DSEG6111: Predictive Modeling: What can we use to predict a student's grade?

Nischal Pant

Executive Summary

For this analysis project for Predictive modeling, we have chosen the dataset called student grade which contains data on student achievements in the secondary education of two different Portuguese schools. The data was collected via student report cards and were models under binary/five-level classification and regression tasks. The main task for this analysis was answering two different response problems for predictors, quantitative and qualitative as well as performing a principal component regression problem.

To grasp a basic understanding of the dataset, we needed to first perform some basic data cleaning and analysis. Firstly we performed a multiple regression in which we figured out that the variable famrel (family relationship), absences, G1 (grade for the first period,) and G2 (grade for 2nd period) had the largest correlation for G3. We also performed a subset selection and stepwise selection analysis for the dataset. Other key findings are also shown below and will be discussed in detail in later sections. The main question in this remains, how can we use these models to understand the data set containing G3, and what is essential the best variable for predicting the G3 variable? Using R studio and techniques learned from the textbook “An Introduction to Statistical Learning” we were able to determine G2 is the best predictor for G3 using the random forrest analysis with a mean squared residual of 0.001609.

```
Call:
lm(formula = G3 ~ ., data = student)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.9339 -0.5532  0.2680  0.9689  4.6461 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -1.115488   2.116958  -0.527 0.598573    
famrel            0.356876   0.114124   3.127 0.001912 ** 
absences          0.045879   0.013412   3.421 0.000698 ***
G1                0.188847   0.062373   3.028 0.002645 ** 
G2                0.957330   0.053460  17.907  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.901 on 353 degrees of freedom
Multiple R-squared:  0.8458,     Adjusted R-squared:  0.8279 
F-statistic: 47.21 on 41 and 353 DF,  p-value: < 2.2e-16
```

We also performed models such as Ridge Regression, Lasso Regression, Partial Least Squares, Regression Trees, Bagging, Random Forests and Boosting. The output for these models, the signification of the outputs, and factors affecting the models will discussed in later sections.

To perform Qualitative response analysis we create a new dataset called pass or fail, in which the entries for G3 (final grade) is converted to 1 = pass and 0 = fail. Then we perform different tests to see the results. Tests such as KNN, Logistic Regression, LDA, QDA, Classification Trees, Bagging, and Random Forests analysis are done. The output for these models, the signification of the outputs, and factors affecting the models will discussed in later sections.

And finally, we will be performing a Principal component regression for the dataset. For this, we will be simplifying the dataset and looking at key parts of the dataset. The main goal is handling high-dimensional data and reducing dimensionality to see which predictors are suitable in the dataset.

Data & Approach: An overview of the data used in your analysis, any data re-engineering and related steps, a brief discussion of the overall approach, analytic goals, and data analytic techniques utilized.

To first perform the analysis we must understand our data first. We have different numerical and categorical variables. The names of the variables are listed below.

- school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira).
- sex - student's sex (binary: 'F' - female or 'M' - male).
- age - student's age (numeric: from 15 to 22).
- address - student's home address type (binary: 'U' - urban or 'R' - rural).
- famsize - family size (binary: 'LE3' - less than or equal to 3 or 'GT3' - greater than 3).
- Pstatus- parent's cohabitation status (binary: 'T' - living together or 'A' - apart).
- Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education).
- Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education).
- Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other').
- Fjob- father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other').
- reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other').
- guardian - student's guardian (nominal: 'mother', 'father' or 'other').
- 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).

- studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours).
- failures - number of past class failures (numeric: n if 0 <= n < 3, else 3).
- schoolsup - extra educational support (binary: yes or no).
- famsup - family educational support (binary: yes or no).
- paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no).
- activities - extra-curricular activities (binary: yes or no).
- nursery - attended nursery school (binary: yes or no).
- higher - wants to take higher education (binary: yes or no).
- internet - Internet access at home (binary: yes or no).
- romantic - with a romantic relationship (binary: yes or no).
- famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent).
- freetime - free time after school (numeric: from 1 - very low to 5 - very high).
- goout - going out with friends (numeric: from 1 - very low to 5 - very high).
- Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high).
- Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high).
- health - current health status (numeric: from 1 - very bad to 5 - very good).
- absences - number of school absences (numeric: from 0 to 93).
-
- Grades which are related with the course subject:
- G1 - first period grade (numeric: from 0 to 20).
- G2 - second period grade (numeric: from 0 to 20)
- G3 - final grade (numeric: from 0 to 20, Output Target)

The dataset did not require any cleaning as it was already organized with no missing entries. All the variables were either integers or characters. This makes the analysis very easy. We then performed the following tests (models and their approach and goals are described below:

Quantitative Response Variable Models

1. Multiple Regression Models, including best subset selection and stepwise selection
    - The purpose of this model was to understand the correlation between the variables in the dataset. We quantified the relationships between the different variables and were given the estimates for G3. In this, we used the `lm(formula = G3 ~ ., data = student)`. It also provided us with the Residual standard error, Multiple R-squared, and the F-statistic.
    - For the Best subset selection, we wanted to identify which subset of predictor variables in conjunction would be the best and most accurate group for modeling. regsubsets(G3 ~ ., data = student). For this, we got a subset of variables through multiple steps of computation and provided multiple combinations.
    - For the Stepwise selection we try to find the same combinations but this is done through adding or removing variables. For this, we used the code stepAIC(lm(G3 ~ ., data = student), direction = "both")

2. Ridge Regression: In this technique, we check to see the less precise outcomes between over-correlating models, this might create instability and inflated standard errors for the coefficient. It also helps control the trade-off between fitting the data well and keeping the coefficients small. The use of cross-validation was also applied and then a final model with the selected lambda was fitted to the entire dataset. It used all the numerical variables from the dataset.
    - y <- student$G3
    - ridge_model <- cv.glmnet(x = X, y = y, alpha = 0)
    - plot(ridge_model)
    - best_lambda <- ridge_model$lambda.min
    - final_model <- glmnet(x = X, y = y, alpha = 0, lambda = best_lambda)
    - ridge_predictions <- predict(final_model, newx = X, s = best_lambda)
    - mse <- mean((ridge_predictions - y)^2)
    - print(paste("Mean Squared Error:", mse))

3. Lasso Regression: The lasso regression also known as Least Absolute Shrinkage and Selection Operator is very similar to the Ridge regression technique but in this method, the model penalizes over-fitting and causes the model to shrink less important coefficients towards zero and allows effective selection of variables for the model.
    - X <- as.matrix(student[, c("studytime", "failures", "absences", "G1", "G2")])
    - y <- student$G3
    - lasso_model <- cv.glmnet(x = X, y = y, alpha = 1)
    - plot(lasso_model)
    - best_lambda <- lasso_model$lambda.min
    - final_lasso_model <- glmnet(x = X, y = y, alpha = 1, lambda = best_lambda)
    - coef(final_lasso_model)
    - lasso_predictions <- predict(final_lasso_model, newx = X, s = best_lambda)
    - lasso_mse <- mean((lasso_predictions - y)^2)
    - print(paste("Mean Squared Error (Lasso):", lasso_mse))
    -
4. Partial Least Squares: This method combines dimension reduction and regression, this is also to combat multicollinearity. PLS allows for the simultaneous modeling of the response variable and the predictor variables. The model does so by transforming the original predictors into a small subset of uncorrelated components which results in molding a more parsimonious model.
    - library(pls)
    - set.seed(123)
    - X <- as.matrix(student[, c("studytime", "failures", "absences", "G1", "G2")])
    - y <- student$G3

   - pls_model <- plsr(y ~ X, ncomp = 2)
   - summary(pls_model)
   - pls_predictions <- predict(pls_model, newdata = X)
   - pls_mse <- mean((pls_predictions - y)^2)
   - print(paste("Mean Squared Error (PLS):", pls_mse))

5. Regression Trees: For this section, we model non-linear patterns between our response variable (G3) and our predictor variables which is also compression. This type of decision tree gives us the probability of relationships and splits to show relationships. We used the Analysis of Variance (ANOVA) method for this regression tree and here is the code:
   - set.seed(123)
   - X <- student[, c("studytime", "failures", "absences", "G1", "G2")]
   - y <- student$G3
   - tree_model <- rpart(y ~ ., data = data.frame(X, y), method = "anova")
   - plot(tree_model)
   - text(tree_model)
   - tree_predictions <- predict(tree_model, newdata = data.frame(X))
   - tree_mse <- mean((tree_predictions - y)^2)
   - print(paste("Mean Squared Error (Decision Tree):", tree_mse))

6. Bagging: This model allows a balance between variance and bias. By creating test and train data we can eliminate fluctuation or noise in dataset. This also allows for the removal of outliers. This data is very high dimensional so creating diverse subsets of features prevents the overfitting of models.
   - set.seed(123)
   - X <- student[, c("studytime", "failures", "absences", "G1", "G2")]
   - y <- student$G3
   - bagging_model <- randomForest(x = X, y = y, ntree = 100)
   - bagging_predictions <- predict(bagging_model, newdata = data.frame(X))
   - bagging_mse <- mean((bagging_predictions - y)^2)
   - print(paste("Mean Squared Error (Bagging):", bagging_mse))

7. Random Forests: The technique used in this model is subsampling and aggregating the predictions of multiple random trees. The output gives us the number of trees created and the no. of variables tried at each split as well as the mean of squared residuals.
   - formula <- as.formula("pass_fail ~ .")
   - rf_model <- randomForest(formula, data = student)
   - print(rf_model)
   - rf_predictions <- predict(rf_model, student)
   - print(rf_predictions)

8. Boosting: The model utilizes combining predictions of multiple simple models and backtracking by removing errors. If a model performs poorly (has a large error) then model the next by training it.
    - student_factorized <- student
    - categorical_columns <- sapply(student_factorized, is.character)
    - student_factorized[categorical_columns] <- lapply(student_factorized[categorical_columns], as.factor)
    - formula <- as.formula("G3 ~ .")
    - boost_model <- gbm(formula, data = student_factorized, distribution = "gaussian", n.trees = 100, interaction.depth = 3)
    - boost_predictions <- predict(boost_model, newdata = student_factorized, n.trees = 100)
    - print(boost_model)

Qualitative Response Variable Models

- KNN: Also known as the K- Nearest neighbor this model allows us to fit a regression model based on the distance between the nearest points of its neighbor. It gives us a value based on the average and weighted average of the target value of its k-nearest neighbor. The smaller the value, the closer the points and the less variance.
    - categorical_vars <- c("school", "sex", "address", "famsize", "Pstatus", "Mjob", "Fjob", "reason", "guardian",
        - "schoolsup", "famsup", "paid", "activities", "nursery", "higher", "internet", "romantic")
    - X <- model.matrix(~ . - 1, data = student[, c("studytime", "failures", "absences", "G1", "G2", categorical_vars)])
    - y <- as.numeric(as.character(student$G3))
    - set.seed(123) # for reproducibility
    - train_indices <- sample(1:nrow(student), 0.8 * nrow(student))
    - train_data <- student[train_indices,]
    - test_data <- student[-train_indices,]
    - knn_model <- train(
    -   x = X[train_indices,],
    -   y = y[train_indices],
    -   method = "knn",
    -   trControl = trainControl(method = "cv", number = 5),
    -   tuneGrid = expand.grid(k = 5) # Set the number of neighbors (k) to 5
    -)

  - knn_predictions <- predict(knn_model, newdata = X[-train_indices, , drop = FALSE])
  - mse <- mean((knn_predictions - y[-train_indices])^2)
  - print(paste("Mean Squared Error (KNN):", mse))

- Logistic Regression: This model allows us to predict the probability of a binary outcome. For this dataset, it is whether the student will pass or fail. 0 or 1. Logistic regression models the relationship between the independent variables and the log-odds of the probability of the positive class.

  - logistic_model <- glm(pass_fail ~ ., data = train_data, family = "binomial")
  - logistic_predictions <- predict(logistic_model, newdata = test_data, type = "response")
  - logistic_predictions_binary <- ifelse(logistic_predictions > 0.5, 1, 0)
  - conf_matrix_logistic <- confusionMatrix(logistic_predictions_binary, test_data$pass_fail)
  - print(conf_matrix_logistic)

- LDA: Also known as Linear Discriminant Analysis is a dimensionality reduction and classification technique that aims to maximize the distance between the means of different classes while minimizing the variance of each class. The matrix that is outputed from the code below is the covariance structure of the data to determine the direction that best discriminate between classes.
  - lda_model <- lda(pass_fail ~ ., data = train_data)
  - lda_predictions <- predict(lda_model, newdata = test_data)
  - conf_matrix_lda <- table(lda_predictions$class, test_data$pass_fail)
  - print(conf_matrix_lda)

- QDA: while being the same as the LDA the Quadratic Discriminat Analysis does not assume that all classes share a common covariance matric and allows each class to have its own covariance matrix.
  - qda_model <- qda(pass_fail ~ ., data = train_data)
  - qda_predictions <- predict(qda_model, newdata = test_data)
  - conf_matrix_qda <- table(qda_predictions$class, test_data$pass_fail)
  - print(conf_matrix_qda)

- Classification Trees: This technique is very simple and creates a decision tree at which each internal node represents a binary decision based on the value of a specific feature.It is as simple as a flow chart.
    - tree_model <- tree(pass_fail ~ ., data = train_data)
    - tree_predictions <- predict(tree_model, newdata = test_data)
    - conf_matrix_tree <- table(tree_predictions, test_data$pass_fail)
    - print(conf_matrix_tree)

- Bagging or Bootstrap Aggregation is a technique that reduces variance and reduces overfitting. This technique involves training multiple instances of the same base bodel on different subset of the training data.
    - bagging_model <- randomForest(x = train_data[, -which(names(train_data) == "pass_fail")],
    - y = train_data$pass_fail,
    - ntree = 100)
    - bagging_predictions <- predict(bagging_model, newdata = test_data)
    - conf_matrix_bagging <- table(bagging_predictions, test_data$pass_fail)
    - print(conf_matrix_bagging)

- Random Forests
    - rf_model <- randomForest(x = train_data[, -which(names(train_data) == "pass_fail")],
    - y = train_data$pass_fail,
    - ntree = 100)
    - rf_predictions <- predict(rf_model, newdata = test_data)l
    - conf_matrix_rf <- table(rf_predictions, test_data$pass_fail)
    - print(conf_matrix_rf)

Principal Components Regression: In conclusion, principle Components Regression (PCR) reduces the original collection of correlated predictors into a smaller set of uncorrelated principle components in order to solve multicollinearity in regression models. Principal component analysis (PCA) reduces dimensionality while maintaining the interpretability of linear regression; this is achieved by PCR. When dealing with multicollinearity or high-dimensional data, it is especially helpful.

- pcr_model <- pcr(pass_fail ~ ., data = student, scale = TRUE)
- pcr_predictions <- as.vector(predict(pcr_model, newdata = student)$calibrate$calX[, 1])
- pcr_predictions_binary <- ifelse(pcr_predictions > 0.5, 1, 0)

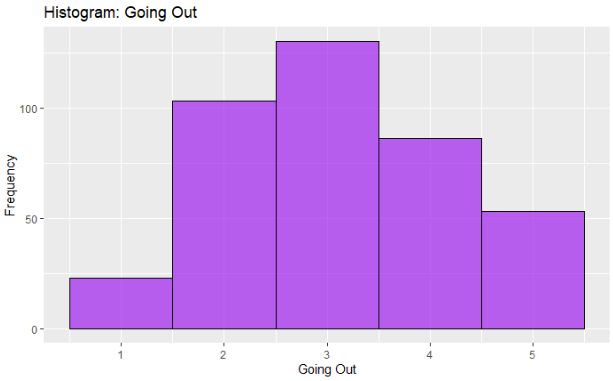
- conf_matrix_pcr <- table(pcr_predictions_binary, student$pass_fail)
- print(conf_matrix_pcr)

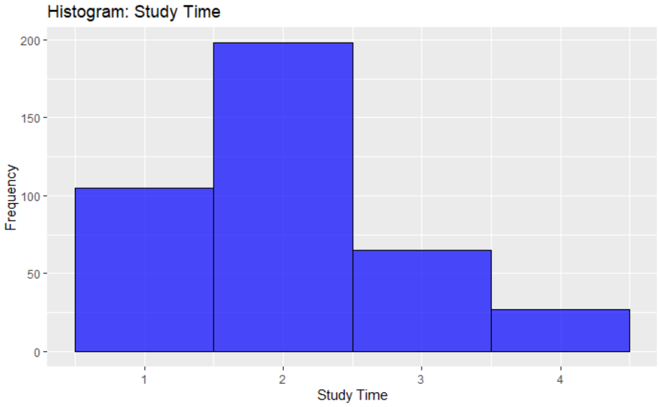
Findings

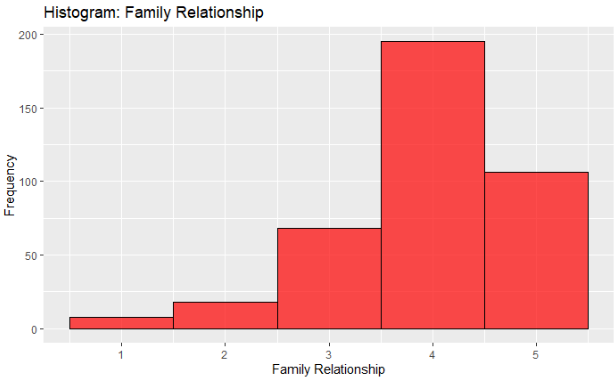
When we first analyse the dataset, it is essential that we understand what the dataset actually looks like. It

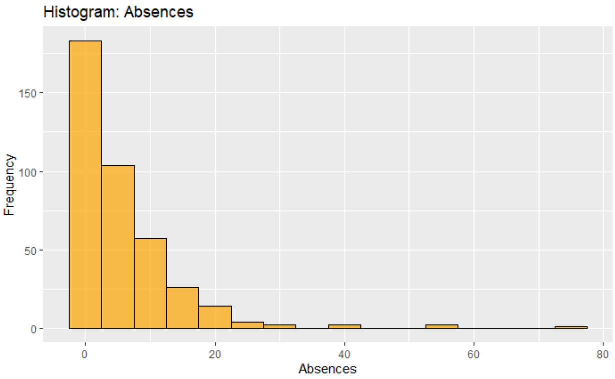
```
'data.frame':	395 obs. of  33 variables:
 $ school    : chr  "GP" "GP" "GP" "GP" ...
 $ sex       : chr  "F" "F" "F" "F" ...
 $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
 $ address   : chr  "U" "U" "U" "U" ...
 $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
 $ Pstatus   : chr  "A" "T" "T"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

The data frame has 395 observations and 33 variables which has almost an even distribution of numerical and categorical variables. We first have to see which ones we can use for our analysis. The good thing about this dataset was that it had levels for categorical datasets which had 4 to 5 levels but that caused issues in later analysis thus some models had error and did not have an output. Some provided for great output which will help us lead to understanding the relationship between the variables even more.









Regression model:

```
Call:
lm(formula = G3 ~ ., data = student)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.9339 -0.5532  0.2680  0.9689  4.6461 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -1.115488   2.116958  -0.527 0.598573    
famrel            0.356876   0.114124   3.127 0.001912 ** 
absences          0.045879   0.013412   3.421 0.000698 ***
G1                0.188847   0.062373   3.028 0.002645 ** 
G2                0.957330   0.053460  17.907  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.901 on 353 degrees of freedom
Multiple R-squared:  0.8458,     Adjusted R-squared:  0.8279 
F-statistic: 47.21 on 41 and 353 DF,  p-value: < 2.2e-16
```

The output for the regression model shows us the intercept, and the correlation estimate for the variables and G3 the final grade. Famrel, absences, G1 and G2. G2 and absences had the largest numbers and it would make sense that since absent students miss the couse and your midterm grades would also determine your final grade. The standard error of the residual was 1.901 which is low. The F-statistic tests the overall significance of the model. A large F-statistic and a small p-value suggest that at least one predictor variable is significantly related to the response variable. The R-squared value of 0.8458 indicated the variance in the response variable of G3 which is explained by the predictor variables at 84%. These results show that there is a great correlation and and response can be explained by the variables.

SUBSET

```
       goout Dalc Walc health absences G1  G2  pass_fail
1  ( 1 ) " "   " "  " "  " "    " "      " " "*" " "
2  ( 1 ) " "   " "  " "  " "    " "      " " "*" "*"
3  ( 1 ) " "   " "  " "  " "    "*"      " " "*" "*"
4  ( 1 ) " "   " "  " "  " "    "*"      " " "*" "*"
5  ( 1 ) " "   " "  " "  " "    "*"      " " "*" "*"
6  ( 1 ) " "   " "  " "  " "    "*"      "*" "*" "*"
7  ( 1 ) " "   " "  " "  " "    "*"      "*" "*" "*"
8  ( 1 ) " "   " "  " "  " "    "*"      "*" "*" "*"
```

To pick the optimum subset for modeling, we needed to determine which subset of predictor variables together would make the most accurate and optimal group. And our output shows that absences G1, and G2 determined G3. In this output I used the G3 variables to create a pass-fail variable for binary analysis We obtained a subset of the variables for this by doing many computing stages, and we offered several combinations. This falls together into our results from the regression model.

STEPWISE

```
lm(formula = G3 ~ age + activities + romantic + famrel + goout +
    absences + G1 + G2 + pass_fail, data = student)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6944 -0.4631  0.1919  0.9692  4.7357
```

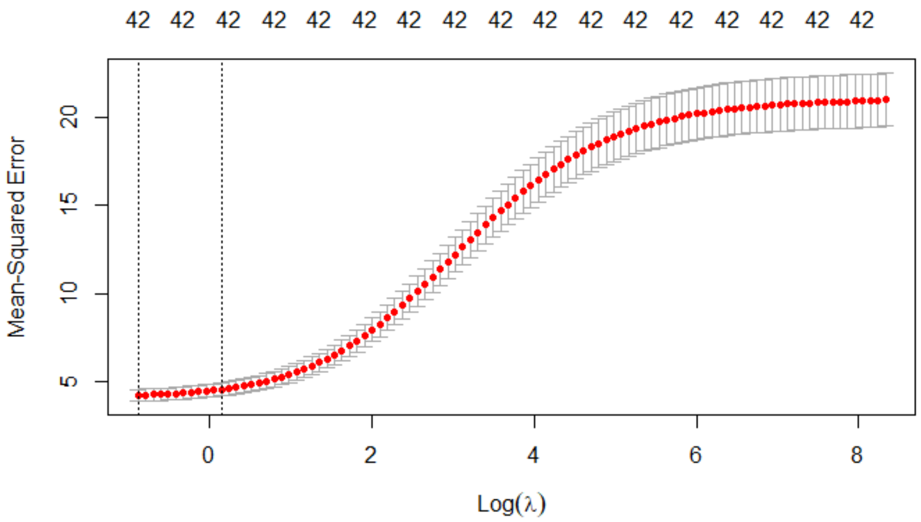
```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -0.20872    1.28625  -0.162  0.87118    
famrel          0.27705    0.09857   2.811  0.00519 ** 
absences        0.05279    0.01120   4.713 3.41e-06 ***
G1              0.11481    0.05141   2.233  0.02610 *  
G2              0.81072    0.04934  16.432  < 2e-16 ***
pass_fail       2.27183    0.27665   8.212 3.33e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.729 on 385 degrees of freedom
Multiple R-squared:  0.8609,	Adjusted R-squared:  0.8576 
F-statistic: 264.7 on 9 and 385 DF,  p-value: < 2.2e-16
```

Our stepwise regression also gave us the same variables and the standard error of only 1.729 which is extremely low, with a Multiple R-squared, the significance is very large at 264.7 with a very small p-value.

RIDGE REGRESSION:

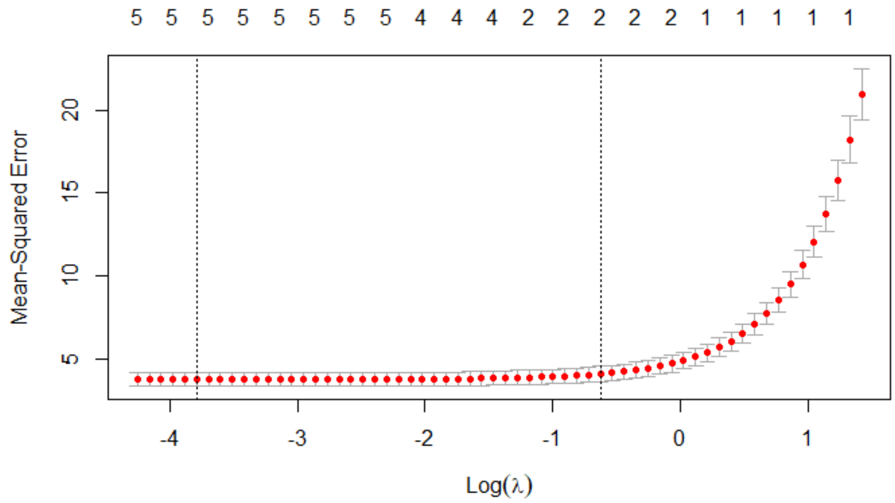
```
[1] "Mean Squared Error: 3.37476386480592"
```



Using this method, we look for less accurate results between models that have excessive correlation; this could lead to instability and exaggerated coefficient standard errors. Additionally, it aids in managing the trade-off between maintaining tiny coefficients and a good fit to the data. After applying cross-validation, the chosen lambda was used to fit the final model to the whole dataset. Every numerical variable in the dataset was utilized. The mean squared error (MSE) of 3.3747, and the ridge regression model's squared difference between the actual and predicted value.

To determine how effectively our ridge regression model is working, we need to compare its MSE to those of other models or models. Generally speaking, a lower MSE indicated greater predictive performance.

LASSO REGRESSION:



Similar to the Ridge regression technique, lasso regression, also called Least Absolute Shrinkage and Selection the model penalizes over-fitting, causes the model to shrink less significant coefficients towards zero, and enables efficient variable selection for the model. We can now compare our Lasso regression with the ridge regression which has a Mean squared error of 3.588 which is almost the same.

```
6 x 1 sparse Matrix of class "dgCMatrix"
                     s0
(Intercept) -1.43947280
studytime   -0.14476299
failures    -0.25479384
absences     0.03377458
G1           0.14679647
G2           0.97445801
[1] "Mean Squared Error (Lasso): 3.58830473585048"
```

PARTIAL LEAST SQUARES

```
Data:  X dimension: 395 5 
      Y dimension: 395 1
Fit method: kernelpls
Number of components considered: 2
TRAINING: % variance explained
```

```
   1 comps  2 comps
X    25.97    95.36
y    80.49    80.74
[1] "Mean Squared Error (PLS): 4.05878617401252"
```

Regression and dimension reduction are combined in this strategy to counter multicollinearity. PLS makes it possible to model the predictor and response variables at the same time. To create a more parsimonious model, the model converts the original predictors into a limited collection of uncorrelated components. This now gives us a slightly higher MSE of 4.058.

BAGGING:

```
[1] "Mean Squared Error (Bagging): 2.8929978430041"
```

RANDOM FORREST

```
Call:
 randomForest(formula = formula, data = student)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 11

          Mean of squared residuals: 0.001609379
                    % Var explained: 99.27
```

This model employs the subsampling and aggregation of several random trees' predictions. The output provides us with the mean of squared residuals, the number of trees constructed, and the number of variables attempted at each split. The best result possible out of any of our models is the random forest regression for our numerical variable with a mean squared residual of 0.001609 which is substantially smaller than any of the other models. This is what the best model would to predict which quantitative indicator can be used to determine g3.

BOOSTING:

```
gbm(formula = formula, distribution = "gaussian", data = student_factorized,
    n.trees = 100, interaction.depth = 3)
A gradient boosted model with gaussian loss function.
100 iterations were performed.
There were 33 predictors of which 29 had non-zero influence.
```

KNN

```
[1] "Mean Squared Error (KNN): 4.47893774059084"
```

LDA:

```
   0  1
0 22  4
1 11 42
```

QDA:

```
   0  1
0 18  3
1 15 43
```

CLASSIFICATION TREE:

```
Warning: NAs introduced by coercionWarning: NAs introduced by coercion
tree_predictions  0  1
               0 33  0
               1  0 46
```

Appendix:

Libraries used in R

- library(leaps)
- library(ggplot2)
- library(glmnet)
- library(pls)
- library(tree)
- library(randomForest)
- library(gbm)
- library(class)
- library(MASS)
- library(tree)
- library(randomForest)
- library(caret)

Dataset used:
https://www.kaggle.com/code/othmanshbeir/student-grade-prediction-accuracy-95-97/notebook

Textbook used:
https://www.statlearning.com/ ISL with R version 2