Student Performance Analysis

Wenxin Zhong

30 January, 2022

Abstract

> Statistical learning methods were applied to student performance data in order to test attributions of student's achievement. A variety of learning techniques were explored and validated. Simple methods like logistic regression show promise, especially given their computational efficiency at test time.

Introduction

Parents and educators are always concerned with their children's performance at school since the school's performance is related to children's development in the long run. Children's grade at school is an important aspect to show children's performance because the grade is based on how much effort the student put into and their willingness to learn new things. Also, a good grade can motivate the children to build confidence. For this reason, children's grade is a significant aspect when research children's performance.

From all the courses the children learned at school, the grade of math is especially a big concern for many parents because Mathematics grades can show the understanding, computing, applying, reasoning and engaging ability of a child. These five features are also interdependent with each other. We are also interested in mathematics performance because students throughout the world take math courses, we think studying on the math grade is more representative and meaningful to understand students' performance.

Method

Data

The dataset is found on UCI Machine Learning [1] and is gathered by Paulo Cortez from the University of Minho, GuimarÃ£es, Portugal. This dataset contains 33 attributes, including student's grade(G1 for the first period, G2 for the second period, G3 for the final grade), demographic, social and many school-related features. The data was collected using school reports and questionnaires.

Some exploratory data analysis can be found in the appendix.

Modeling

In order to detect the students' achievement, several classification strategies were explored. Both multiclass models and binary models were considered into use. we use following modeling strategies:

- k-Nearest Neighbour model, through the use of `caret` package.
- Random Forests, though the use of the `ranger` package. (The `ranger` packages implements random forests, as well as extremely randomized trees. The difference is considered a tuning parameter.)

-Boosted Model, using training data from "caret" package, and Stochastic Gradient Boosting through the use of the gbm method.

-logistic regression with a lasso regression, we created a matrix for use with cv.glmnet() function to fit a logistic regression with alpha=1.

-logistic regression with a ridge regression, we created a matrix for use with cv.glmnet() function to fit a logistic regression with alpha=0.

Evaluation

All models were tuned using 10-fold cross-validation through the use of the `caret` package. Multiclass models and binary models were both tuned for accuracy.

Models were ultimately evaluated based on their ability to predict the students' math grade level. Compared with multiclass models and binary models, all binary models have higher accuracy than multiclass models. Thus, binary models is better than multiclass models in evaluating the students' math grade in real life.

Multiclass Classification

```
cv_multi = trainControl(method = "cv", number = 10)
```

```
#knn model for multiclass classification
set.seed(42)
fit_multiclass_knn = train(
  G3_letter ~ . - G3 - G3_bin,
  data = stumath_trn,
  method = "knn",
  trControl = trainControl(method = "cv", number = 10)
)
```

```
# random forest model for multiclass classification
set.seed(42)
fit_multiclass_rf = train(
  G3_letter ~ . - G3 - G3_bin,
  data = stumath_trn,
  method = "ranger",
  trControl = trainControl(method = "cv", number = 10),
  verbose = FALSE
)
```

Binary Classification

```
#boosted model for binary classification
set.seed(42)
fit_bin_gbm = train(
  form = G3_bin ~ . - G3_letter - G3,
  data = stumath_trn,
  method = "gbm",
  trControl = trainControl(
    method = "cv",
    number = 10,
    classProbs = TRUE,
    summaryFunction = twoClassSummary
  ),
  metric = "Sens",
  verbose = FALSE
)
```

```
#random forest for binary classification
set.seed(42)
fit_bin_rf = randomForest(
  G3_bin ~ . - G3 - G3_letter,
  data = stumath_trn,
  mtry = 10,
  ntree = 200
)
```

```
#logistic regression with a lasso regression penalty
set.seed(42)
fit_glmnet_lasso = cv.glmnet(
  math_trn_x_bin,
  stumath_trn$G3_bin,
  nfolds = 10,
  alpha = 1 ,
  family = "binomial"
)
```

```
#logistic regression with a ridge regression penalty
set.seed(42)
fit_glmnet_ridge = cv.glmnet(
  math_trn_x_bin,
  stumath_trn$G3_bin,
  nfolds = 10,
  alpha = 0 ,
  family = "binomial"
)
```

Result

Based on the results of the final grade letter table, the result matches practical case. Students who have F in our calculated method predict as fail.

We use accuracy to find the best model. Acoording to accuracy, all binary models have high accuracy, and the best model is the random Forest model with binary classification since it has the highest accuracy.

Table: **Multiclass KNN Model**, Cross-Validated Binary Predictions versus Multiclass Response, Percent

	Final Grade Letter					
	A	B	C	D	E	F
Predict: Fail	0.000	0.000	0.000	0.508	4.061	18.782
Predict: Pass	4.569	6.599	16.751	18.274	21.827	8.629

Table: **Multiclass Random Forest**, Cross-Validated Binary Predictions versus Multiclass Response, Percent

	Final Grade Letter					
	A	B	C	D	E	F
Predict: Fail	0.000	0.000	0.000	0.000	5.076	23.858
Predict: Pass	4.569	6.599	16.751	18.782	20.812	3.553

Cross-Validation

Table: **Binary Logistic Regression**, Cross-Validated Binary Predictions versus Multiclass Response, Percent

	True Number of Valves					
	A	B	C	D	E	F
Predict: Fail	0.000	0.000	0.000	0.000	0.000	27.411
Predict: Pass	4.569	6.599	16.751	18.782	25.888	0.000

Accuracy Comparison

Table:Accuracy Comparison

Model	Accuracy
KNN Classification Model	0.631
Random Forest Classification Model	0.778
GBM Binary Model	0.914
Random Forest Binary Model	0.919
Logistic Regression With a Lasso Regression Penalty	0.919
Logistic Regression With a Ridge Regression Penalty	0.884

Discussion

The results show promise, the accuracy for binary model using `randomForest` method is high and shows its reliability. The below table also summarizes the results of the chosen model on a held-out test dataset. The output valus are still ideal, which means our models can be applicable to reality.

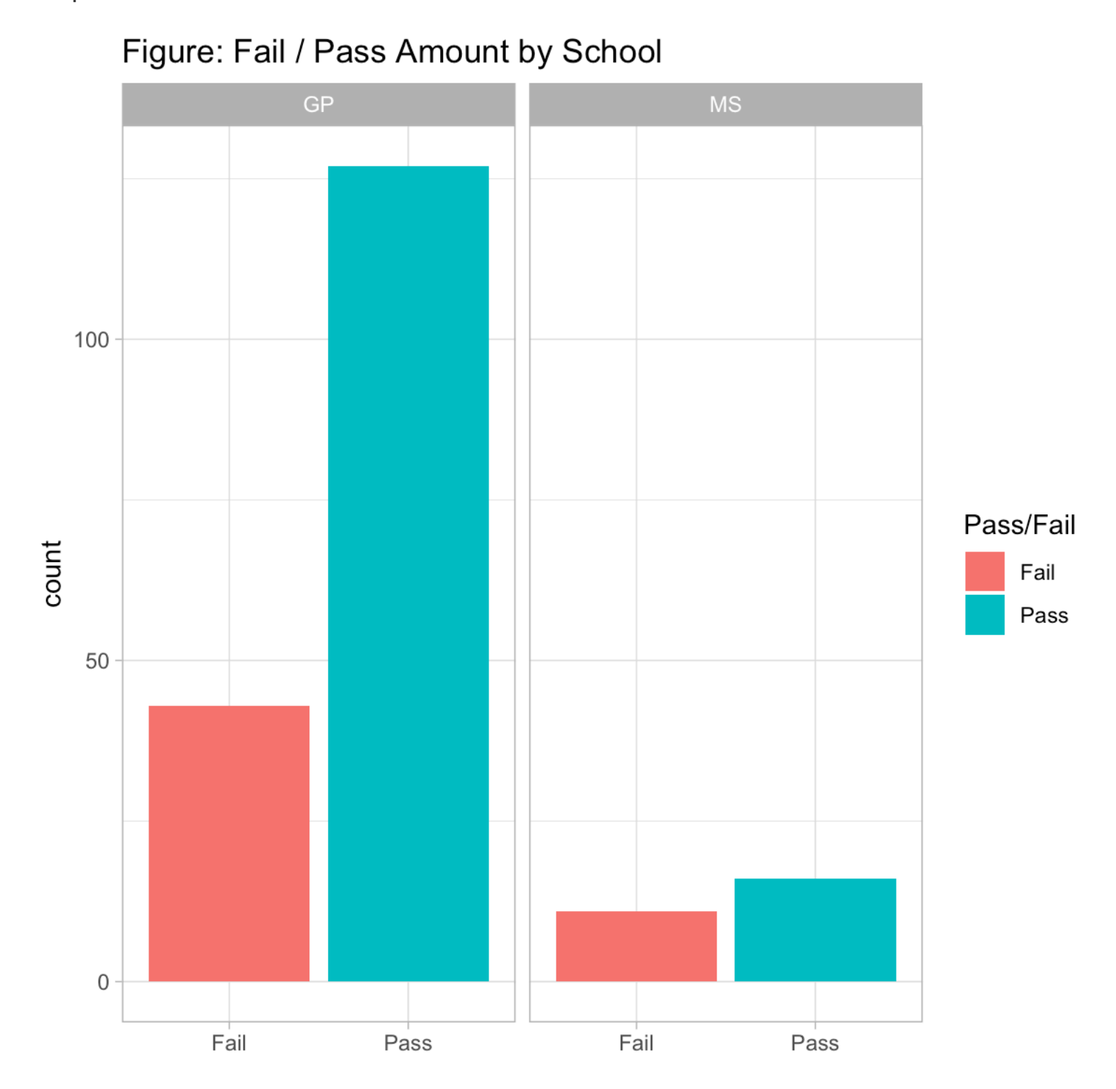
Table: Test Results, **Binary RandomForest Model**, Percent

	True Number of Valves					
	A	B	C	D	E	F
Predict: Fail	0.000	0.000	0.000	0.000	4.545	34.848
Predict: Pass	4.545	4.545	13.636	12.626	21.717	3.535

Also, the model shows that students' math grade is highly related to factors we used, including school, sex, age mother and father's education, studytime, number of past class failures,extra educational support,family educational support and so on, and such many variables is one reason why our accuracy is high. In real life, there are many things can affect students' achievement at school, and that is the same as what we analysis in our project. So, it is not an easy thing if one want to improve his/her performance at school, because he/she need to put more efforts in study and change his/hers study habits.

Despite the somewhat promising result, some serious issues occurred with this dataset. Firstly, there are problems with the sampling procedure used to collect the data. More data from school Gabriel Pereira than Mousinho da Silveira was collected in the dataset. This issue would be problematic since there are definitely existing confounders between different schools, such as teaching style or school type (public or private). In addition, the data was collected specifically from Portugal. Using the model outside the nation might also result in terrible extrapolation.

Additional analysis based on updated data collection is recommended.



Appendix

Data Dictionary

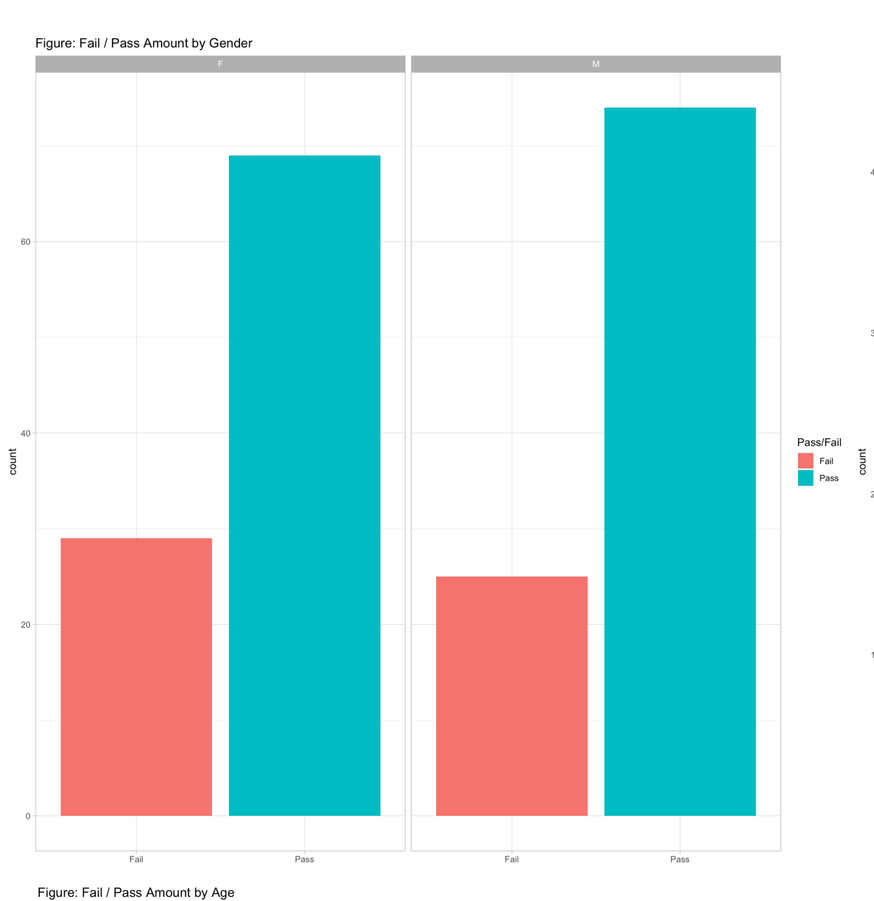
- `school` - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
- `sex` - student's sex (binary: 'F' - female or 'M' - male)
- `age` - student's age (numeric: from 15 to 22)
- `address` - student's home address type (binary: 'U' - urban or 'R' - rural)
- `Medu` - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
- `Fedu` - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
- `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 1<=n<3, else 4)
- `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)
- `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)

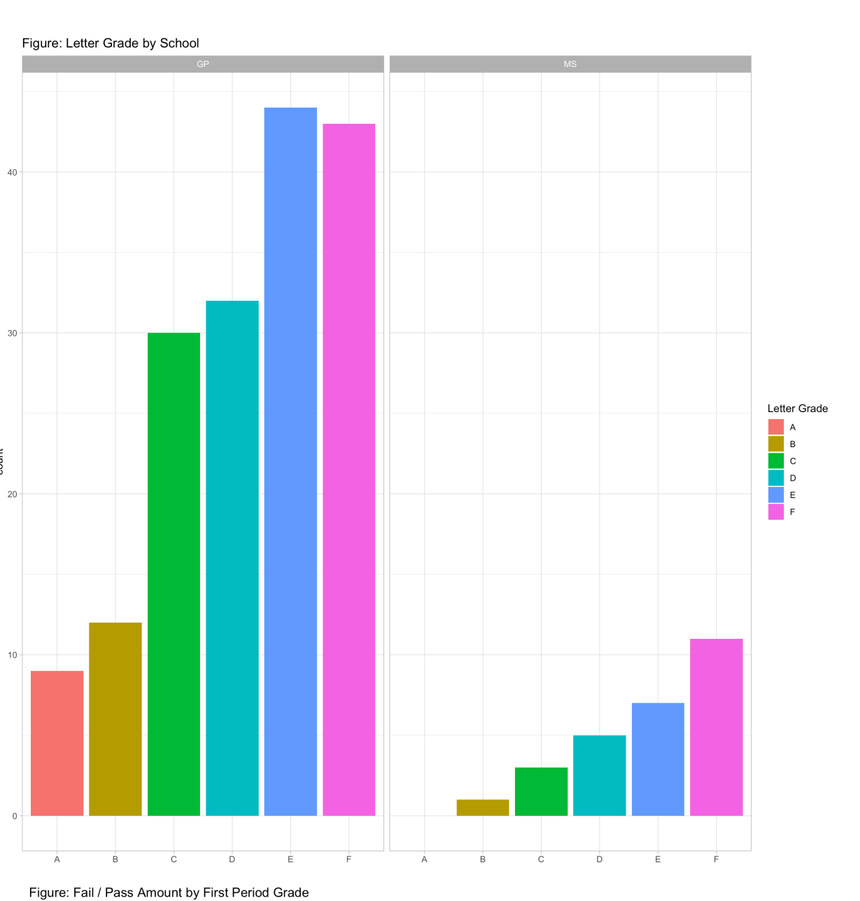
See the documentation for the `ucidata` package or the UCI website for additional documentation.

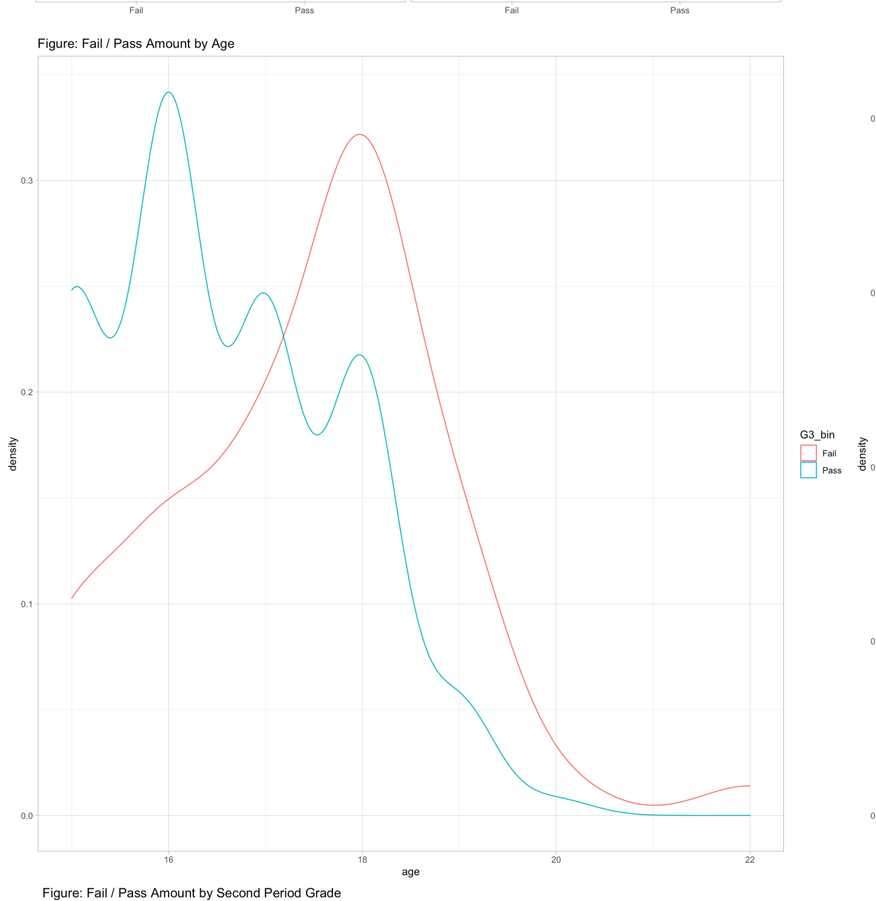
EDA

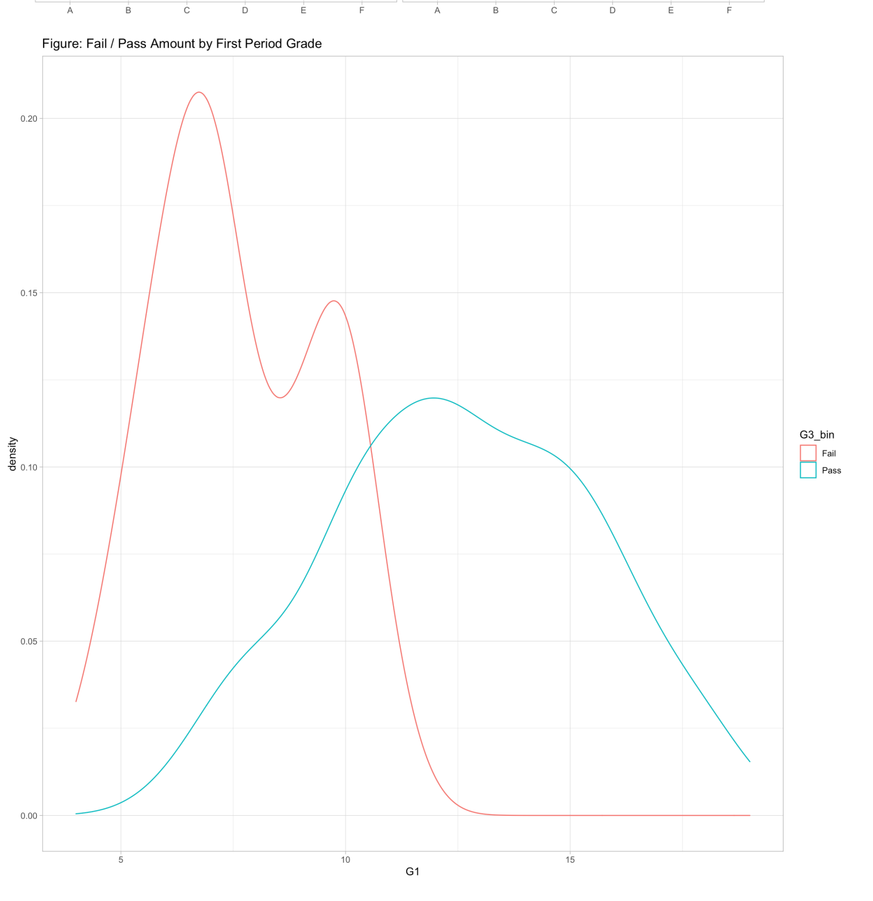
Table: Statistics by Outcome, Training Data

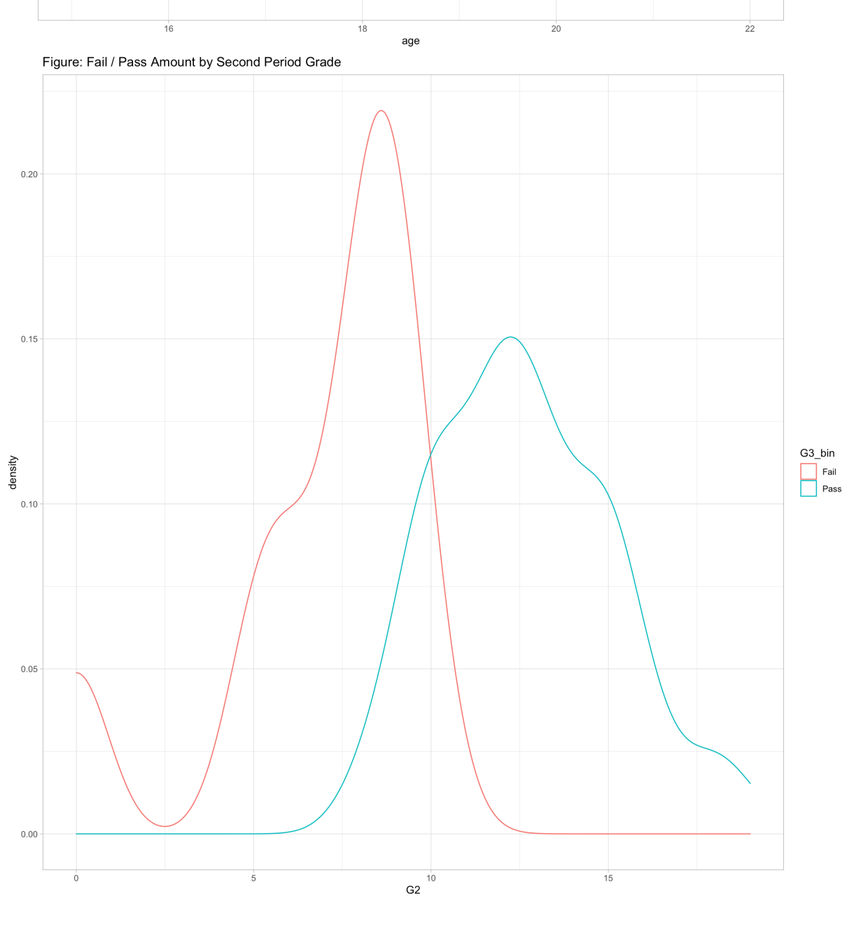
		Fail / Pass Amount				
G3_bin	Count	5th Percent	1st Quantile	Median	3rd Quantile	95th Percent
Fail	54	0	0	8	8	9
Pass	143	10	11	13	15	18











Additional Results

Table: KNN Multiclass Classification Result

k	Accuracy	Kappa	AccuracySD	KappaSD
5	0.555	0.432	0.085	0.110
7	0.595	0.483	0.089	0.114
9	0.573	0.456	0.100	0.127

Table: Random Forest Multiclass Classification Result

mtry	min.node.size	splitrule	Accuracy	Kappa	AccuracySD	KappaSD
2	1	gini	0.581	0.460	0.084	0.105
2	1	extratrees	0.532	0.393	0.125	0.164
13	1	gini	0.707	0.627	0.059	0.073
13	1	extratrees	0.670	0.580	0.059	0.074
25	1	gini	0.717	0.640	0.072	0.090
25	1	extratrees	0.701	0.621	0.047	0.058

Table: GBM Binary Classification Result

	shrinkage	interaction.depth	n.minobsinnode	n.trees	ROC	Sens	Spec	ROCSD	SensSD	SpecSD
1	0.1	1	10	50	0.977	0.850	0.916	0.026	0.174	0.065
4	0.1	2	10	50	0.975	0.830	0.929	0.027	0.198	0.067
7	0.1	3	10	50	0.968	0.833	0.916	0.035	0.137	0.065
2	0.1	1	10	100	0.969	0.833	0.916	0.038	0.184	0.065
5	0.1	2	10	100	0.970	0.857	0.930	0.032	0.165	0.058
8	0.1	3	10	100	0.962	0.820	0.915	0.038	0.184	0.074
3	0.1	1	10	150	0.966	0.800	0.922	0.037	0.197	0.063
6	0.1	2	10	150	0.967	0.820	0.922	0.035	0.184	0.063
9	0.1	3	10	150	0.968	0.837	0.908	0.033	0.157	0.083

Table: Logistic Regression With a Lasso Regression Penalty Summary

	Length	Class	Mode
lambda	100	-none-	numeric
cvm	100	-none-	numeric
cvsd	100	-none-	numeric
cvup	100	-none-	numeric
cvlo	100	-none-	numeric
nzero	100	-none-	numeric
call	6	-none-	call
name	1	-none-	character
glmnet.fit	13	lognet	list
lambda.min	1	-none-	numeric
lambda.1se	1	-none-	numeric
index	2	-none-	numeric

Table: Logistic Regression With a Ridge Regression Penalty Summary

	Length	Class	Mode
lambda	100	-none-	numeric
cvm	100	-none-	numeric
cvsd	100	-none-	numeric
cvup	100	-none-	numeric
cvlo	100	-none-	numeric
nzero	100	-none-	numeric
call	6	-none-	call
name	1	-none-	character
glmnet.fit	13	lognet	list
lambda.min	1	-none-	numeric
lambda.1se	1	-none-	numeric
index	2	-none-	numeric

1. Student Performance Data Set↩