Predictive Analytics on Student Performance

Amrita Jose, Asmita Singh
Volgenau School of Engineering, George Mason University

Authors' Note

This paper was prepared for STAT 515 – Visualization for Analytics, taught by Professor Richard S. Sigman. Correspondence concerning this paper should be addressed to Amrita Jose or Asmita Singh, Department of Data Analytics Engineering, Volgenau School of Engineering, George Mason University, Fairfax, VA 22030
Contact: ajose4@masonlive.gmu.edu, asingh62@masonlive.gmu.edu

Abstract

A research journal on 'Factors Affecting Student Performance', published by the Global Journals, one of the leading research publications in the world, claimed that the primary factors affecting the academic performance of students in colleges include several socio-economic causes such as the education of the parents, teacher-student ratio, distance of colleges, family income and student attendance among few others. Similar studies have been conducted on the performance of college students and the factors contributing to it. The research continues to determine the best predictors of academic performance in order to take appropriate measures that will ensure a healthy environment for the students to grow and learn. This paper discusses four predictive analysis methods to determine the best predictors of the academic performance of students in secondary education of two Portuguese schools. The datasets were saved as CSV files, and the analysis was done using R programming with the designs based on the general guidelines for compelling visualizations.

Keywords: Predictors of academic performance, socio-economic causes, teacher-student ratio

Predictive Analytics on Student Performance

Academic performance determines the probability of higher academic and career success of the students. Hence, it is essential to capture the determinants of student performance in order to understand the challenges faced by them and also ensure that the students are surrounded by an environment that will help them to progress and excel.

Source and Dataset Description

Predictive Analysis is the process of using various data analysis methods to predict or forecast data outcomes based on current or historical data. This analysis helps to forecast trends and generate insights with significant precision. This paper covers four such predictive analysis methods that determine the best factors affecting the performance of students in the secondary education of two Portuguese schools. The dataset for the project was obtained from the UCI Machine Learning Repository and described attributes pertaining to the performance of students in two subjects – Math and Portuguese. There are 33 such data attributes, which include student grades, demographic, personal, social and school-related features. The data for both subjects were collected using school reports and questionnaires. The combined dataset of both subjects has 1044 records. The unit of analysis is each student of the class as the dataset describes the performance of each student.

Data Transformation

The student performance data for the two different subjects – Math and Portuguese were in two separate CSV files which were merged using R code to perform all the relevant data analysis. Also, the two datasets did not have any separate data field to identify what subject the data represented. Hence, a field indicating the respective subject was added to each file before the merge.

The dataset had no missing values or inconsistencies and hence did not require data tidying. The presence of any missing values was validated using R code.

Research Questions

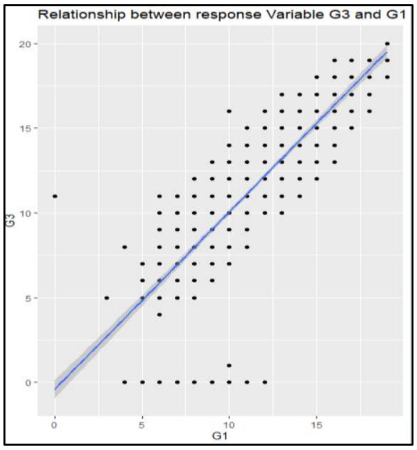
The research questions formulated for the project based on the dataset areas listed below: -

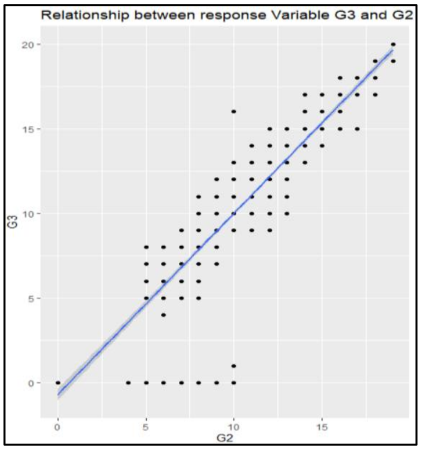
1. What is the effect of addictions, such as the Internet and alcohol on students' academic performance?
2. Can demographic and social variables influence the grade of the student?
3. Do factors such as parental employment and parental education affect student grades?
4. Does high travel time to school and low study time allocation affect student's performance?
5. Do past academic failures affect the grades of students?
6. Does extra educational support improve the overall grade?
7. How can extracurricular activities impact the final grade?
8. Does an ambitious student have better chances of getting good grades?
9. Does poor health status or high absenteeism contribute to bad grades/performance?
10. Define the correlation between the grades - G3, G2, and G1.

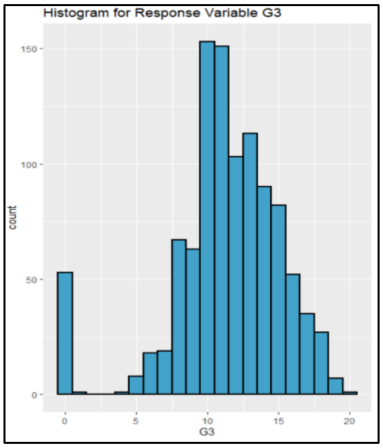
Exploratory Data Analysis

The four data analysis methods used to predict the performance, or the final grade of the students are – Linear Regression, Logistic Regression, Classification Tree and Random Forest for a continuous variable (also known as Regression Forest). All four predictive analysis models were developed using R Programming. In order to avoid the issue of overfitting, all four analysis methods have been trained first and then predicted on a test set. The response variable in the models is the overall grade of the student, named 'G3'. The predictor variables include the personal, social, demographic and school-related variables in the dataset.

On plotting the relationships between the grades, it was observed that there is high collinearity between G3, G1 and G3, G2. This can be seen in the screenshot below: -



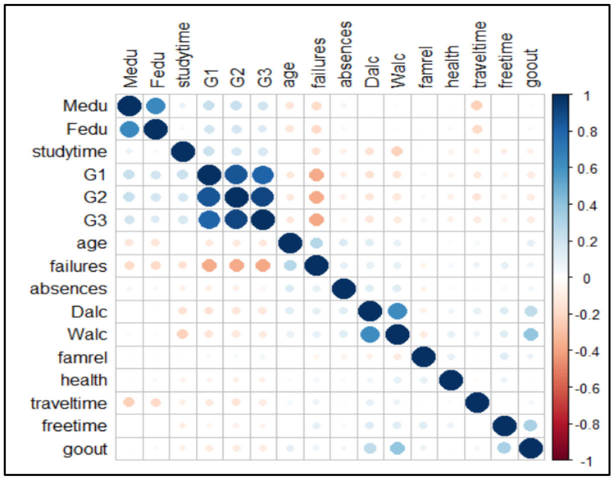


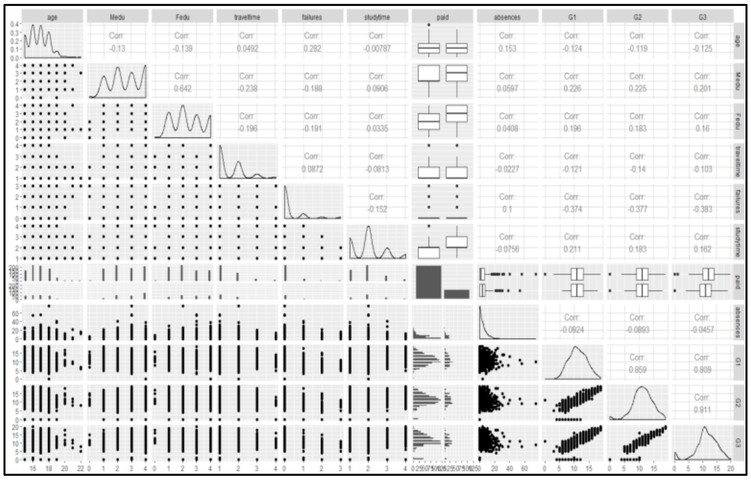


Also, the histogram for G3 shows that there a few students who scored zero, and the median score of the class is 10. Histograms for number of failures and absences in class were also plotted and it was observed that most of the students had managed to pass the exam. The histograms for G3, failures and absences are provided in the Appendix. (Refer Appendix: Exploratory Data Analysis: Histograms for failures and absences).

Correlation Analysis

Correlation plot helped in exploring the data and understanding the relationships between the numeric data present in the dataset. It could be observed that there is a high correlation between G1, G2, and G3. It implies that the performing students in class would get high values of G1, G2 and hence G3 whereas poor performing students would get low values of G1, G2 and hence G3 . Failure has a negative correlation with grades G1, G2 and G3. There is also correlation between father's and mother's education. The weekend and weekday alcohol consumption are correlated as well. The correlation and ggplot2 pair plot are shown below.





Linear Regression

Linear Regression is a data analysis and statistical model that is used primarily for the following: (a) to see if an independent variable is correctly predicting the dependent variable and (b) which are the best predictors for the dependent (response) variable.

Developing the Model

The model was developed by splitting the dataset into train and test sets with the train set being 70% of the data and the test being the remaining 30%. The R code for model creation using the train and test sets is provided in the Appendix (Refer Appendix: R Code for Linear Regression model with numeric predictors).

The model was built to predict grade G3 based on the expected predictor variables. Figure 1 below shows the output of the model with the numeric variables.

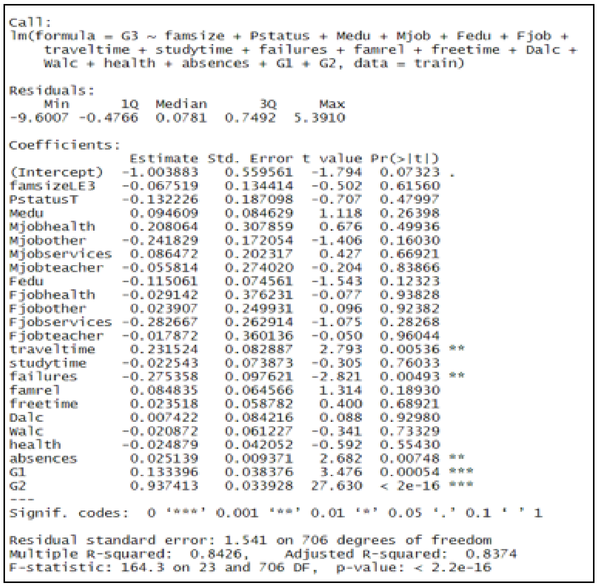
As can be observed from the above screenshot of the model output, the significant predictors at 0.05 level of significance are 'G1','G2', 'travel time,' 'absences' and 'past failures,' all of which have positive coefficients (except past failures) and are hence positively correlated to grade G3. Hence the model indicates that the grade or performance of the student varies with low and high values of G1, G2, travel time, absences, and past failures. The accuracy of the model was validated to check if it fits well on the test dataset.

The overall quality of the linear regression fit can be assessed using the following three quantities: Residual Standard Error (RSE), Adjusted R2, and F-Statistic. The lower the RSE, the better the model fits the data. In the first linear model, RSE was 1.54 which meant that the observed value deviates from the predicted value by 1.54 units. The Error Rate of 14% was very low. The Adjusted R2 was 0.8374, which is good. The F-Statistic was 164.3 producing a p-value less than 0.05, which indicates that the model is highly significant (Refer Appendix: Results for first Linear Model Accuracy and Error Rate)

Developing the Model with Significant Predictors

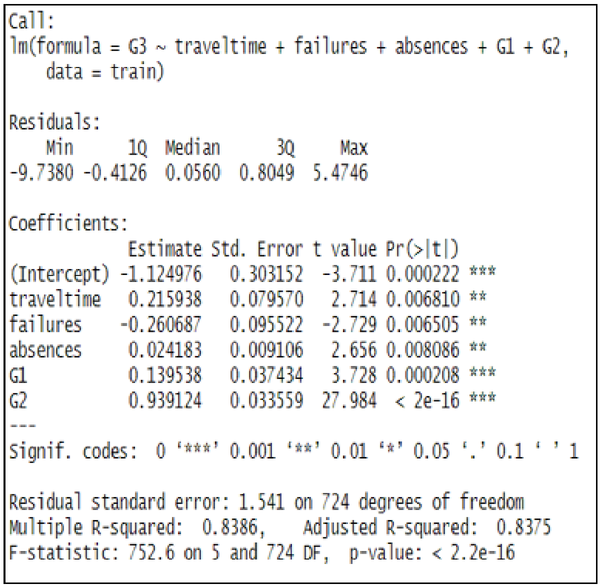
The linear regression model was developed by using the significant predictors ‘G1’,’G2’, ‘travel time,’ ‘absences’ and ‘past failures’ obtained from the previous model. The model was developed by splitting the data into train and test sets with the train set being 70% of the data and the test being the remaining 30%. The R code for model creation with significant predictors using train data is given in the Appendix (Refer Appendix: R Code for Linear Regression model with significant predictors). Figure 2 shows the output of the model developed.

Figure 1: Model with numeric variables



```
Call:
lm(formula = G3 ~ famsize + Pstatus + Medu + Mjob + Fedu + Fjob +
    traveltime + studytime + failures + famrel + freetime + Dalc +
    Walc + health + absences + G1 + G2, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-9.6007 -0.4766  0.0781  0.7492  5.3910

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.003883   0.559561  -1.794  0.07323 .
famsizeLE3   -0.067519   0.134414  -0.502  0.61560
PstatusT     -0.132226   0.187098  -0.707  0.47997
Medu          0.094609   0.084629   1.118  0.26398
Mjobhealth    0.208064   0.307859   0.676  0.49936
Mjobother    -0.241829   0.172054  -1.406  0.16030
Mjobservices  0.086472   0.202317   0.427  0.66921
Mjobteacher  -0.055814   0.274020  -0.204  0.83866
Fedu         -0.115061   0.074561  -1.543  0.12323
Fjobhealth   -0.029142   0.376231  -0.077  0.93828
Fjobother     0.023907   0.249931   0.096  0.92382
Fjobservices -0.282667   0.262914  -1.075  0.28268
Fjobteacher  -0.017872   0.360136  -0.050  0.96044
traveltime    0.231524   0.082887   2.793  0.00536 **
studytime    -0.022543   0.073873  -0.305  0.76033
failures     -0.275358   0.097621  -2.821  0.00493 **
famrel        0.084835   0.064566   1.314  0.18930
freetime      0.023518   0.058782   0.400  0.68921
Dalc          0.007422   0.084216   0.088  0.92980
Walc         -0.020872   0.061227  -0.341  0.73329
health       -0.024879   0.042052  -0.592  0.55430
absences      0.025139   0.009371   2.682  0.00748 **
G1            0.133396   0.038376   3.476  0.00054 ***
G2            0.937413   0.033928  27.630  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.541 on 706 degrees of freedom
Multiple R-squared:  0.8426,    Adjusted R-squared:  0.8374
F-statistic: 164.3 on 23 and 706 DF,  p-value: < 2.2e-16
```

Figure 2: Model with significant predictors



```
Call:
lm(formula = G3 ~ traveltime + failures + absences + G1 + G2,
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7380 -0.4126  0.0560  0.8049  5.4746

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.124976   0.303152  -3.711 0.000222 ***
traveltime   0.215938   0.079570   2.714 0.006810 **
failures    -0.260687   0.095522  -2.729 0.006505 **
absences     0.024183   0.009106   2.656 0.008086 **
G1           0.139538   0.037434   3.728 0.000208 ***
G2           0.939124   0.033559  27.984  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.541 on 724 degrees of freedom
Multiple R-squared:  0.8386,    Adjusted R-squared:  0.8375
F-statistic: 752.6 on 5 and 724 DF,  p-value: < 2.2e-16
```

Also, the overall quality of the linear regression fit can be assessed using the following three quantities: Residual Standard Error (RSE), Adjusted R2, and F-Statistic. The results of RSE, Accuracy, F-statistic and Error Rate for the model are provided in the Appendix. (Refer Appendix: Output: Linear Regression Accuracy, RSE, Error Rate calculations and Prediction on test data)

In the revised model, RSE is 1.54, which means that the observed value deviates from the predicted value by 1.54 units. It is the same as previous model RSE value. The Error Rate came down to 13%, which is very low. The Adjusted R2 is 0.837, which is good. The F-Statistic has improved to 752.6, producing a p-value less than 0.05, which is highly significant (Refer Appendix: Results for Linear Regression RSE, Error Rate calculations).

Predictions

The predictions were made using test data in order to evaluate the performance of our regression model. The value of R2 is 0.83 implies that observed and the predicted outcome values are highly correlated, which is very good. The prediction error RMSE is 1.63, representing an error rate of 14.4 %, which is good. The grade G3 was predicted using test data. Here the model is predicting some negative values. The results of the analysis for model are provided in the Appendix (Refer Appendix - Output: Linear Regression Accuracy, RSE, Error Rate calculations and Prediction on test data)

Diagnostic Plots of Linear Regression

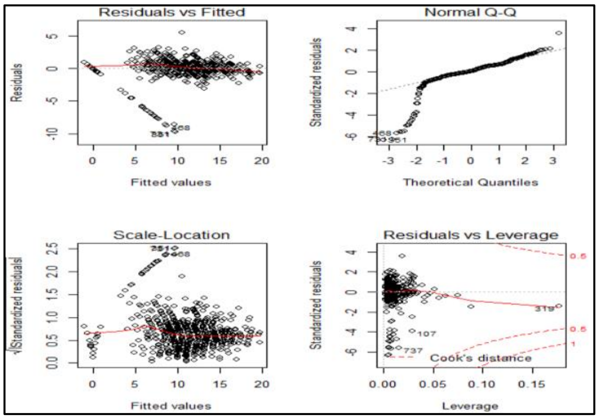
The assumptions are verified by plotting the residuals.

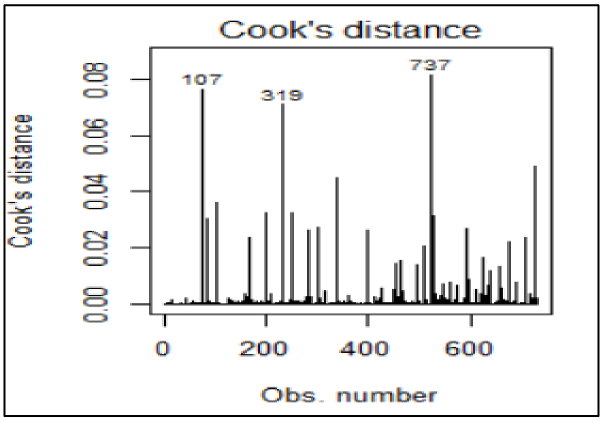
Linearity of the data: The linearity assumption was checked by inspecting the Residuals vs Fitted plot (1st plot). The red line is approximately horizontal at zero, and there is no pattern in the residual plot. It suggested that linearity between the predictors and the outcome variables holds.

Homogeneity of variance: The scale-location plot shows that residuals are not spread equally along the range of predictors. The variability of the residual points is decreasing with the value of the fitted outcome variable suggesting heteroscedasticity.

Normality of residuals: The normal probability plot of residuals is not following a straight line. The normality assumption does not hold true.

Outliers and influential values: The residual vs. leverage plot shows the top 3 most extreme points (#107, #319, #737), with a standardized residual below -5. The points 107, 319, 737 data points are outside the Cook’s distance lines. The observation 737 has larger Cook's distance than other data points in Cook's distance plot; this observation doesn't stand out in other plots. We can say that the values are influential to the regression results.





Logistic Regression

Logistic Regression is a data analysis and statistical model that is used to predict the probability of a binary or dichotomous response variable. The overall grade of a student - G3, being interval data, needed to be converted to a dichotomous variable to perform logistic regression on it. All predictor variables were also converted to dichotomous variables for better variability and classification accuracy in the model.

Creating Dichotomous Variables

The dichotomous variable created for the dependent variable – G3 was ‘pooroverallgrade.’ Overall grades less than or equal to 10 (median of the grades) were classified as poor grades. The dichotomous variables created for the predictors considered in the logistic regression model are – ‘Large family’, ‘Separated parents’, ‘Well educated mother’, ‘Working mother’, ’Well educated father’, ’Working father’, ‘High travel time’, ’Less study time’, ’High past failures’, ‘Poor family relationships’, ‘Less free time’, ‘High social life’, ‘High alcohol consumption’, ‘Poor health’, ’High absenteeism’, ‘low G1’(low 1st period grade) and ’low G2’(low second period grade).

Developing the Model

The logistic regression model was developed by splitting the dataset into train and test sets with the train set being 70% of the data and the test being the remaining 30%. The R code for model creation using the train and test sets is provided in the Appendix (Refer Appendix: R Code for Logistic Regression Model Creation). Figure 3 shows the model output.

As can be observed from the screenshot of the model output (Figure 3), the significant predictors at 0.1 level of significance are ‘G1’,’G2’, ‘travel time,’ and ‘past failures,’ all of which have positive coefficients and are hence positively correlated to poor grades. Hence the model indicates that the grade or performance of the student decreases with low G1, low G2, high travel time and high past failures.

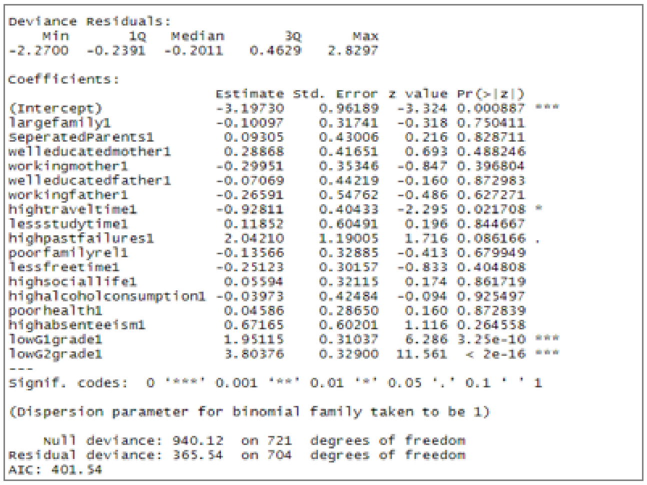
The model was then validated to determine if it fits well on the test set, and the classification accuracy and probabilities were computed. The R code for the above is provided in the Appendix (Refer Appendix: R Code for Testing the Trained Logistic Regression Model). On calculating McFadden's pseudo R squared for the model (equivalent of Adjusted R Squared in Linear Regression), the value obtained was 0.61 which indicates that the model is an excellent fit. The classification accuracy of the model turned out to be 92.54%.

The R code and the results for the McFadden's pseudo R squared, classification accuracy, Analysis of Variance (ANOVA), prediction of probabilities, and the confusion matrix is provided in the Appendix. (Refer Appendix: Results of Analysis of Logistic Regression Model)

Developing Model with Significant Predictors

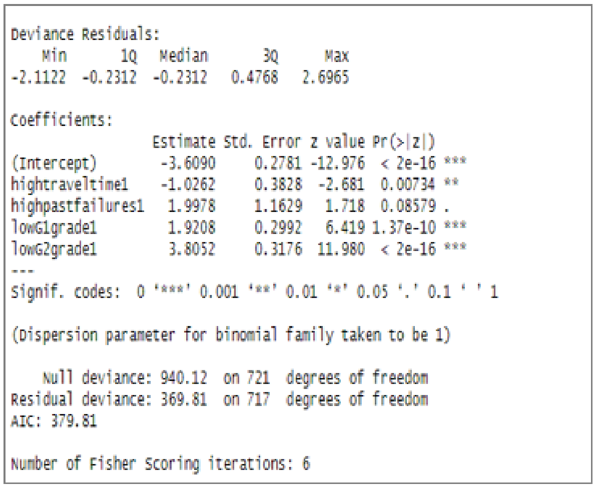
The model was developed again using only the significant predictors from the previous model. McFadden's pseudo R squared value for the new model was 0.6 which indicates that the model is an excellent fit. The classification accuracy of the model increased to 93.47%. Figure 4 shows model developed with significant predictors.

Figure 3: Model with numeric variables



```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2700  -0.2391  -0.2011   0.4629   2.8297

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             -3.19730    0.96189  -3.324 0.000887 ***
largefamily1            -0.10097    0.31741  -0.318 0.750411
seperatedParents1        0.09305    0.43006   0.216 0.828711
welleducatedmother1      0.28868    0.41651   0.693 0.488246
workingmother1          -0.29951    0.35346  -0.847 0.396804
welleducatedfather1     -0.07069    0.44219  -0.160 0.872983
workingfather1          -0.26591    0.54762  -0.486 0.627271
hightraveltime1         -0.92811    0.40433  -2.295 0.021708 *
lessstudytime1           0.11852    0.60491   0.196 0.844667
highpastfailures1        2.04210    1.19005   1.716 0.086166 .
poorfamilyrel1          -0.13566    0.32885  -0.413 0.679949
lessfreetime1           -0.25123    0.30157  -0.833 0.404808
highsociallife1          0.05594    0.32115   0.174 0.861719
highalcoholconsumption1 -0.03973    0.42484  -0.094 0.925497
poorhealth1              0.04586    0.28650   0.160 0.872839
highabsenteeism1         0.67165    0.60201   1.116 0.264558
lowG1grade1              1.95115    0.31037   6.286 3.25e-10 ***
lowG2grade1              3.80376    0.32900  11.561  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 940.12  on 721  degrees of freedom
Residual deviance: 365.54  on 704  degrees of freedom
AIC: 401.54
```

Figure 4: Model with significant predictors



```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1122  -0.2312  -0.2312   0.4768   2.6965

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -3.6090     0.2781 -12.976  < 2e-16 ***
hightraveltime1    -1.0262     0.3828  -2.681  0.00734 **
highpastfailures1   1.9978     1.1629   1.718  0.08579 .
lowG1grade1         1.9208     0.2992   6.419 1.37e-10 ***
lowG2grade1         3.8052     0.3176  11.980  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 940.12  on 721  degrees of freedom
Residual deviance: 369.81  on 717  degrees of freedom
AIC: 379.81

Number of Fisher Scoring iterations: 6
```

The results of the analysis for model using significant predictors are provided in the Appendix. (Refer Appendix: Results of Analysis of Logistic Regression Model using Significant Predictors)

Random Forest

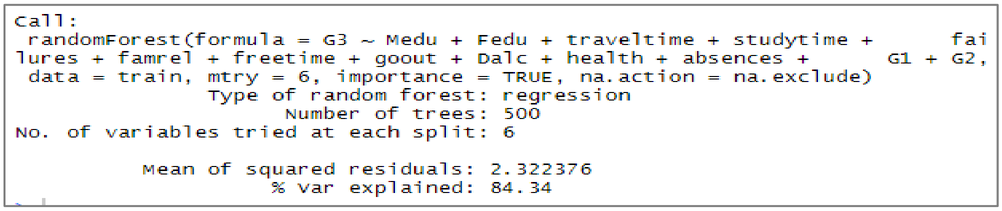
Random Forest is a classification model that builds a multitude of decision trees using feature randomness and outputs a class, which is the mean prediction of the decision trees. A large number of trees produce stable models. In this paper, Random forest was applied to the dataset to predict the student performance on the test set, determine the critical features or variables predicting it and obtain the mean of the decision trees.

Predictor Variables for Random Forest

The predictor variables for Random Forest include mother's education, father's education, travel time, study time, failures, family relationships, free time, social life, alcohol consumption, health, absences, first-period grade(G1) and second-period grade(G2). The response variable is the overall performance or grade of the students which is given by G3.

Developing the Model

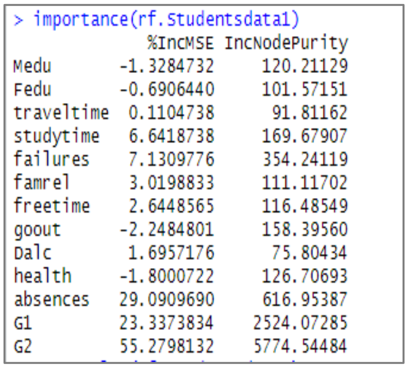
The model was developed by splitting the dataset into train and test sets with a train set being 70% of the data and test being the remaining 30%. The random forest model was developed with 500 trees and a bunch of ‘mtry’ values to determine the best and optimal value that tunes the model and provides high accuracy. The R code for model creation using train and test sets is provided in the Appendix (Refer Appendix: R Code for Random Forest Model Creation). The output of the model can be seen in the following screenshot: -



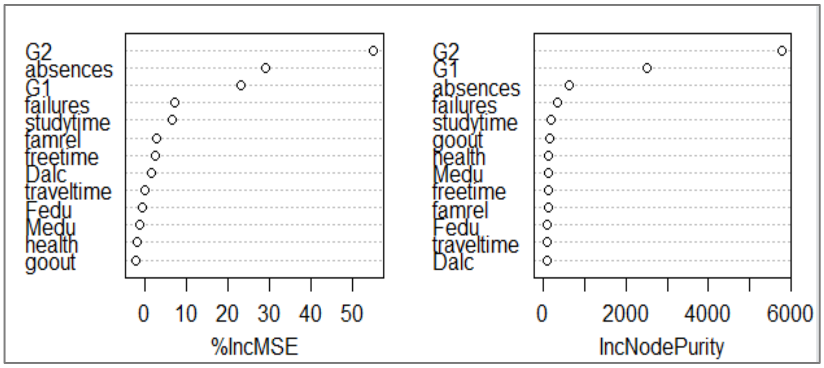
```
Call:
 randomForest(formula = G3 ~ Medu + Fedu + traveltime + studytime +       fai
lures + famrel + freetime + goout + Dalc + health + absences +      G1 + G2,
 data = train, mtry = 6, importance = TRUE, na.action = na.exclude)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 2.322376
                    % Var explained: 84.34
```

As can be observed from the above screenshot, the model indicated the right amount of variability with its variance being 84.34% when mtry=6 and the number of trees is 500. The mean of squared residuals is 2.32, which is a low value and indicates that the model is a good fit. The model was run on the test set and the mean prediction was 2.34.

On determining the variable importance, it was observed that the most important variables are G1, G2, absences, and failures. This can be determined from the screenshot of the output given below: -

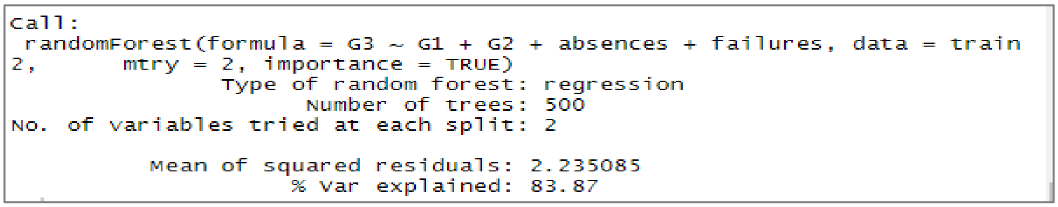


```
> importance(rf.Studentsdata1)
              %IncMSE IncNodePurity
Medu       -1.3284732     120.21129
Fedu       -0.6906440     101.57151
traveltime  0.1104738      91.81162
studytime   6.6418738     169.67907
failures    7.1309776     354.24119
famrel      3.0198833     111.11702
freetime    2.6448565     116.48549
goout      -2.2484801     158.39560
Dalc        1.6957176      75.80434
health     -1.8000722     126.70693
absences   29.0909690     616.95387
G1         23.3373834    2524.07285
G2         55.2798132    5774.54484
```



On trying different values for ‘mtry,’ it was observed that the model indicated the highest variability with its variance being 84.44% when mtry was 8, and the number of trees was 500. The mean of squared residuals was also lesser and was equal to 2.3.

The model was finally developed using only the four most essential features obtained in the previous model. The number of decision trees considered was 500, and the value of mtry was 2. 83.87% of the variance was explained in the model. The mean of squared residuals was the least and was equal to 2.23 thus indicating that the model is a good fit as can be seen in the screenshot below. The R code for random forest using the significant predictors are provided in the Appendix (Refer Appendix: R Code for Random Forest with Significant Predictors)



```
Call:
 randomForest(formula = G3 ~ G1 + G2 + absences + failures, data = train
2,       mtry = 2, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 2.235085
                    % Var explained: 83.87
```

Classification Tree

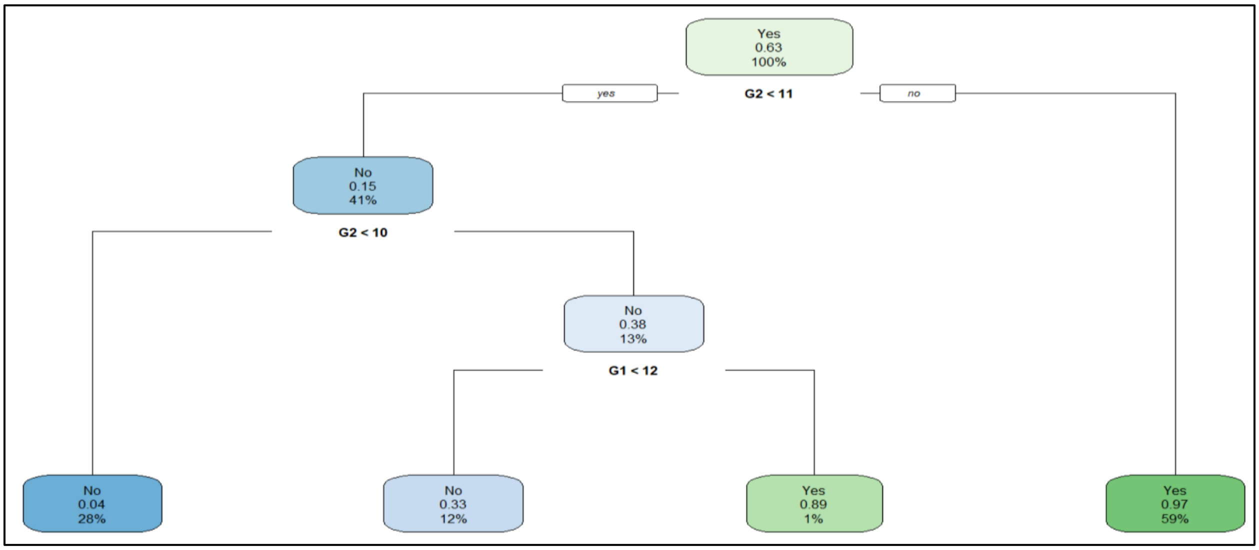
Classification trees is used to predict a categorical outcome. The purpose of creating this model is to predict how many students are more likely to be high performing students in class. Classification tree was applied to the dataset to predict if the student performance if high or not. For it, we created a new variable “High” using grade G3 containing values “Yes/No”. The data was shuffled to avoid poor prediction.

Developing the Model

The model was developed by splitting the dataset into train and test sets with train set being 70% of the data and test being the remaining 30%. The classification tree was developed with all the variables to provide high accuracy. The R code for model creation using train and test sets is provided in the Appendix (Refer Appendix: R Code for Classification Tree model creation with all variables).

On determining the variable importance, it was observed that the most important variables are G1, G2, higher education, Mothers education, and failures. This can be determined from the screenshot of the output given in the Appendix (Refer Appendix - Results for Classification Tree model important predictors in decreasing order).

The output of the model can be seen in the following screenshot: -



At the top, it is the overall probability of scoring High marks by a student. 63% of students will score high marks. This node asks whether grade G2 <10. It shows 41% of the non-performing students will have 15% probability of scoring high marks. If G1 is not less than 10, then at the next node, it checks if G1<12. If yes, then 89% of the students will be high performing students.

Predictions

The model was tested on test data to predict the accuracy of the model. From the confusion matrix, we can say that it correctly predicted 101 non-performing students but misclassified 13 good students as non-performing. By analogy, the model misclassified 15 students as high performing while they turned out to be non-performing students. The accuracy of the model on test dataset is 91.08%, which is very high. The results for Confusion matrix and Accuracy is given in the Appendix. (Refer Appendix: Results for Classification Tree prediction on Test Data, Confusion Matrix and Accuracy calculation).

We tried to improve the model’s accuracy over default value. A function was developed to return the accuracy by tuning the parameters like maximum depth, minimum split, and minimum number of sample leaf nodes. However, the accuracy remained the same as previous model i.e. 91.08%. The results for function creation for tuning the parameters and tuning accuracy calculation is given in Appendix. (Refer Appendix: Results for Tuning the parameters of Classification Tree and tuned Accuracy).

Challenges

Creating dichotomous variables for many predictor variables in logistic regression was a challenge. Dichotomous variables had to be created for the predictors as they introduced better variability in the model. The classification tree, despite having high accuracy was not displaying all the decision nodes and the branches. Finding the optimal mtry for Random Forest that would give good variability and the best accuracy.

Special Efforts

Dichotomous variables were created for all the predictors in order to improve the accuracy and variability in the logistic regression model. The datasets for Portuguese and Math subjects were merged using R code, and a field was created to identify the data corresponding to the subject.

Conclusion

Multiple data analysis methods can be used on a dataset based on the requirement, feasibility, and how the data needs to be interpreted. Identifying the best predictive methods, improving the accuracy of the created models and reducing the error is the key to data analysis and interpretation. In this paper, all the four models gave an accuracy of more than 80% with low residual or error values, thus indicating that the models created were a good fit but not overfitted. Logistic Regression had the highest accuracy of all the models (93.47%) followed by Classification Tree (91.08%). The best predictors of the overall performance of students in the schools are a first-period grade(G1), second-period grade(G2), absences in class, past failures and travel time to school.

References

i. Student Mathematics and Portuguese Dataset. (2014) [Data file]. Retrieved from https://archive.ics.uci.edu/ml/datasets/student+performance

ii. https://globaljournals.org/GJMBR_Volume12/3-Factors-Affecting-Students-Academic.pdf

iii. R Development Core Team. (2018). R: A Language and Environment for Statistical Computing. [Software]. Vienna, Austria: R Foundation for Statistical Computing, Vienna, Austria.

iv. Taiyun Wei and Viliam Simko (2017). R package ‘corrplot’: Visualization of a Correlation Matrix (Version 0.84). Available from https://github.com/taiyun/corrplot

v. Wickham, Hadly. (2017). ggplot2: Elegant Graphics for Data Analysis (3.1.0) [Package]. New York:Springer-Verlag.

vi. Hadley Wickham (2017). tidyverse: Easily Install and Load the 'Tidyverse.' R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

vii. Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.1. https://CRAN.R-project.org/package=dplyr

viii. Jarek Tuszynski (2019). caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc. R package version 1.17.1.2. https://CRAN.R-project.org/package=caTools

ix. Jared P. Lander (2018). coefplot: Plots Coefficients from Fitted Models. R package version 1.2.6. https://CRAN.R-project.org/package=coefplot

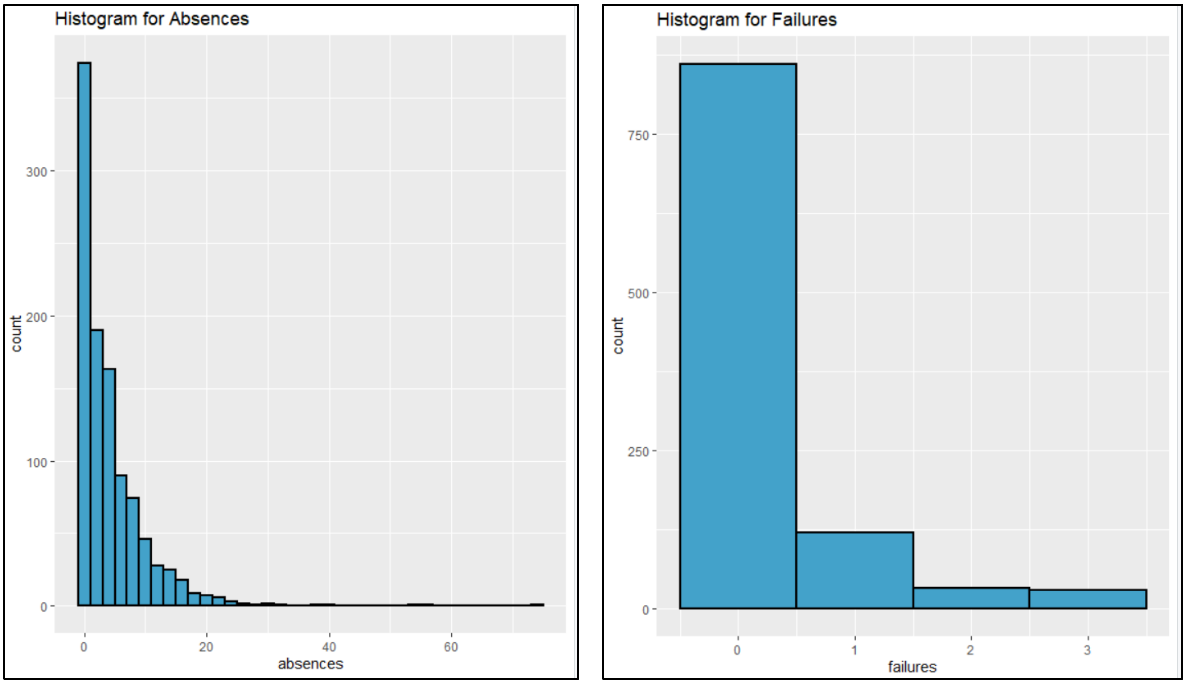
x. Terry Therneau and Beth Atkinson (2019). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15. https://CRAN.R-project.org/package=rpart

xi. Williams, G. J. (2011), Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery, Use R!, Springer.

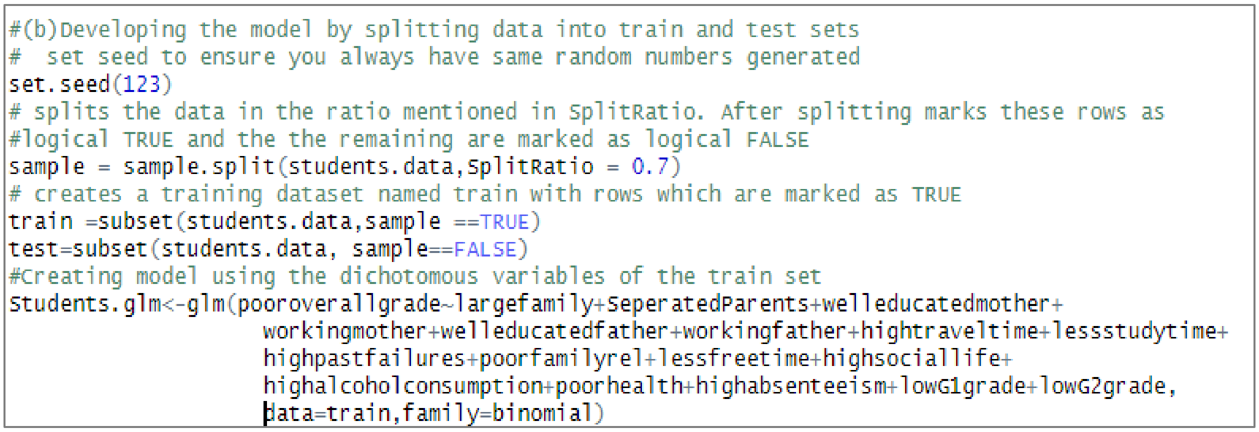
xii. Gareth James, Daniela Witten, Trevor Hastie, and Rob Tibshirani (2017). ISLR: Data for an Introduction to Statistical Learning with Applications in R. R package version 1.2. https://CRAN.R-project.org/package=ISLR

Appendix

Exploratory Data Analysis: Histograms for failures and absences

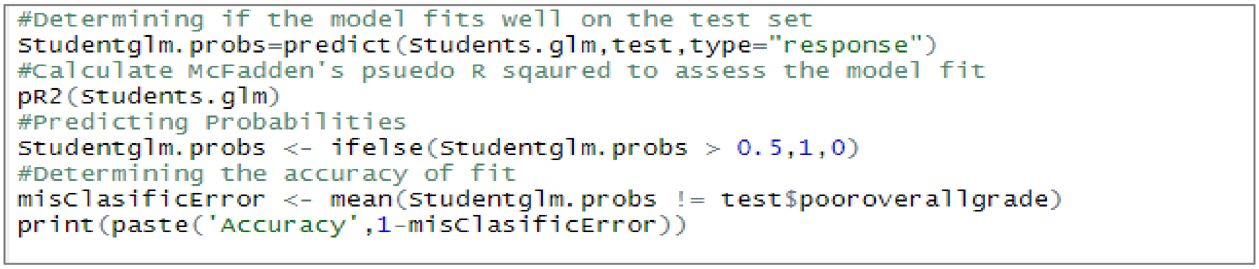


R Code for Logistic Regression Model Creation



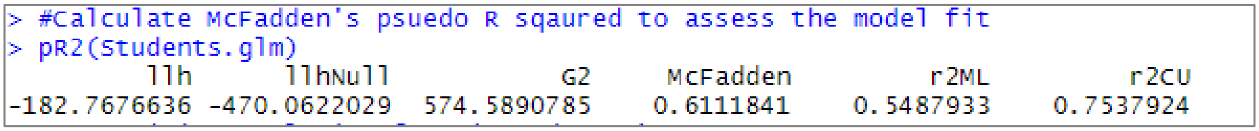
```
#(b)Developing the model by splitting data into train and test sets
#  set seed to ensure you always have same random numbers generated
set.seed(123)
# splits the data in the ratio mentioned in SplitRatio. After splitting marks these rows as
#logical TRUE and the the remaining are marked as logical FALSE
sample = sample.split(students.data,SplitRatio = 0.7)
# creates a training dataset named train with rows which are marked as TRUE
train =subset(students.data,sample ==TRUE)
test=subset(students.data, sample==FALSE)
#Creating model using the dichotomous variables of the train set
Students.glm<-glm(pooroverallgrade~largefamily+SeperatedParents+welleducatedmother+
                          workingmother+welleducatedfather+workingfather+hightraveltime+lessstudytime+
                          highpastfailures+poorfamilyrel+lessfreetime+highsociallife+
                          highalcoholconsumption+poorhealth+highabsenteeism+lowG1grade+lowG2grade,
                          data=train,family=binomial)
```

R Code for Testing the Trained Logistic Regression Model

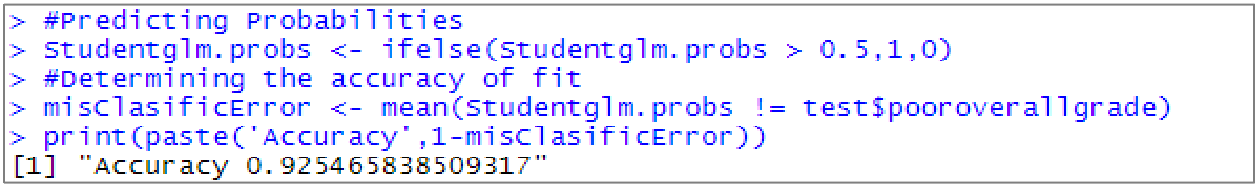


```
#Determining if the model fits well on the test set
Studentglm.probs=predict(Students.glm,test,type="response")
#Calculate McFadden's psuedo R sqaured to assess the model fit
pR2(Students.glm)
#Predicting Probabilities
Studentglm.probs <- ifelse(Studentglm.probs > 0.5,1,0)
#Determining the accuracy of fit
misClasificError <- mean(Studentglm.probs != test$pooroverallgrade)
print(paste('Accuracy',1-misClasificError))
```

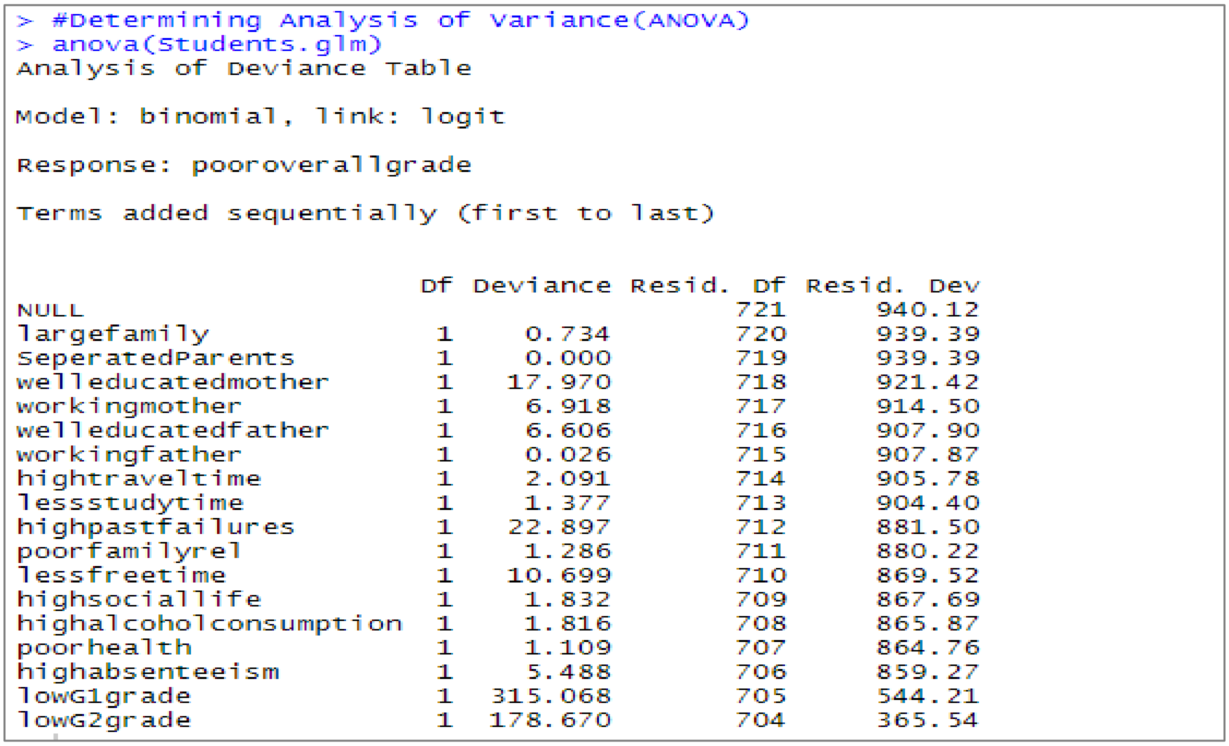
Results of Analysis of Logistic Regression Model



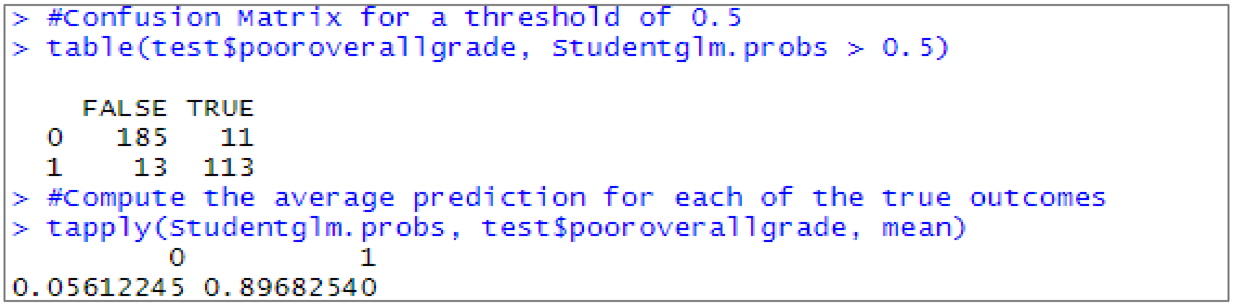
```
> #Calculate McFadden's psuedo R sqaured to assess the model fit
> pR2(Students.glm)
         llh       llhNull            G2      McFadden          r2ML          r2CU 
-182.7676636 -470.0622029  574.5890785    0.6111841     0.5487933     0.7537924 
```



```
> #Predicting Probabilities
> Studentglm.probs <- ifelse(Studentglm.probs > 0.5,1,0)
> #Determining the accuracy of fit
> misClasificError <- mean(Studentglm.probs != test$pooroverallgrade)
> print(paste('Accuracy',1-misClasificError))
[1] "Accuracy 0.925465838509317"
```

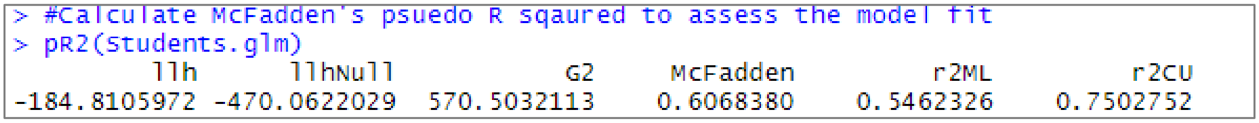


```
> #Determining Analysis of Variance(ANOVA)
> anova(Students.glm)
Analysis of Deviance Table

Model: binomial, link: logit

Response: pooroverallgrade

Terms added sequentially (first to last)


                       Df Deviance Resid. Df Resid. Dev
NULL                                     721     940.12
largefamily             1    0.734       720     939.39
SeperatedParents        1    0.000       719     939.39
welleducatedmother      1   17.970       718     921.42
workingmother           1    6.918       717     914.50
welleducatedfather      1    6.606       716     907.90
workingfather           1    0.026       715     907.87
hightraveltime          1    2.091       714     905.78
lessstudytime           1    1.377       713     904.40
highpastfailures        1   22.897       712     881.50
poorfamilyrel           1    1.286       711     880.22
lessfreetime            1   10.699       710     869.52
highsociallife          1    1.832       709     867.69
highalcoholconsumption  1    1.816       708     865.87
poorhealth              1    1.109       707     864.76
highabsenteeism         1    5.488       706     859.27
lowG1grade              1  315.068       705     544.21
lowG2grade              1  178.670       704     365.54
```

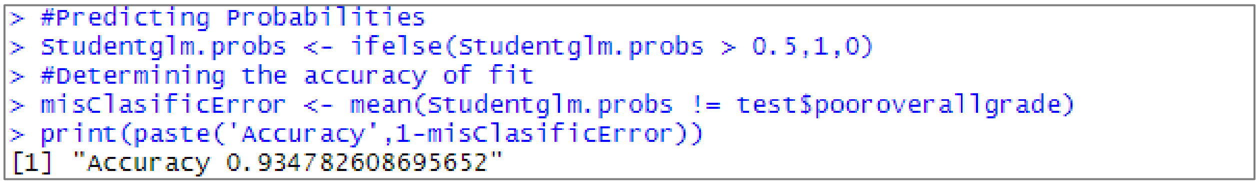


```
> #Confusion Matrix for a threshold of 0.5
> table(test$pooroverallgrade, Studentglm.probs > 0.5)

    FALSE TRUE
  0   185   11
  1    13  113
> #Compute the average prediction for each of the true outcomes
> tapply(Studentglm.probs, test$pooroverallgrade, mean)
         0          1
0.05612245 0.89682540
```

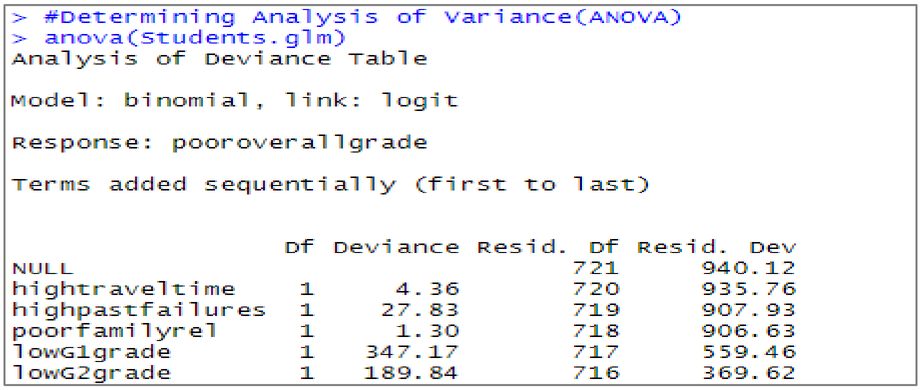
Results of Analysis of Logistic Regression Model using Significant Predictors



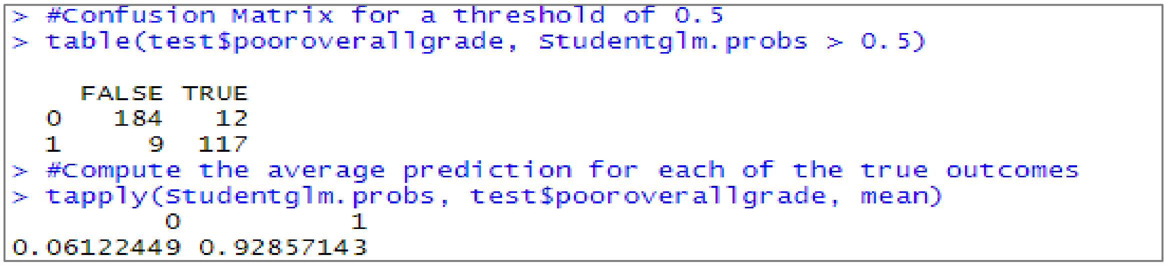
```
> #Calculate McFadden's psuedo R sqaured to assess the model fit
> pR2(Students.glm)
         llh      llhNull           G2     McFadden         r2ML         r2CU
-184.8105972 -470.0622029  570.5032113    0.6068380    0.5462326    0.7502752
```



```
> #Predicting Probabilities
> Studentglm.probs <- ifelse(Studentglm.probs > 0.5,1,0)
> #Determining the accuracy of fit
> misClasificError <- mean(Studentglm.probs != test$pooroverallgrade)
> print(paste('Accuracy',1-misClasificError))
[1] "Accuracy 0.934782608695652"
```

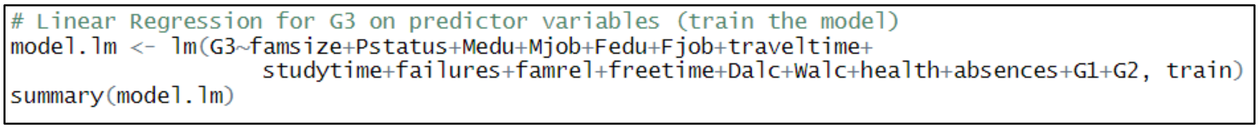


```
> #Determining Analysis of Variance(ANOVA)
> anova(Students.glm)
Analysis of Deviance Table

Model: binomial, link: logit

Response: pooroverallgrade

Terms added sequentially (first to last)


                  Df Deviance Resid. Df Resid. Dev
NULL                                721     940.12
hightraveltime     1     4.36       720     935.76
highpastfailures   1    27.83       719     907.93
poorfamilyrel      1     1.30       718     906.63
lowG1grade         1   347.17       717     559.46
lowG2grade         1   189.84       716     369.62
```



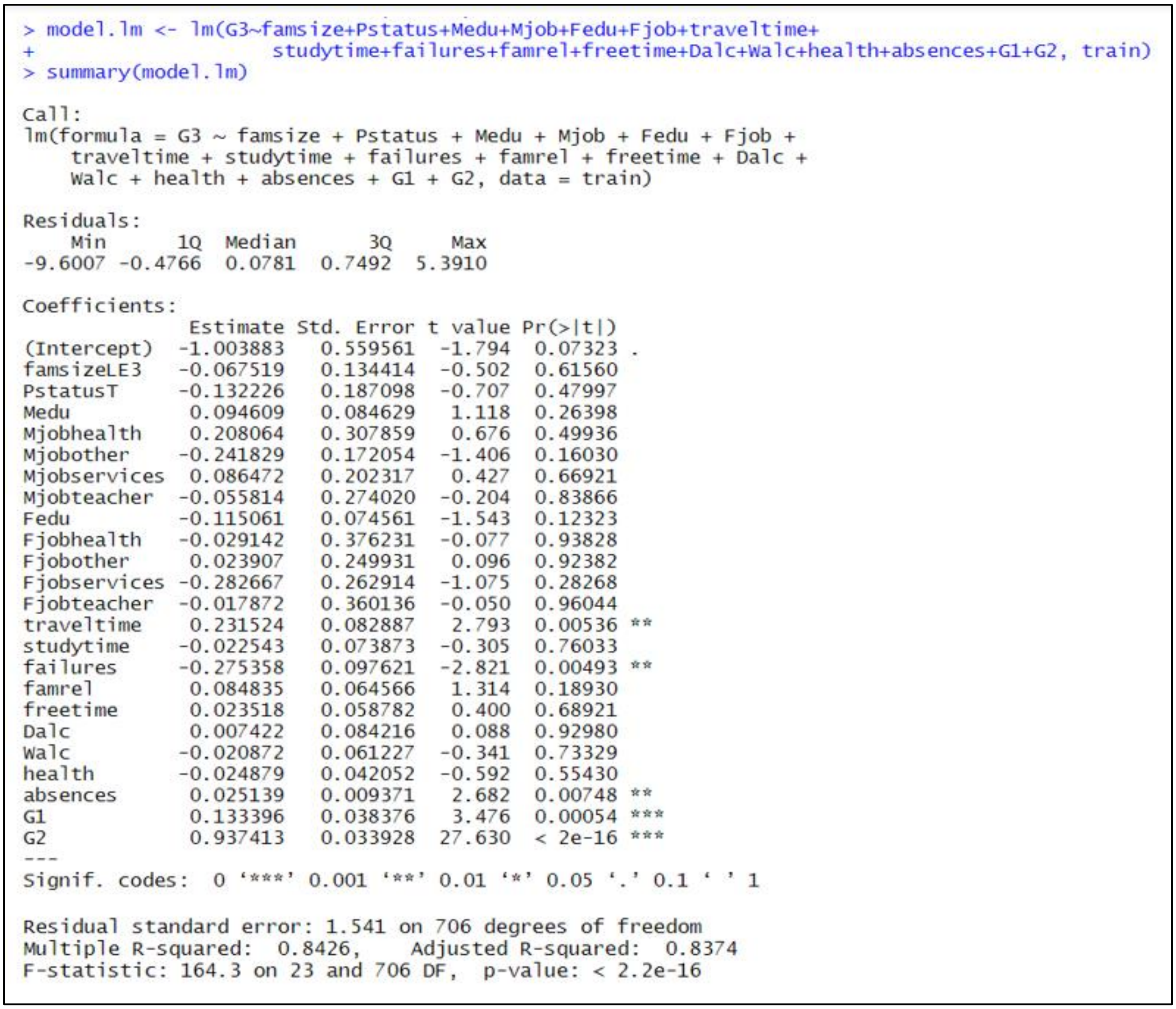
```
> #Confusion Matrix for a threshold of 0.5
> table(test$pooroverallgrade, Studentglm.probs > 0.5)

    FALSE TRUE
  0   184   12
  1     9  117
> #Compute the average prediction for each of the true outcomes
> tapply(Studentglm.probs, test$pooroverallgrade, mean)
         0          1
0.06122449 0.92857143
```

R Code for Linear Regression model with numeric predictors



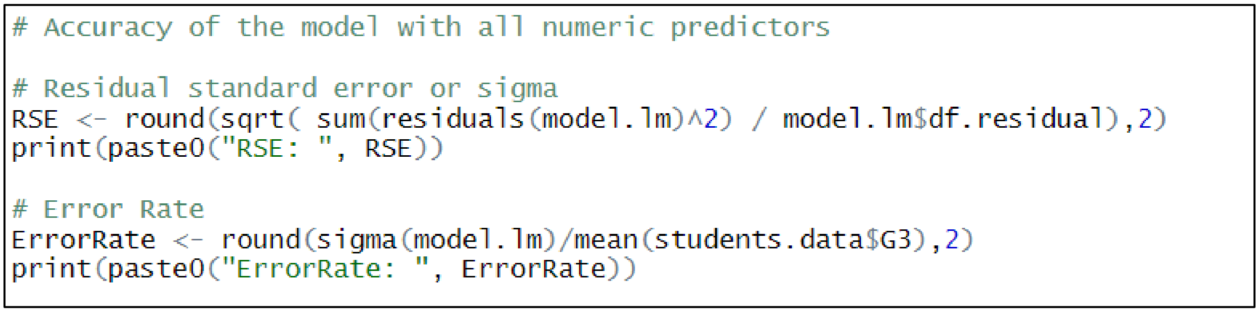
```
# Linear Regression for G3 on predictor variables (train the model)
model.lm <- lm(G3~famsize+Pstatus+Medu+Mjob+Fedu+Fjob+traveltime+
                 studytime+failures+famrel+freetime+Dalc+Walc+health+absences+G1+G2, train)
summary(model.lm)
```

Results for Linear Regression model output with numeric predictors



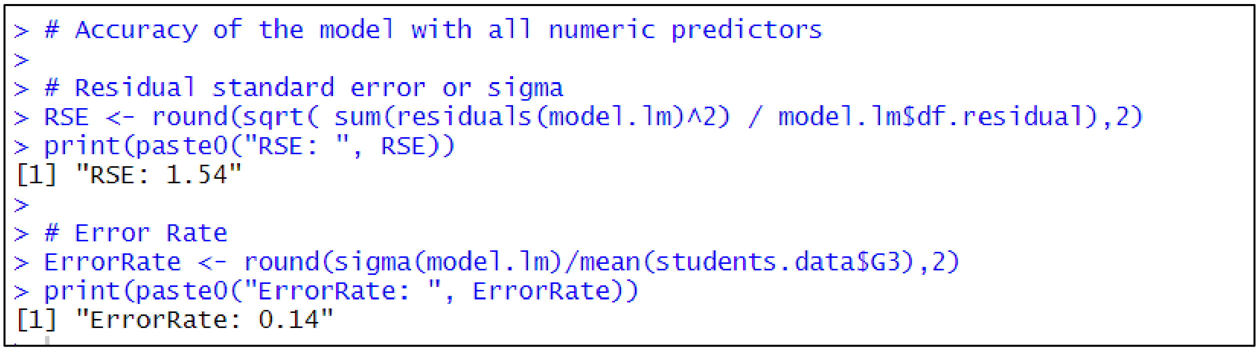
```
> model.lm <- lm(G3~famsize+Pstatus+Medu+Mjob+Fedu+Fjob+traveltime+
+                     studytime+failures+famrel+freetime+Dalc+Walc+health+absences+G1+G2, train)
> summary(model.lm)

Call:
lm(formula = G3 ~ famsize + Pstatus + Medu + Mjob + Fedu + Fjob + 
    traveltime + studytime + failures + famrel + freetime + Dalc + 
    Walc + health + absences + G1 + G2, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.6007 -0.4766  0.0781  0.7492  5.3910 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.003883   0.559561  -1.794  0.07323 .  
famsizeLE3  -0.067519   0.134414  -0.502  0.61560    
PstatusT    -0.132226   0.187098  -0.707  0.47997    
Medu         0.094609   0.084629   1.118  0.26398    
Mjobhealth   0.208064   0.307859   0.676  0.49936    
Mjobother   -0.241829   0.172054  -1.406  0.16030    
Mjobservices  0.086472   0.202317   0.427  0.66921    
Mjobteacher -0.055814   0.274020  -0.204  0.83866    
Fedu        -0.115061   0.074561  -1.543  0.12323    
Fjobhealth  -0.029142   0.376231  -0.077  0.93828    
Fjobother    0.023907   0.249931   0.096  0.92382    
Fjobservices -0.282667   0.262914  -1.075  0.28268    
Fjobteacher -0.017872   0.360136  -0.050  0.96044    
traveltime   0.231524   0.082887   2.793  0.00536 ** 
studytime   -0.022543   0.073873  -0.305  0.76033    
failures    -0.275358   0.097621  -2.821  0.00493 ** 
famrel       0.084835   0.064566   1.314  0.18930    
freetime     0.023518   0.058782   0.400  0.68921    
Dalc         0.007422   0.084216   0.088  0.92980    
Walc        -0.020872   0.061227  -0.341  0.73329    
health      -0.024879   0.042052  -0.592  0.55430    
absences     0.025139   0.009371   2.682  0.00748 ** 
G1           0.133396   0.038376   3.476  0.00054 ***
G2           0.937413   0.033928  27.630  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.541 on 706 degrees of freedom
Multiple R-squared:  0.8426,	Adjusted R-squared:  0.8374 
F-statistic: 164.3 on 23 and 706 DF,  p-value: < 2.2e-16
```

R Code for first Linear Model Accuracy and Error Rate



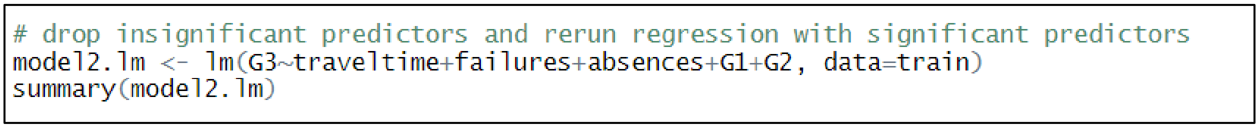
```
# Accuracy of the model with all numeric predictors

# Residual standard error or sigma
RSE <- round(sqrt( sum(residuals(model.lm)^2) / model.lm$df.residual),2)
print(paste0("RSE: ", RSE))

# Error Rate
ErrorRate <- round(sigma(model.lm)/mean(students.data$G3),2)
print(paste0("ErrorRate: ", ErrorRate))
```

Results for first Linear Model Accuracy and Error Rate



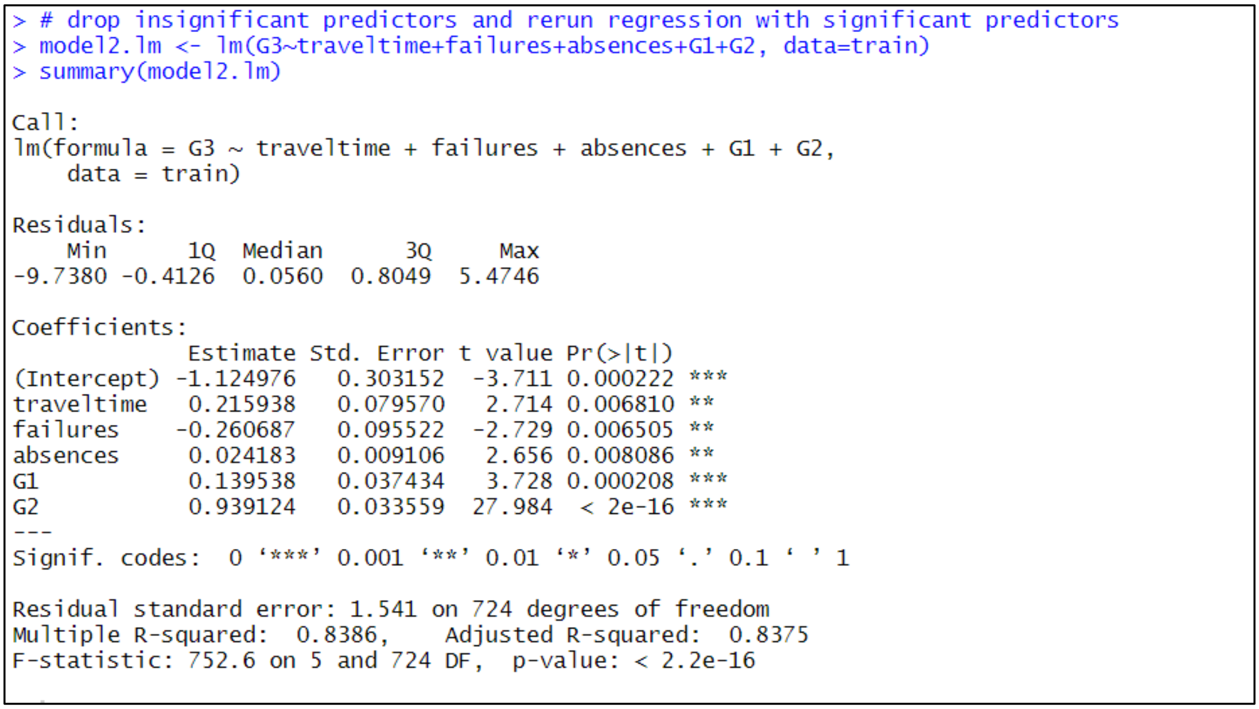
```
> # Accuracy of the model with all numeric predictors
> 
> # Residual standard error or sigma
> RSE <- round(sqrt( sum(residuals(model.lm)^2) / model.lm$df.residual),2)
> print(paste0("RSE: ", RSE))
[1] "RSE: 1.54"
> 
> # Error Rate
> ErrorRate <- round(sigma(model.lm)/mean(students.data$G3),2)
> print(paste0("ErrorRate: ", ErrorRate))
[1] "ErrorRate: 0.14"
```

R Code for Linear Regression model with significant predictors



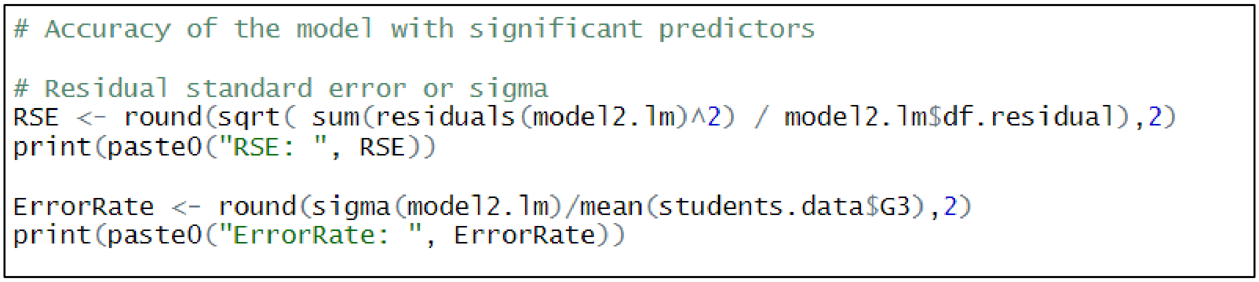
```
# drop insignificant predictors and rerun regression with significant predictors
model2.lm <- lm(G3~traveltime+failures+absences+G1+G2, data=train)
summary(model2.lm)
```

Results for Linear Regression model output with significant predictors



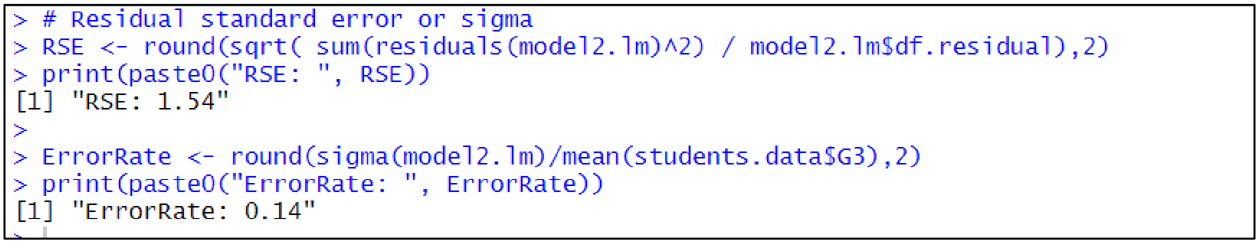
```
> # drop insignificant predictors and rerun regression with significant predictors
> model2.lm <- lm(G3~traveltime+failures+absences+G1+G2, data=train)
> summary(model2.lm)

Call:
lm(formula = G3 ~ traveltime + failures + absences + G1 + G2, 
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.7380 -0.4126  0.0560  0.8049  5.4746 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.124976   0.303152  -3.711 0.000222 ***
traveltime   0.215938   0.079570   2.714 0.006810 ** 
failures    -0.260687   0.095522  -2.729 0.006505 ** 
absences     0.024183   0.009106   2.656 0.008086 ** 
G1           0.139538   0.037434   3.728 0.000208 ***
G2           0.939124   0.033559  27.984  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.541 on 724 degrees of freedom
Multiple R-squared:  0.8386,	Adjusted R-squared:  0.8375 
F-statistic: 752.6 on 5 and 724 DF,  p-value: < 2.2e-16
```

R Code for Linear Regression RSE, Error Rate calculations



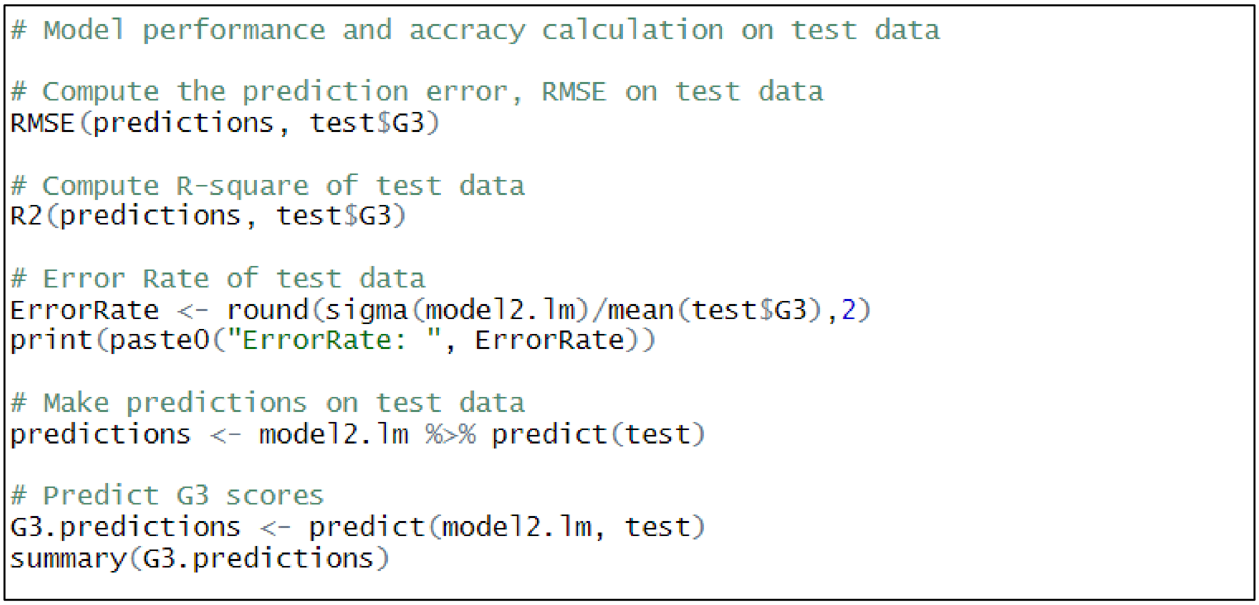
```
# Accuracy of the model with significant predictors

# Residual standard error or sigma
RSE <- round(sqrt( sum(residuals(model2.lm)^2) / model2.lm$df.residual),2)
print(paste0("RSE: ", RSE))

ErrorRate <- round(sigma(model2.lm)/mean(students.data$G3),2)
print(paste0("ErrorRate: ", ErrorRate))
```

Results for Linear Regression RSE, Error Rate calculations



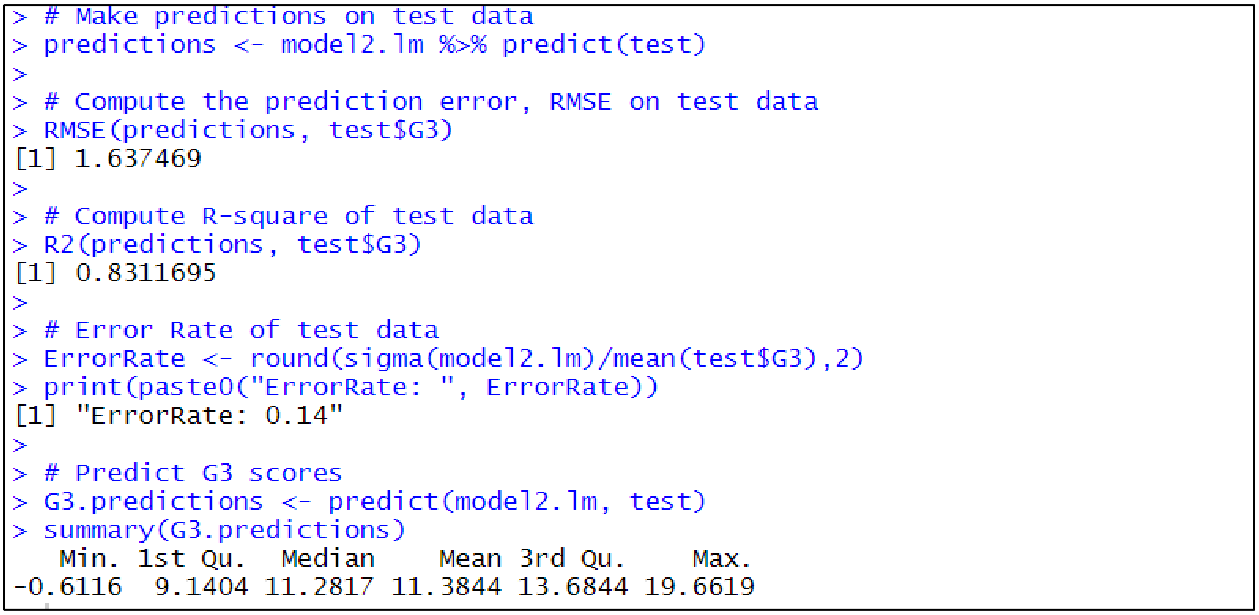
```
> # Residual standard error or sigma
> RSE <- round(sqrt( sum(residuals(model2.lm)^2) / model2.lm$df.residual),2)
> print(paste0("RSE: ", RSE))
[1] "RSE: 1.54"
>
> ErrorRate <- round(sigma(model2.lm)/mean(students.data$G3),2)
> print(paste0("ErrorRate: ", ErrorRate))
[1] "ErrorRate: 0.14"
```

R Code for Linear Regression for performance and accuracy calculation on test data



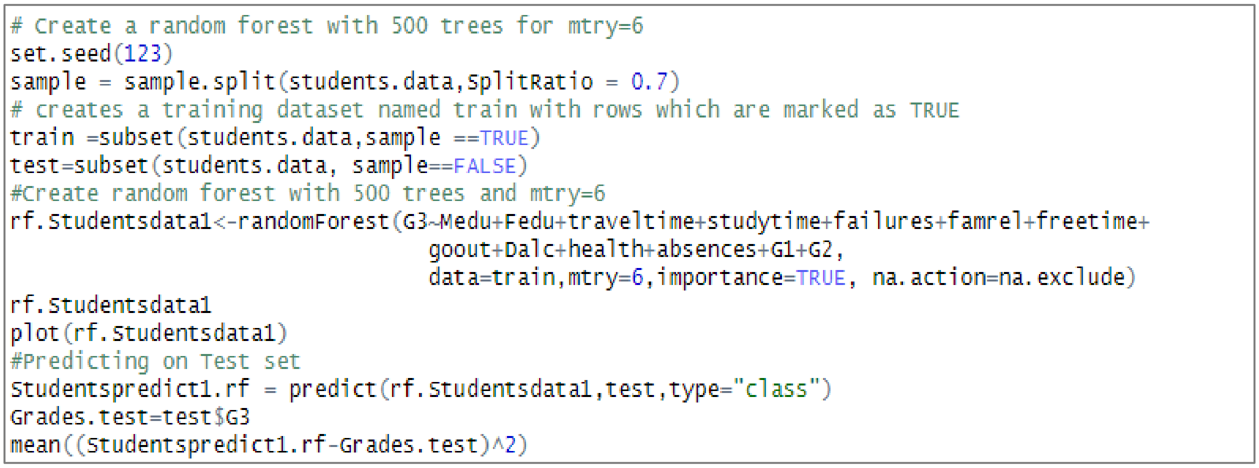
```
# Model performance and accracy calculation on test data

# Compute the prediction error, RMSE on test data
RMSE(predictions, test$G3)

# Compute R-square of test data
R2(predictions, test$G3)

# Error Rate of test data
ErrorRate <- round(sigma(model2.lm)/mean(test$G3),2)
print(paste0("ErrorRate: ", ErrorRate))

# Make predictions on test data
predictions <- model2.lm %>% predict(test)

# Predict G3 scores
G3.predictions <- predict(model2.lm, test)
summary(G3.predictions)
```

Results for Linear Regression for performance and accuracy calculation on test data



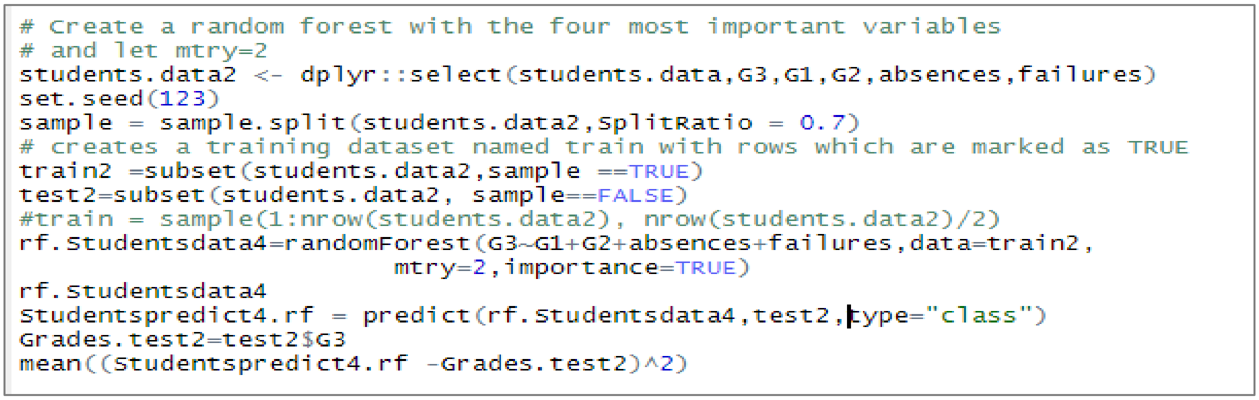
```
> # Make predictions on test data
> predictions <- model2.lm %>% predict(test)
>
> # Compute the prediction error, RMSE on test data
> RMSE(predictions, test$G3)
[1] 1.637469
>
> # Compute R-square of test data
> R2(predictions, test$G3)
[1] 0.8311695
>
> # Error Rate of test data
> ErrorRate <- round(sigma(model2.lm)/mean(test$G3),2)
> print(paste0("ErrorRate: ", ErrorRate))
[1] "ErrorRate: 0.14"
>
> # Predict G3 scores
> G3.predictions <- predict(model2.lm, test)
> summary(G3.predictions)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-0.6116  9.1404 11.2817 11.3844 13.6844 19.6619 
```

R Code for Random Forest Model Creation



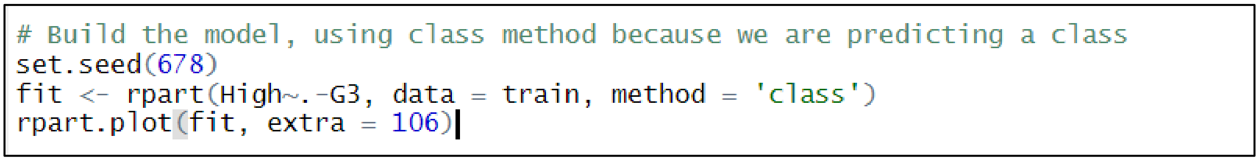
```
# Create a random forest with 500 trees for mtry=6
set.seed(123)
sample = sample.split(students.data,SplitRatio = 0.7)
# creates a training dataset named train with rows which are marked as TRUE
train =subset(students.data,sample ==TRUE)
test=subset(students.data, sample==FALSE)
#Create random forest with 500 trees and mtry=6
rf.Studentsdata1<-randomForest(G3~Medu+Fedu+traveltime+studytime+failures+famrel+freetime+
                                goout+Dalc+health+absences+G1+G2,
                                data=train,mtry=6,importance=TRUE, na.action=na.exclude)
rf.Studentsdata1
plot(rf.Studentsdata1)
#Predicting on Test set
Studentspredict1.rf = predict(rf.Studentsdata1,test,type="class")
Grades.test=test$G3
mean((Studentspredict1.rf-Grades.test)^2)
```

Random Forest with Significant Predictors



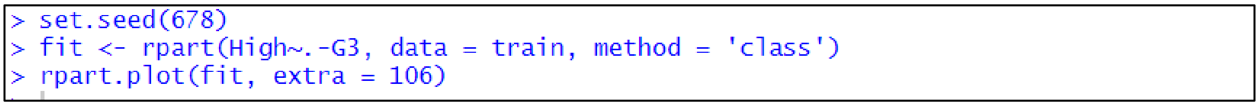
```
# Create a random forest with the four most important variables
# and let mtry=2
students.data2 <- dplyr::select(students.data,G3,G1,G2,absences,failures)
set.seed(123)
sample = sample.split(students.data2,SplitRatio = 0.7)
# creates a training dataset named train with rows which are marked as TRUE
train2 =subset(students.data2,sample ==TRUE)
test2=subset(students.data2, sample==FALSE)
#train = sample(1:nrow(students.data2), nrow(students.data2)/2)
rf.Studentsdata4=randomForest(G3~G1+G2+absences+failures,data=train2,
                        mtry=2,importance=TRUE)
rf.Studentsdata4
Studentspredict4.rf = predict(rf.Studentsdata4,test2,type="class")
Grades.test2=test2$G3
mean((Studentspredict4.rf -Grades.test2)^2)
```

R Code for Classification Tree model creation with all variables



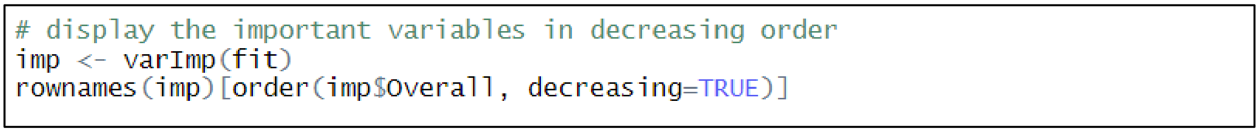
```
# Build the model, using class method because we are predicting a class
set.seed(678)
fit <- rpart(High~.-G3, data = train, method = 'class')
rpart.plot(fit, extra = 106)
```

Results for Classification Tree model creation with all variables



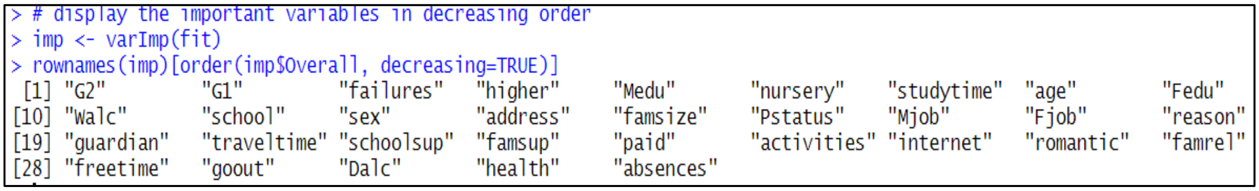
```
> set.seed(678)
> fit <- rpart(High~.-G3, data = train, method = 'class')
> rpart.plot(fit, extra = 106)
```

R Code for Classification Tree model important predictors in decreasing order



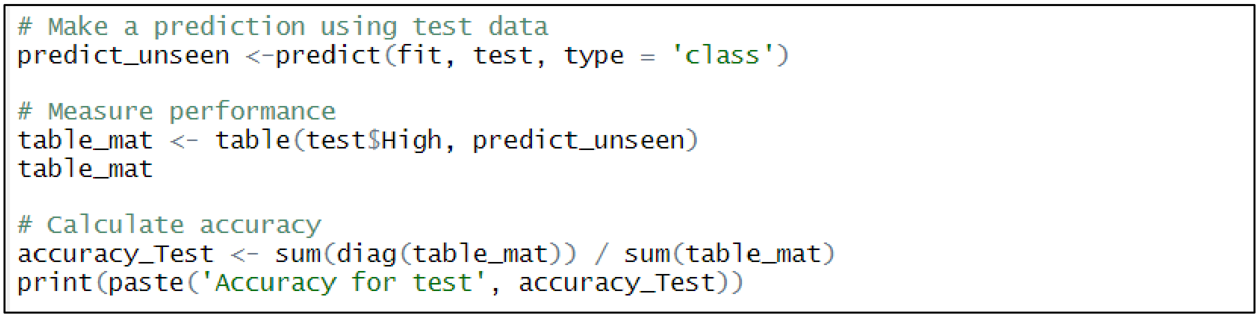
```
# display the important variables in decreasing order
imp <- varImp(fit)
rownames(imp)[order(imp$Overall, decreasing=TRUE)]
```

Results for Classification Tree model important predictors in decreasing order



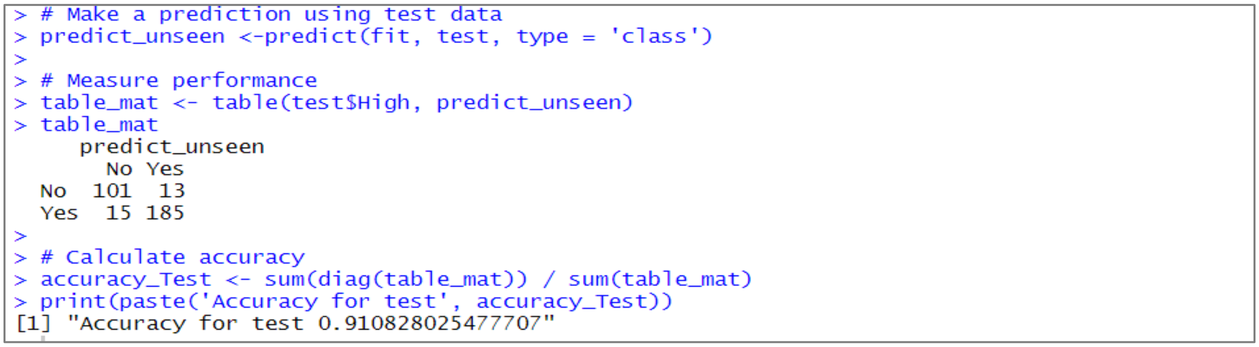
```
> # display the important variables in decreasing order
> imp <- varImp(fit)
> rownames(imp)[order(imp$Overall, decreasing=TRUE)]
 [1] "G2"         "G1"         "failures"   "higher"     "Medu"       "nursery"    "studytime"  "age"        "Fedu"
[10] "Walc"       "school"     "sex"        "address"    "famsize"    "Pstatus"    "Mjob"       "Fjob"       "reason"
[19] "guardian"   "traveltime" "schoolsup"  "famsup"     "paid"       "activities" "internet"   "romantic"   "famrel"
[28] "freetime"   "goout"      "Dalc"       "health"     "absences"
```

R Code for Classification Tree prediction on Test Data, Confusion Matrix and Accuracy calculation



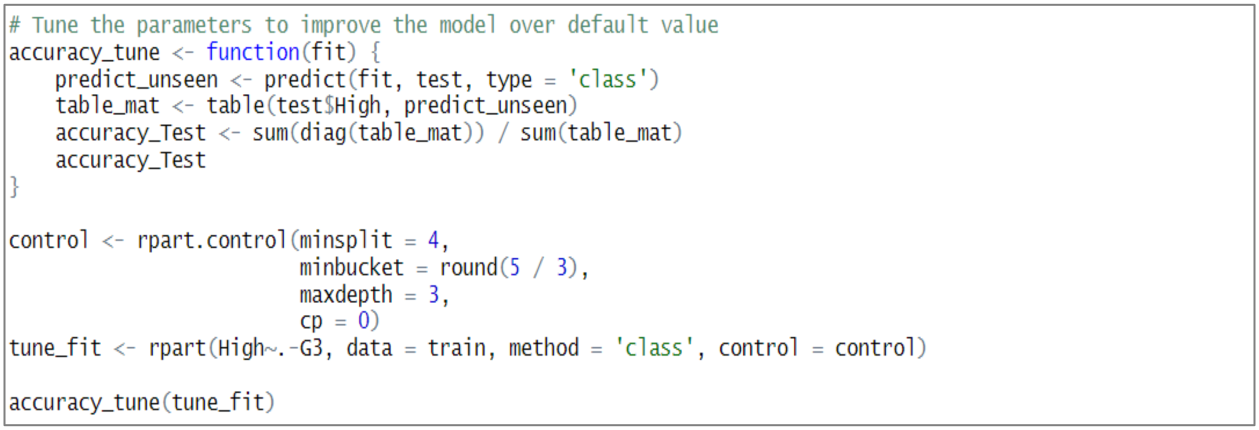
```
# Make a prediction using test data
predict_unseen <-predict(fit, test, type = 'class')

# Measure performance
table_mat <- table(test$High, predict_unseen)
table_mat

# Calculate accuracy
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
print(paste('Accuracy for test', accuracy_Test))
```

Results for Classification Tree prediction on Test Data, Confusion Matrix and Accuracy calculation



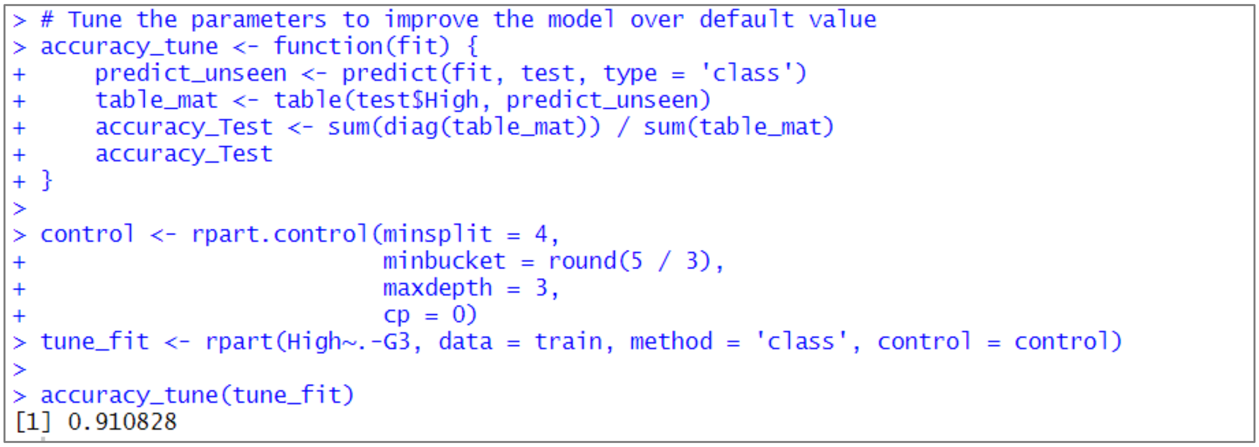
```
> # Make a prediction using test data
> predict_unseen <-predict(fit, test, type = 'class')
>
> # Measure performance
> table_mat <- table(test$High, predict_unseen)
> table_mat
     predict_unseen
       No Yes
  No  101  13
  Yes  15 185
>
> # Calculate accuracy
> accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
> print(paste('Accuracy for test', accuracy_Test))
[1] "Accuracy for test 0.910828025477707"
```

R Code for Tuning the parameters of Classification Tree and tuned Accuracy



```
# Tune the parameters to improve the model over default value
accuracy_tune <- function(fit) {
    predict_unseen <- predict(fit, test, type = 'class')
    table_mat <- table(test$High, predict_unseen)
    accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
    accuracy_Test
}

control <- rpart.control(minsplit = 4,
                         minbucket = round(5 / 3),
                         maxdepth = 3,
                         cp = 0)
tune_fit <- rpart(High~.-G3, data = train, method = 'class', control = control)

accuracy_tune(tune_fit)
```

Results for Tuning the parameters of Classification Tree and tuned Accuracy



```
> # Tune the parameters to improve the model over default value
> accuracy_tune <- function(fit) {
+     predict_unseen <- predict(fit, test, type = 'class')
+     table_mat <- table(test$High, predict_unseen)
+     accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
+     accuracy_Test
+ }
>
> control <- rpart.control(minsplit = 4,
+                          minbucket = round(5 / 3),
+                          maxdepth = 3,
+                          cp = 0)
> tune_fit <- rpart(High~.-G3, data = train, method = 'class', control = control)
>
> accuracy_tune(tune_fit)
[1] 0.910828
```