Using Machine Learning to Predict Academic Performance:

Maheen Syed
December 2nd 2019
Data Science Initiative, Brown University
https://github.com/msyed96/DATA1030-Project

Introduction:

The goal of this project is to examine the different ways in which educational attainment may be impacted in classroom settings. The main question to explore is whether or not I can predict the academic performance of students in a classroom, based on the various features I know about the students. The specific dataset that I explore is the *Student Performance in a Mathematics Classroom in Portugal* dataset, which was obtained from the UCI Machine Learning Repository. My main motivation to explore this dataset stems from my personal interest in education - specifically secondary/high school education. As someone who has been actively involved in several teaching roles I am always keen to see how I can make my classroom more accessible to students. Ultimately it is important for educators to understand the different elements that may or may not impact the performance of their students. This knowledge, can help equip educators with the tools needed to meet the diverse needs of students in order to make education more inclusive and accessible for all students, and not just a select few.

The dataset that I explore highlights student performance in a secondary school mathematics classroom, in two schools that are based in Portugal. The main marker of academic performance is grades achieved in the classroom . The grades are appointed on a scale of 0 to 20 points. The students are evaluated three times a year - with their final performance gauged by a final examination. My model is a classification model where the final grade (G3) is the target variable and has the labels of pass or fail, depending on the points obtained by the students. The dataset has 33 well defined features including but not limited to age, students' family size, mother's job, father's job, weekly study time, number of past class failures, grades on the first exam, grades on the second exam, etc.

Exploratory Data Analysis:

While performing EDA my main goal was to study the properties of my target variable and look at the class specific distributions of both my numerical features as well as my categorical features.

My analysis started off with calculating the balance of my dataset. This balance allows to gauge what percentage of the data values fall in the specific classes. In the case of my dataset, about 33% of data fell in Class 0 (failures) and 67% of the data fell in Class 1 (passes), indicating that the data was balanced.

Next, I examined the relationship between the several categorical features in relation to the target variable G3. The visualizations that stood out include:

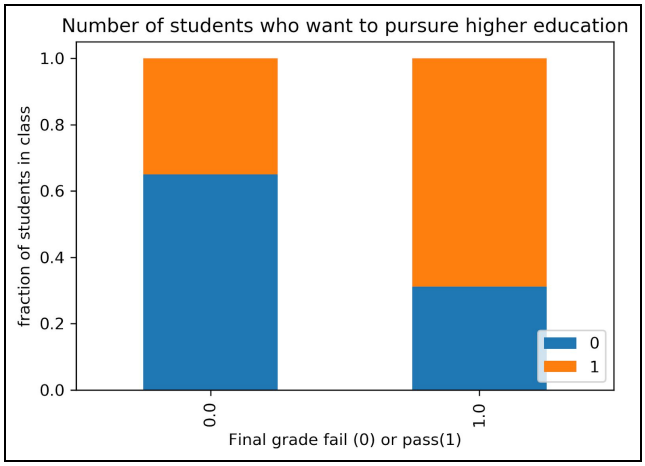


Fig. 1

Fig. 1: Those students that aspire to go to high school and college etc. tend to perform significantly better on the final examination than those who do not hold such aspirations.

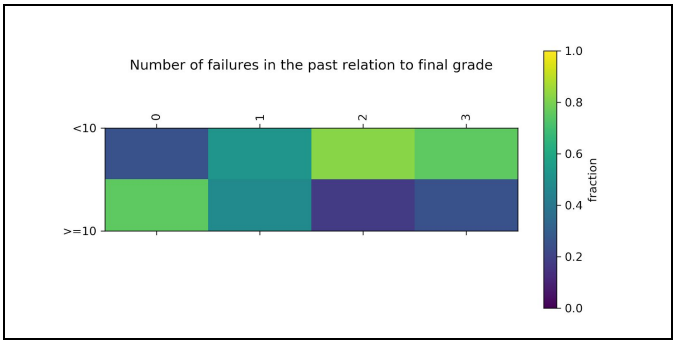
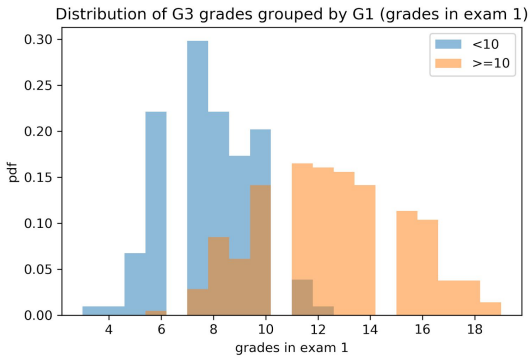
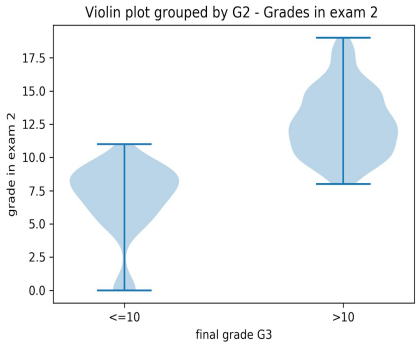


Fig. 2

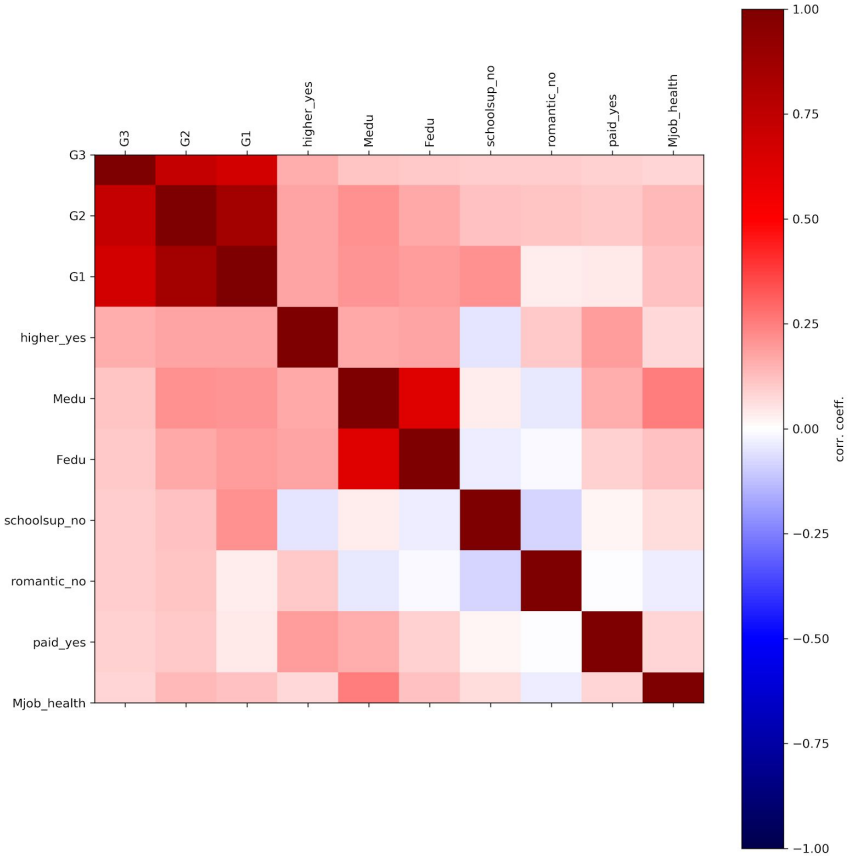
Fig. 2: Those students that have failed in past exams are more likely to fail in future exams. This is something I expected but it was interesting to see the data confirm it.

I then examined my numerical features with respect to the target variable. It was clear from this examination that G3 had a very strong correlation with the grades in the first tem, G1 and the grades in the second term, G2 as can be seen in the visualizations below:





This was also noticeable in the correlation matrix of my various features with respect to the target variable G3:



Based on these results, I concluded that while it is inherently difficult to predict G3 without G1 and G2 (as these have such high predictive powers), this prediction has much more value for teachers as G1 and G2 have a high impact on performance of the students. Therefore in my further analysis, I aimed to examine three specific models:

- Configuration I: Includes all variables - G1 and G2 included
- Configuration l II: Excludes G2
- Configuration III: Excludes G1 and G2 .

Methods:

To start my analysis, I grouped the data in my target variable G3 whereby values < 10 represented a failure and values >= 10 represented a passing grade on the final exam. I then label encoded my target variable. I applied one hot encoding to my categorical variables and standard scaling to my numerical features (the ordinal features had already been encoded in the original data set). I originally had 395 rows and 33 columns and after preprocessing I had an additional 50 columns. I did not have any missing values in my dataset. The preprocessing of variables was done using pipelines.

For each configuration (I, II & III), I developed an ML pipeline, using K Fold Cross Validation (CV) pipeline. The CV process allows one to train learners using one set of data and testing it using a different set. The pipeline method allows to perform both CV and parameter tuning effectively which is why I chose to use this method. I performed splitting of my data, preprocessing via pipelines as well as parameter tuning in my K fold CV pipeline function. Firstly, I split my data and created my train, test and other data (for X and y), and then split my other data based on K Fold sampling (as my data was balanced). I applied consistent method of sampling for each ML model I used.

The supervised ML models I used for my classification problem hyper parameter tuning were logistic regression, random forest (RF) classification and SVM classification. For logistic regression, I tuned the parameter C using eight values spaced evenly on a log space between 10^-2 and 10^2 , using a lasso regularization. For RF, I tuned two parameters - the max depth with values between 2 and 20 evenly spaced at intervals of 5, and the min sample split, with values between 1 and 30 evenly spaced at intervals of 5. For SVC, I tuned parameter C, with values [0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000] and gamma with values [0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]. The values chosen for these parameters were done so as to avoid edge cases.

I used the accuracy score as a metric to evaluate my classification model(s) because I was dealing with a balanced classification problem. Furthermore, to measure uncertainties due to splitting and due to non-deterministic ML methods I would redo my split for ten different random states and calculated the standard deviation of the accuracy score for each ML model (for each Configuration I, II & III). This is showcased in the Table A below:

	Model Standard Deviation (STD)		
Supervised ML Model:	**Configuration I (G1 + G2) (STD)**	**Configuration II (G2 Excluded) (STD)**	**Configuration III (G1 + G2 Excluded) (STD)**
Logistic Regression (LR)	+/- 0.0248	+/- 0.0457	+/-0.0425
Random Forest (RFC)	+/- 0.02547	+/- 0.0362	+/- 0.0520
Support Vector Classifier (SVC)	+/- 0.0322	+/- 0.0490	+/- 0.0594

Table A.

Results:

Table B below, show-cases the mean accuracy score for each supervised ML model (where the balance of the data is **0.6708** with respect to each Model I, II and III.

	Mean Accuracy Score		
Supervised ML Model	**Configuration I (G1 + G2)**	**Configuration II (G2 Excluded)**	**Configuration III (G1 + G2 Excluded)**
LR	**0.8936**	0.8215	**0.6861**
RFC	0.8797	**0.8139**	0.6721
SVC	0.8999	0.8063	0.6759

Table B.

Table C below showcases how many standard deviations the specific ML model was, above or below the baseline accuracy score. This was calculated by using the formula:

Standard deviations above baseline = Model Accuracy - Baseline Accuracy / Standard Deviation

This calculation was then used to gauge the best ML model for each case I, II and III, as seen in Table C. The model with highest numbers of standard deviations above baseline performed best.

	Standards Deviations Above Baseline Accuracy:		
	Configuration I	**Configuration II**	**Configuration III**
LR	**8.98**	3.29	**0.36**
RFC	8.20	**3.95**	0.025
SVC	7.11	2.76	0.0859
Best ML Model Based on STD above baseline accuracy	LR	RFC	LR

Table C.

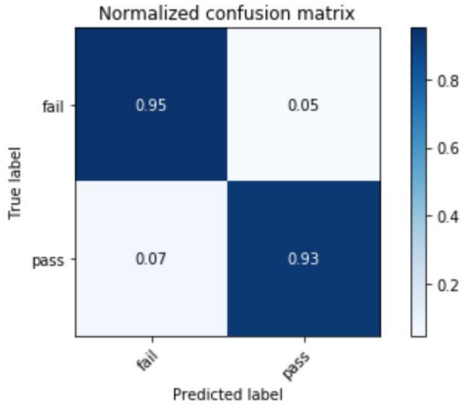
Therefore, the best ML model for Configuration I is Logistic Regression. For configuration II it is Random Forest. For configuration III, Logistic Regression is the best model.

Finally, Table D below showcases the specific best parameters for the best models for each configuration. The specific parameters chosen gave highest accuracy value for the model. The parameters also lie near the middle of the range of parameters specified for tuning.

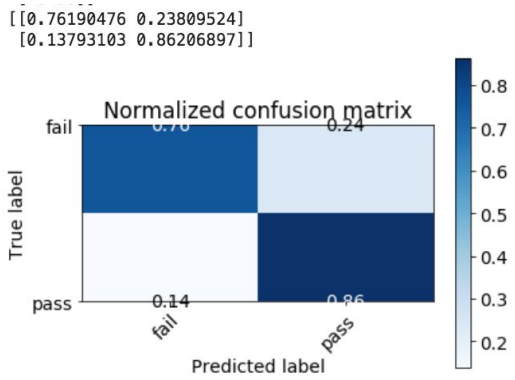
Configuration #	**Best Supervised ML Model**	**Best Parameter Value(s)**
I	Logistic Regression	C = 0.5179
II	RFC	Max Depth=11 , Min Sample Split = 7
III	Logistic Regression	C = 0.13894

Table D.

To analyze these results further, I plotted the confusion matrices for the best models for each configuration (I, II,III) . The matrix below, for example, represents the logistic regression model for Configuration I. It can be seen that this model does a very good job of classifying points correctly.

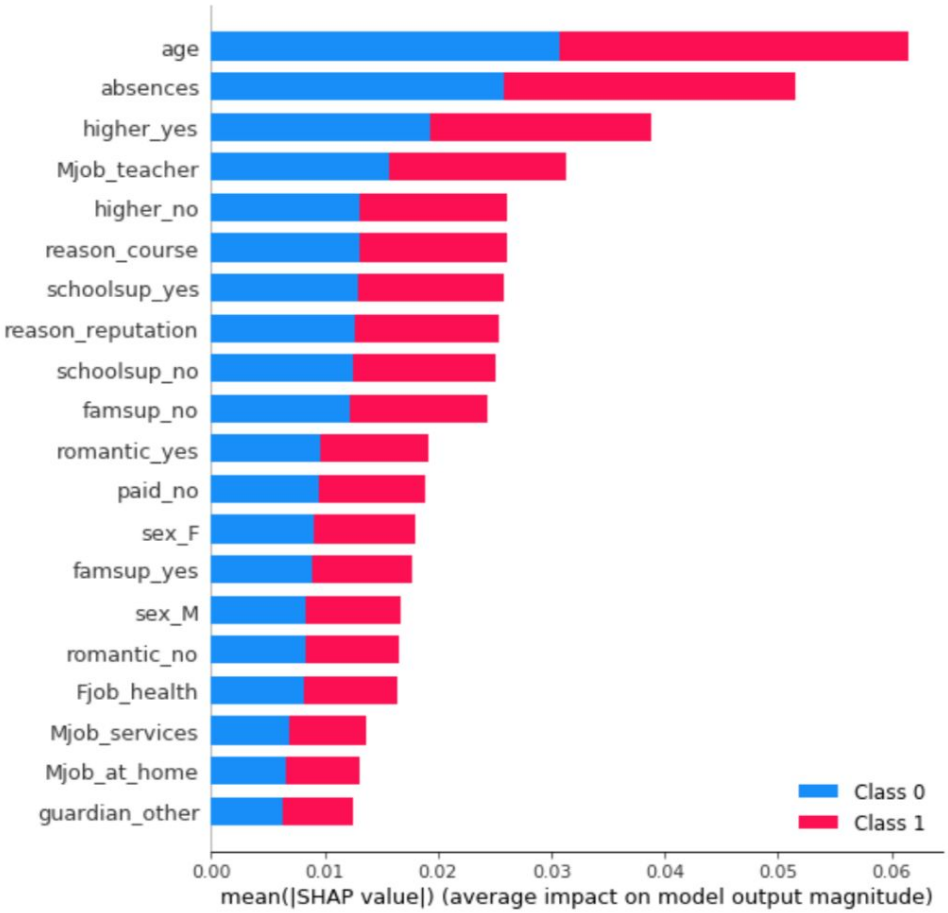


The confusion matrix for Configuration II showcases that the model does a decent job of classifying points correctly:

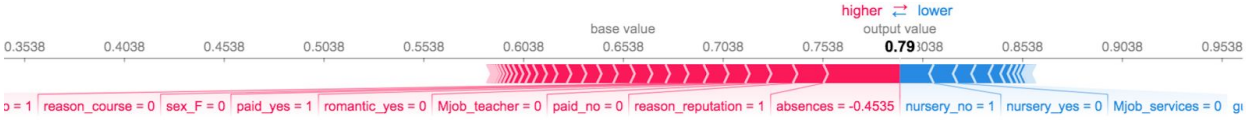


Based on the various results above we can see that Configuration I will be very accurate because G1 and G2 are very good indicators of how the student will do on the final exam. However, in terms of actionable insights, it is pretty limited, as by the time G2 is known, there might not be much that the teachers can do to help students who will get a bad G3 grade. Configuration II performs pretty well and the machine learning results *fit into an academic context* as the model allows educators to spot students that will be in academic trouble by the end of the term relatively early on The third configuration is interesting because the educators can keep an eye on kids who will be in academic trouble from the beginning of the term which would be a tremendous resource. The accuracy of this model, is significantly less than for Configuration I and Configuration II (due to the lack of highly predictive G1 and G2), however, it still above baseline accuracy and gives us insights into the students' performances. For example, using feature importance results generated by SHAP's Tree Estimator, educators would be interested in reaching out to those students with higher number of absences. Methods to increase attendance in class can include implementing roll calls/ using apps like Top Hat! It is also worthwhile for educators to research the reason behind the impact of additional school support/parents jobs/ etc. on student performance.

Global Feature Importance (using SHAP Tree Estimator):



Local Feature Importance (using SHAP Tree Estimator):



Outlook:

For future analysis, many steps can be taken to improve the model. As a very limited number of overall features in this dataset showcase importance, feature selection methods for the data should be further explored. Furthermore, information from a larger number of school districts, different class years, etc. should be gathered in order to give more insight into the model. It would also be interesting to observe data collected in different countries and gauge the impact of various cultural and social norms on grades Getting more insight about factors like commute time to school can also be interesting to analyze as such factors may take away from study time and impact final grades.

The weak spot of my modeling approach is that I used all the features in my data instead of performing certain feature selection methods. Thus to improve the model feature engineering which involves careful selection of features in order to avoid dealing with any extra noise from the rest of the data, can improve the results. This, along with additional techniques like principal component analysis and Xgboost can enhance the model predictions and interpretability. I hope to address these in the future as I continue to work on this project.

References:

P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7

Datasets involving student performance analysis:

https://www.kaggle.com/pamhohhgkgm/student-data-analysis

https://www.kaggle.com/samuelmjoseph/student-grade-prediction-using-decisiontree