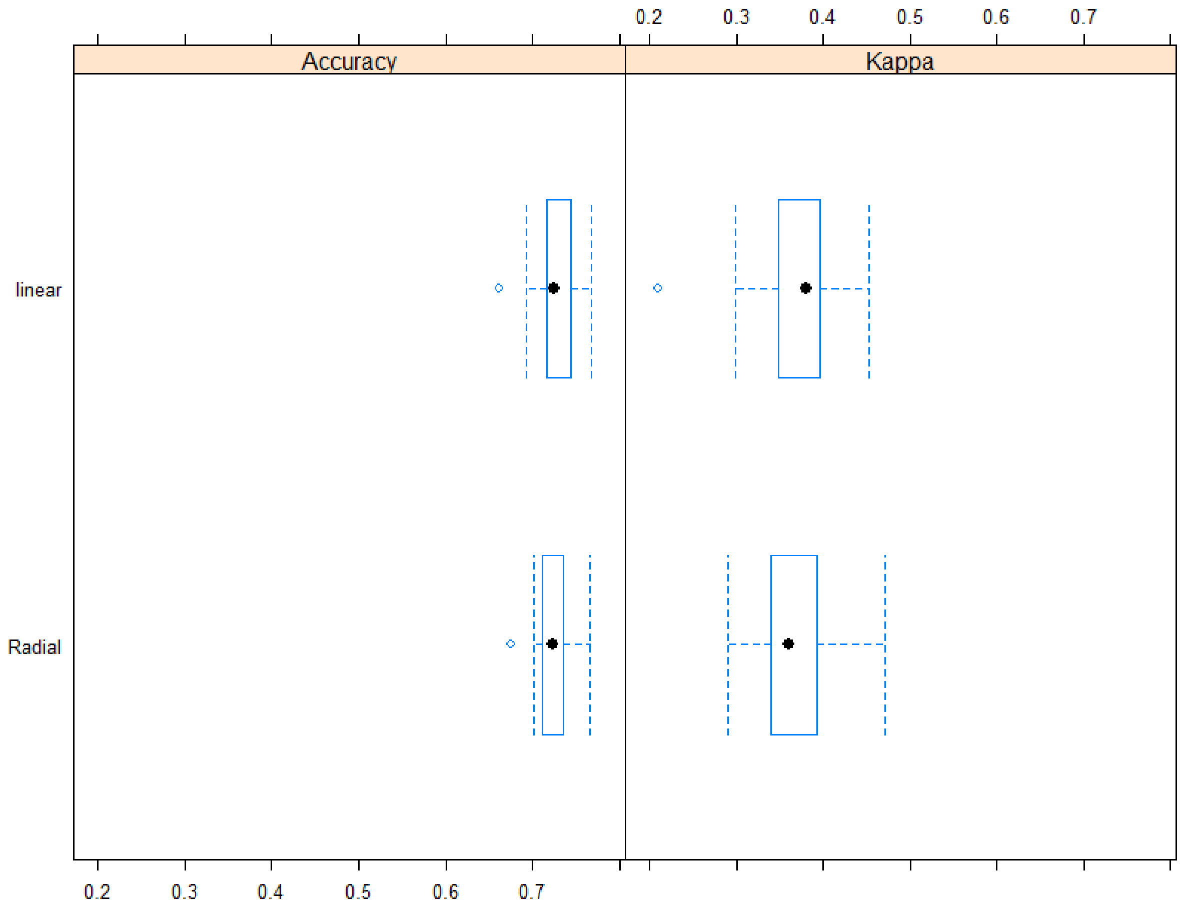
#Model Development using Train Data-[SVM Linear]
```
cartModelIgain<- train(studentDataset_train[-svmLinear_grade <- train(FinalGrade ~. -
FinalGrade,studentDataset_train[-c(26,27)],method = "svmLinear",tuneGrid = grid,trainControl =
ctrl)
```

#Model Development using Train Data-[SVM Radial]
```
svmRadialKernel_grade <- train(FinalGrade ~. - FinalGrade,studentDataset_train[-c(26,27)],method
="svmRadial",tuneGrid = grid_radial,trainControl= ctrl)
```

#Predict Our Model using Test Data
```
svmLinearPrediction_grade <- predict(svmLinear_grade,studentDataset_test[-c(26,27)])
svmRadialPrediction_grade <- predict(svmRadialKernel_grade,studentDataset_test[-c(26,27)])
```

#Confusion Matrix to predict the Actual/Predicted TRUE and FALSE
```
confusionMatrix(svmLinearPrediction_grade,studentDataset_test$FinalGrade)
```

#Plot for the comparision of SVM Linear and Radial



Attribute Selection

Attribute Selection:

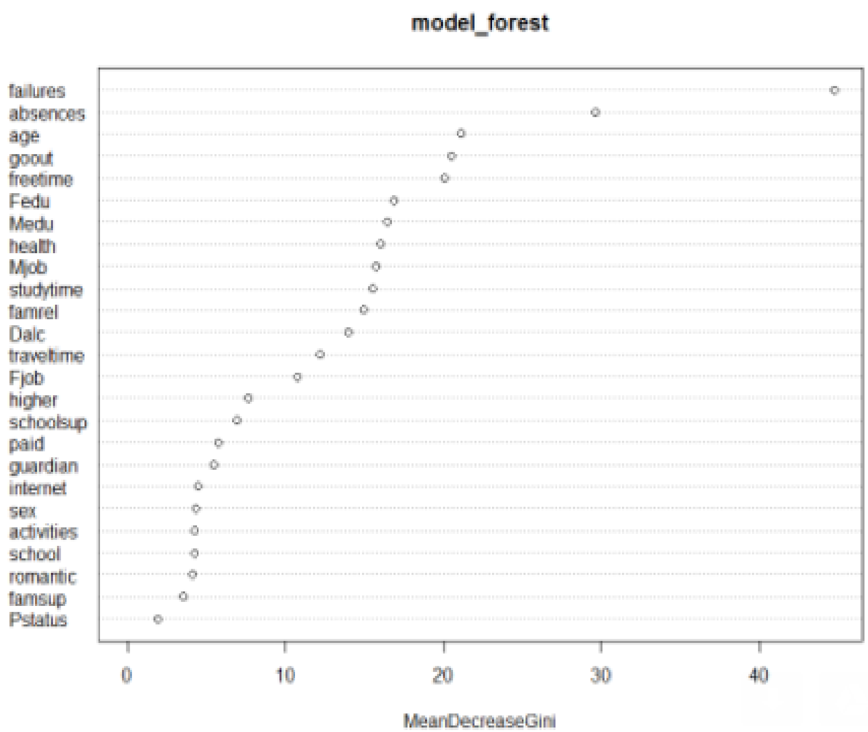
1) **Information Gain**
   It is used to select the best combination of attributes required for best model accuracy. The Information gain function is used here to know the best combination of attributes found in the data set.

   With entropy defined as : **H = − pk log2 pk K i=1**
   Then the change in entropy, or Information Gain, is defined as:
   ΔH = H − mL m HL − mR m HR
   where m is the total number of instances, with mk instances belonging to class k, where K = 1, … ,

```
library(FSelector)
best_feature <- information.gain(FinalGrade ~ .,studentDataSet)
print(best_feature)
cutoff.biggest.diff(best_feature)
```

2) **Gini index**
   It is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with lower Gini index should be preferred.The Gini index is defined as:
   Gini = 1 − pk 2 K i=1
   where pk denotes the proportion of instances belonging to class k K = 1, … , k .)

   importance(features)



Evaluation Method

Background of Model Development

The first step is to predict the model by including the Previous Exam Results. In this case, a total of X outcomes out of n has incorrectly classified and total of Y outcomes out of n was correctly classified.

In order to evaluate the effectiveness of a prediction model, predicted values must be compared

	Predicted as True	**Predicted as False**
Actually True	True Positive(TP)	False Negative(FN)
Actually False	False Positive(FP)	True Negative(TN)

Possible Prediction Result

Accuracy = (TP + TN)/(TP+TN+FP+FN)

Along with the accuracy we will calculate Precision and Recall Percentage as well

1) Precision = TP/(TP+FP)
2) Recall = TP/(TP+FN)

Precision and recall are used together to make a better evaluation. The main idea is that accurately predicting positive outcome is not enough. A good predictive model must have a good combination of successful positive predictions and successful negative predictions

Implementation and Results

1) Logistic Regression Model

Let us consider the 2 cases for logistic Regression

Case 1 : Basic Logistic Regression Algorithm

Accuracy of the Model	72%
Key Notes: Previous Results marks are Excluded during the Model Development Threshold is used as 0.5 as a hit and trial method for predicting the actual Pass/Fail.	

	Predicted Passed	Predicted Fail
Actual Passed	89	43
Actual Fail	53	159

Table 1. Confusion matrix for Basic Logistic Algorithm Excluding the Previous Results

```
LogisticModel_1 <- glm(FinalGrade ~ . -FinalGrade,studentDataset_train[-c(27,26)],family = "binomial")
LogisticPredict_1 <- predict(LogisticModel_1,studentDataset_test[-c(27,26)],type = "response")
table(Actualvalue = studentDataset_test$FinalGrade,PredictedValue=LogisticPredict_1 >0.5)
```

Case 2: Logistic Regression Model by using the threshold value obtained by ROCR

Accuracy of the Model	75%
Key Notes: Previous Results marks are Excluded during the Model Development. Threshold is used as 0.4,since we have obtained from ROCR Curve Implementation	

	Predicted Passed	Predicted Fail
Actual Passed	79	53
Actual Fail	30	182

Table 2. Confusion matrix for Logistic Algorithm Excluding the Previous Results and using threshold value obtained from ROCR curve

```
library(ROCR)
ROCRPrediction <- prediction(predict_train,studentDataset_train$FinalGrade)
ROCRPerformance <- performance(ROCRPrediction,"tpr","fpr")
plot(ROCRPerformance,print.cutoffs.at=seq(0.1,by=0.1))
```

Inference:

We have used the backward Elimination technique to remove the unwanted features and analysed the ROCR curve for predicting the Actual/Predicted Pass &Fail correctly and for good accuracy(i.e false negative rate should be as minimum as possible).

2) Decision Tree Model

Let us consider the 3 cases for Decision Tree Model using C50

Case 1 : Basic C50 Algorithm

Accuracy of the Model	67%
Tree Size	56
Number of Samples	700
Number of Predictors	25
Key Notes: Previous Results marks are Excluded during the Model Development	

	Predicted Passed	Predicted Fail
Actual Passed	102	82
Actual Fail	29	131

Table 3. Confusion matrix for Basic C50 Algorithm Excluding the Previous Results

```
library(gmodels)
C50model<-C5.0(studentDataset_train[-c(26,27,28)],studentDataset_train$FinalGrade)
C50predict<-predict(C50model,studentDataset_test[-c(26,27,28)])
CrossTable(C50predict,studentDataset_test$FinalGrade)
sum(c50predict == studentDataset_test$FinalGrade) / length(studentDataset_test$FinalGrade)
```

Case 2 : Tuning C50 Algorithm

Accuracy of the Model	72%
Average Tree Size	50.9
Number of Samples	700
Number of Predictors	25
Key Notes: 1) Previous Results marks are Excluded during the Model Development. 2) Number of trials(trials = 10) are utilized to improve the performance of the model.	

	Predicted Passed	Predicted Fail
Actual Passed	88	51
Actual Fail	43	162

Table 4. Confusion matrix for Basic C50 Algorithm with tuning Excluding the Previous Results

```
C50model_tuned<-C5.0(studentDataset_train[-
c(26,27,28)],studentDataset_train$FinalGrade,trials=10)
C50predict_tuned<-predict(C50model_tuned,studentDataset_test[-c(26,27,28)])
CrossTable(c50predict_tuned,studentDataset_test$FinalGrade)
```

Case3: Pruning C50 Algorithm

Accuracy of the Model	69%
Tree Size	25 before Pruning and 10 after pruning
Key Notes: 1) Previous Results marks are Excluded during the Model Development. 2) Pruning method is used here to enhance the Model Accuracy	

	Predicted Passed	Predicted Fail
Actual Passed	122	54
Actual Fail	53	115

Table 5. Confusion matrix for Basic C50 Algorithm with pruning Excluding the Previous Results

Prune the Model(reducing the Tree Size from 25 to 10) to get better Accuracy

```
prune_model_1 <- prune.misclass(tree_model_1,best = 10)
cv_tree_pr <- cv.tree(prune_model_1, FUN = prune.misclass)
```

Inference:

For Decision Tree Model using C50,
we have to tuned the model and pruned the model (i.e removing the sub-tree) to enhance our model accuracy.

2) Decision Tree Implementation using CART

Let us consider the 2 cases for Decision Tree Model using CART

Case1: Splitting Criteria as "information gain"

Accuracy of the Model	73.84
Sample Sizes	630, 630, 630, 630, 630, 630, ...
Number of Predictors	25
Sample	700
Key Notes: 1) Previous Results marks are excluded during the Model Development. 2) Model Accuracy when the splitting criteria= information gain	

	Predicted Passed	Predicted Fail
Actual Passed	77	35
Actual Fail	55	177

Table 6. Confusion matrix for CART Algorithm with "information gain as splitting criteria" Excluding the Previous Results

```
library(caret)
trctrl <- trainControl(method = "repeatedcv",number = 10,repeats = 3)

#Syntax: train(target_variable,data_set,...)
cartModelIgain<- train(studentDataset_train[-
c(26,27,28)],studentDataset_train$FinalGrade,trControl = trctrl,method = "rpart",parms = list(split =
"information"))

#Predict
cartIgainPrediction <- predict(cartModelIgain,studentDataset_test)

#Calculating Accuracy using cofusion Marix method
confusionMatrix(cartIgainPrediction,studentDataset_test$FinalGrade)
```

Case2: Splitting Criteria as "gini index"

Accuracy of the Model	74.13%
Sample Sizes	700
Number of Predictors	25
Key Notes: 1) Previous Results marks are excluded during the Model Development. 2) Model Accuracy when the splitting criteria= gini index	

	Predicted Passed	Predicted Fail
Actual Passed	83	40
Actual Fail	49	172

Table 7. Confusion matrix for CART Algorithm with "gini index as splitting criteria" Excluding the Previous Results

```
#Training the Decision Tree classifier with criterion as gini index

cartModelGini <- train(studentDataset_train[-c(26,27,28)],studentDataset_train$FinalGrade,method
= "rpart",parms = list(split = "gini"),trControl=trctrl,tuneLength=10)

cartGiniPrediction <- predict(cartModelGini, newdata = studentDataset_test)
confusionMatrix(cartGiniPrediction,studentDataset_test$FinalGrade)
```

Inference :

For Decision Tree Model using CART,
splitting criteria as "gini index" is giving more accuracy than "information gain"

3)RandomForest

Basic Random Forest Algorithm

Accuracy of the Model	77%
Key Notes: 1) Previous Results marks are excluded during the Model Development. 2) Model Accuracy during normal build	

	Predicted Passed	Predicted Fail
Actual Passed	103	49
Actual Fail	29	163

Table 8. Confusion matrix for Random Forest Algorithm Excluding the Previous Results

```
bestmtry <- tuneRF(studentDataset_train[-c(26,27,28)],studentDataset_train$FinalGrade,stepFactor =
1.2,improve = 0.01,trace = T,plot = T)
model_forest <- randomForest(FinalGrade ~ . -FinalGrade, data = studentDataset_train[-c(26,27)],mtry =
6,ntree = 500 )
predict <- predict(model_forest,studentDataset_test[-c(26,27,28)])
confusionMatrix(predict,studentDataset_test$FinalGrade)
importance(model_forest)
varImpPlot(model_forest)
varImpPlot(model_forest)
```

3)SVM Linear and SVM Radial

Case1: SVM Linear Model

Accuracy of the Model	77.6%
Number of samples	700
Number of predictors	25
Key Notes: 1) Previous Results marks are excluded during the Model Development. 2) Model Accuracy during SVM Linear	

	Predicted Passed	Predicted Fail
Actual Passed	71	17
Actual Fail	60	196

Table 10. Confusion matrix for SVM linear Excluding the Previous Results

```
svmLinearPrediction_grade <- predict(svmLinear_grade,studentDataset_test[-c(26,27)])
summary(svmLinearPrediction_grade)

#Accuracy using confusion matrix
#confusionMatrix(prediction object,targetvariable of test Data)
confusionMatrix(svmLinearPrediction_grade,studentDataset_test$FinalGrade)
```

Case2: SVM Radial

Accuracy of the Model	76.45%
Number of Samples	700
Number of Predictors	25
Key Notes: 1) Previous Results marks are excluded during the Model Development. 2) Model Accuracy during SVM Radial	

	Predicted Passed	Predicted Fail
Actual Passed	72	22
Actual Fail	59	191

Table 11. Confusion matrix for SVM Radial Excluding the Previous Results

```
grid_radial <- expand.grid(sigma = c(0.01,0,015,0.2),C= c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5,
1.75, 2,5))
svmRadialKernel_grade <- train(FinalGrade ~. - FinalGrade,studentDataset_train[-c(26,27)],method
="svmRadial",tuneGrid = grid_radial,trainControl= ctrl)
svmRadialPrediction_grade <- predict(svmRadialKernel_grade,studentDataset_test[-c(26,27)])
confusionMatrix(svmRadialPrediction_grade,studentDataset_test$FinalGrade)
```

Evaluation between the Models.

Let us Consider the Accuracy of all the cases:

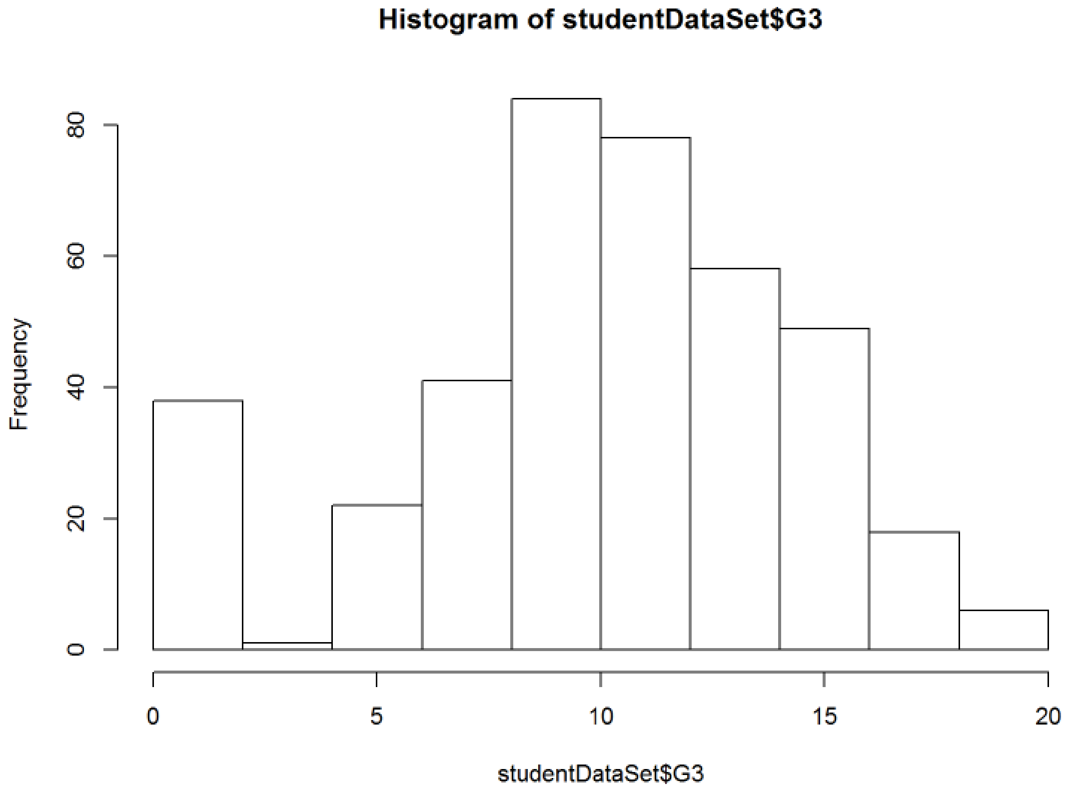
Model Selection with scenario based	Accuracy	Additional Notes
Basic Logistic Regression Model	72%	Featured Engineering is applied
Enhanced Logistic Regression Model	75%	ROCR Curve is utilized to know the threshold and to enhance the model
Basic Decision Tree Model using C50	67%	Featured Engineering is applied
Tuned Decision Tree Model using C50	72%	"Trials" method is used to boost the model,by repetitive iteration of the mode for tree construction
Pruned Decision Tree Model using C50	69%	"sub-tree" are removed by plotting the cv.tree method and using prune method
Random Forest	77%	Feature Engineering is applied here.
Decision Tree Model using CART(information gain as splitting criteria)	73%	Information gain is used as criteria to split the nodes
Decision Tree Model using CART(gini index as splitting criteria)	74%	Gini index is used as criteria to split the nodes
SVM Linear Model Kernel	77%	Linear Kernel
SVM Radial Model Kernel	76%	Radial kernel

Note: We have mentioned the Accuracy which had the best Result for the respective Algorithm

Model Selection	Accuracy	Precision	Recall
Logistic Regression	75%	75%	78%
Decision Tree	74%	76%	83%
Random Forest	77%	76%	84%
SVM	77%	71%	83%

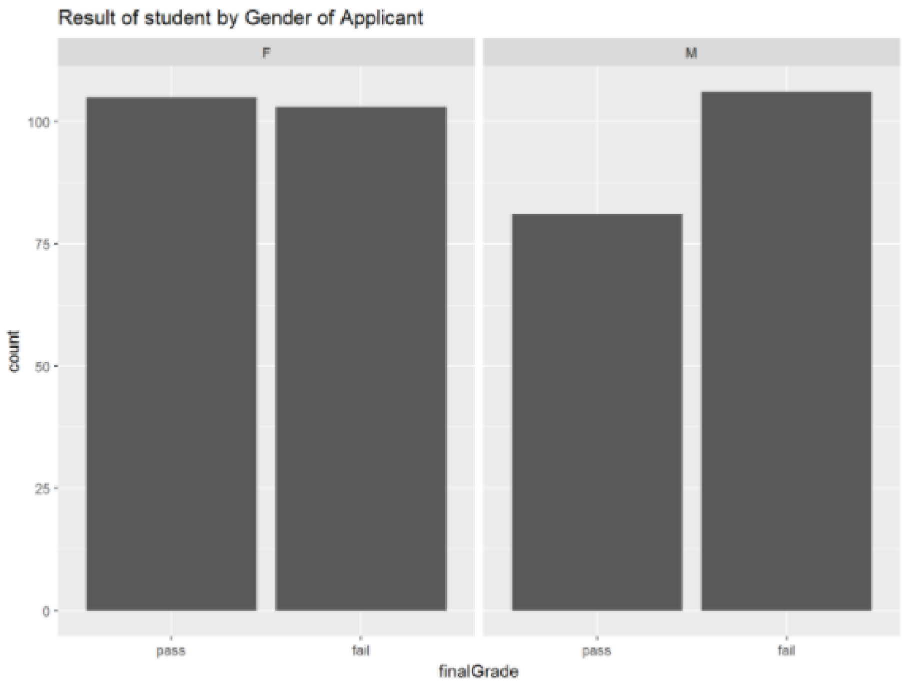
Table 12. Method comparison for the data set excluding the Previous Exam Results

Plots:

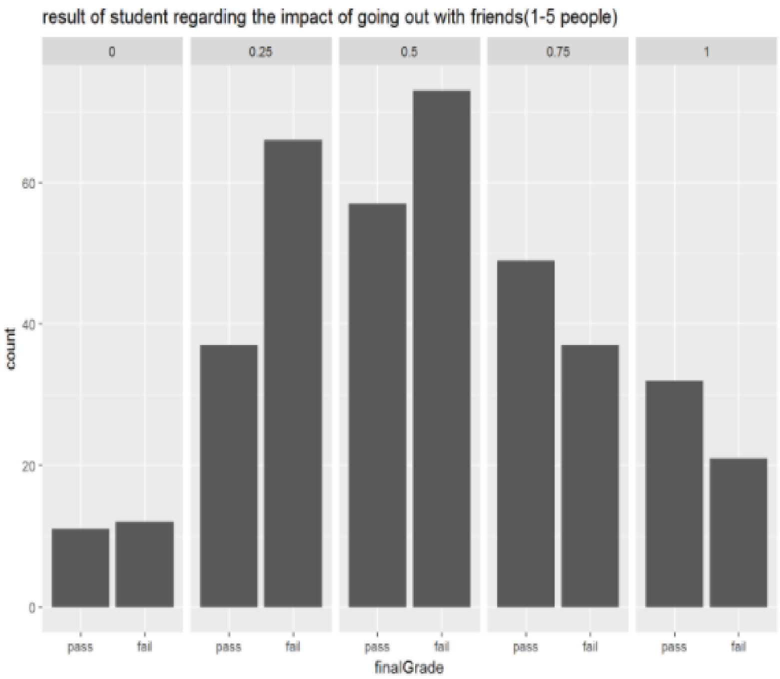



```
print(ggplot(studentDataSet, aes(x=finalGrade))+geom_bar()+facet_grid(.~sex)+ggtitle("Result of student by Gender of Applicant"))
```

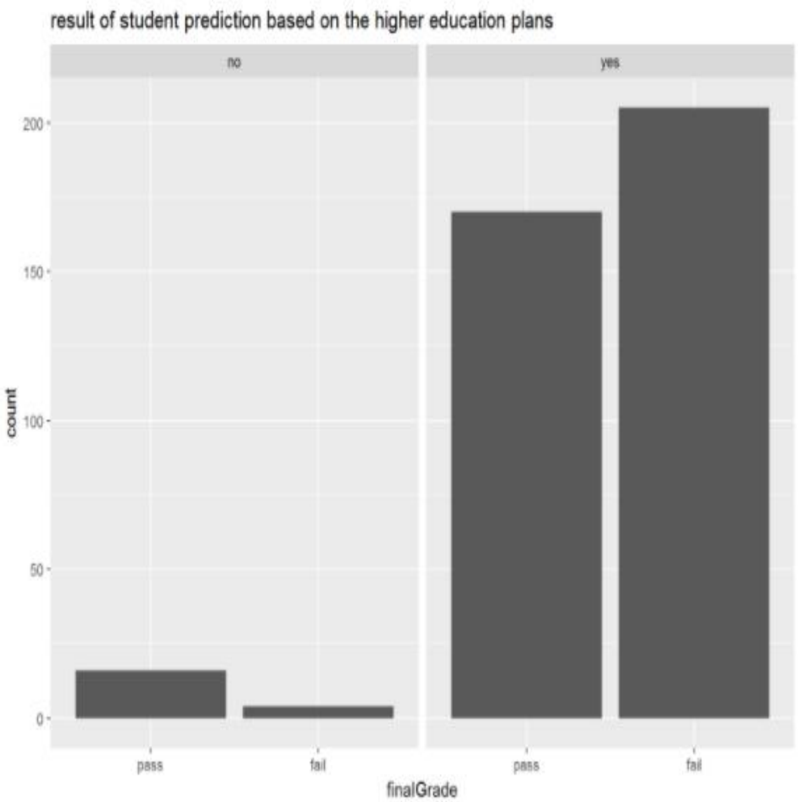
```
## Warning in plyr::split_indices(scale_id, n): '.Random.seed' is not an
## integer vector but of type 'NULL', so ignored
```

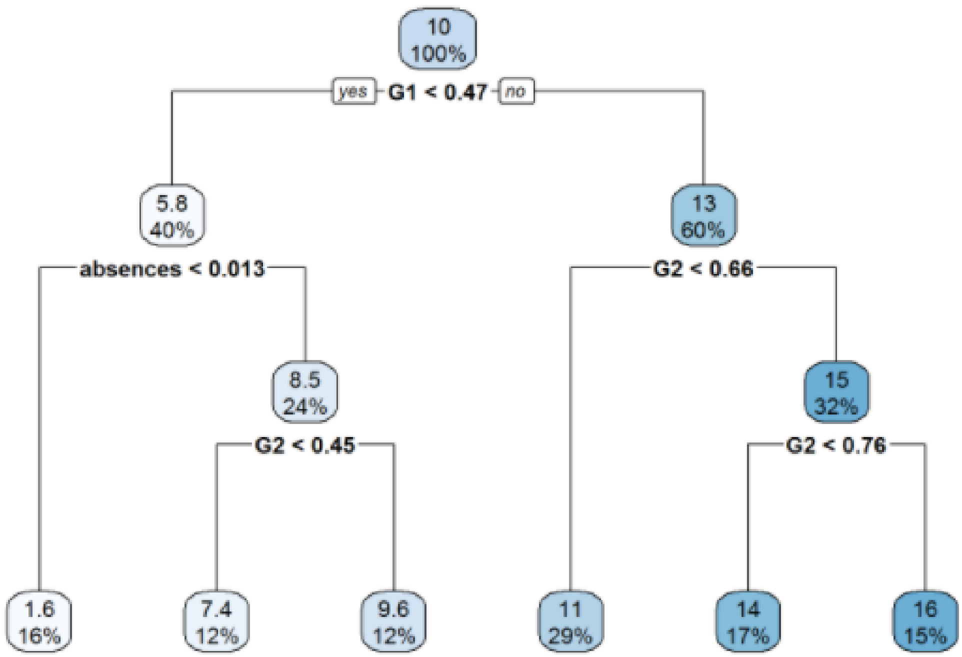


```
print(ggplot(studentDataSet,aes(x=finalGrade)) + geom_bar()+facet_grid(.~goout)+ggtitle("result of student regarding the impact of going out with friends(1-5 people)"))
```




```
print(ggplot(studentDataSet,aes(x=finalGrade)) + geom_bar()+facet_grid(.~higher)+ggtitle("result of student prediction based on the higher education plans"))
```






```
rpart.plot(tree.model_plot_extra,type = 3,digits = 3,fallen.leaves = T)
```