Predicting Student Math Performance Using Machine Learning

John Mathew, Joe Huang, and Zakir Makhani

December 2021

Abstract

Grades are often seen as one of the most important metrics of a child's performance in their childhood and potentially a determinant for their future success. As a result, parents and educators often wonder what type of environment for a child provides the optimal outcome. Machine learning algorithms seem to hold a large amount of potential for this field due to the abundance of available data on the grades of children, along with details about their specific lives. Through using these values, the goal of this project is to analyze the predictability of a child's grade with the knowledge of their upbringing, as well as examine what are the most important factors for a child's future grade success. We utilized three different hypertuned classification models which include SVMs, Logistic Regression, and Neural Networks. Among these three, they all yielded significantly greater prediction accuracy than a random guess, however Neural Networks had performance issues in comparison to the other two options due to a lack of big data.

1 Introduction

Our project seeks to investigate the predictability of student outcome in mathematics, measured by whether or not the student passes or fails the class. We also seek to discover which variables are most important to student success. To do this, we use three different classification methods, logistic regression, support vector machine, and neural network, to predict student outcomes in mathematics on a dataset containing the statistics and outcomes for secondary school students in Portugal. The dataset had 395 samples, each representing a student and different statistics about them. There were a total of 33 features, each being used to predict the student's final grade in mathematics, shown in the following table:

Table 1: The preprocessed student related variables

Attribute	Description (Domain)
sex	student's sex (binary: female or male)
age	student's age (numeric: from 15 to 22)
school	student's school (binary: *Gabriel Pereira* or *Mousinho da Silveira*)
address	student's home address type (binary: urban or rural)
Pstatus	parent's cohabitation status (binary: living together or apart)
Medu	mother's education (numeric: from 0 to 4[a])
Mjob	mother's job (nominal[b])
Fedu	father's education (numeric: from 0 to 4[a])
Fjob	father's job (nominal[b])
guardian	student's guardian (nominal: mother, father or other)
famsize	family size (binary: ≤ 3 or > 3)
famrel	quality of family relationships (numeric: from 1 – very bad to 5 – excellent)
reason	reason to choose this school (nominal: close to home, school reputation, course preference 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 $1 \leq n < 3$, else 4)
schoolsup	extra educational school support (binary: yes or no)
famsup	family educational support (binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
paidclass	extra paid classes (binary: yes or no)
internet	Internet access at home (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
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)
Walc	weekend alcohol consumption (numeric: from 1 – very low to 5 – very high)
Dalc	workday 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)

a 0 – none, 1 – primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education.
b teacher, health care related, civil services (e.g. administrative or police), at home or other.

The final grade is given on a scale from 0 to 20, where any grade less than 10 is considered failing and any grade greater is considered passing. Our goal was to compare the predictive ability of the three different classification methods as well as to compare which variables they found most important for the prediction.

2 Technical Approach

Before training our models, we had to preprocess the data in a way that was conducive for inputting it into models. This meant that once we read in our csv data, we first had to convert it to a DataFrame. Once it was converted to a Dataframe, it needed to be transformed so that each column represented a feature, along with a numerical value associated with that feature. For features that were already numerical values, this was quite simple. However, for non-continuous values, we had to map each of the unique values within a column to continuous values such as {"'at_home"' : 0, "'health"' : 1, "'teacher"' : 2, "'other"' : 3, "'services"' : 4}. Beyond this, we then had to determine what values to be kept because they held potential scientific significance. This meant removing features which felt slightly less significant such as "address," "famsize," or "romance" in favor of features that seem like they'd have more direct correlation such as past grades, sex, or whether or not their homes have internet. Once all this preprocessing was done, we then converted the DataFrame into y which

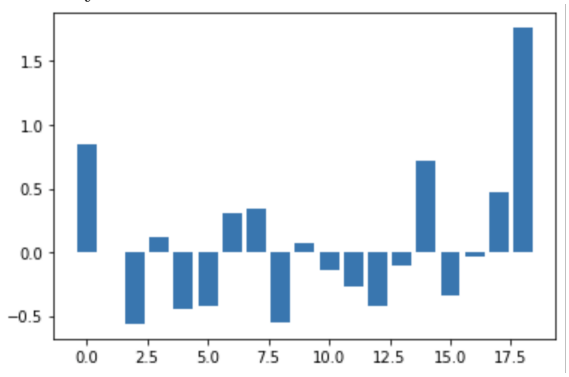
was the column of final grades, and X which was every other column in the DataFrame. We then utilized a built-in function called train_test_split to convert X and y into X_train, y_train, X_test, and y_test for training purposes. The training data contains 60% of the data while the testing contains 40% due to it being a common convention. Once all these steps were completed, we were able to feed this data into our models for training and testing.

The approach we utilized for creating and training an SVM Classifier was first importing the necessary libraries. This includes importing SVC and GridSearchCV from sklearn. SVC was the premade SVM created in sklearn that we initialized to use as the SVM classifier. To figure out the ideal parameters for this SVM, GridSearchCV was used to test out various possible parameters to see which one outputs the highest accuracy assessment on the test set. A notable piece of info to take into account is that the penalty function used in any of the models is always squared l2 penalty. The hyperparameters tested for the SVM include what type of kernel (i.e. RBF, Polynomial, Sigmoid, or Linear), degree which is only applicable for the polynomial, and a regularization parameter that is inversely proportional to the strength of the regularization. For the logistic regression, we imported the LogisticRegression class from sklearn, which contains a method fit. Such a fit method allows us to "Fit the model according to the given training data" [Pedregosa]. The only variable we decided to pass the LogisticRegression was max_iter, which determines the maximum number of iterations we would like to perform if the regression does not converge. Thus, by default our linear regression uses an l2 norm penalty function and uses a limited memory BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm as a solver to optimize the regression. Furthermore, we use the default inverse regularization parameter, C, as 1.0. For hypertuning, we tune the C, the penalty function (as only certain penalty functions are compatible to certain solvers, we only use the l2 penalty function or no penalty), the maximum number of iterations, and the solver (newton conjugate gradient [newton-cg], limited memory BFGS [LBFGS], Stochastic Average Gradient [SAG], and SAGA [a variant of of SAG]). We perform our tuning using the GridSearchCV function, along with a pipeline and random forest classifier.

We also used a neural network to predict the student outcomes. We used the PyTorch library to create a neural network, which takes in the learning rate, number of nodes, number of epochs, and momentum. The neural network had 4 hidden layers, each with half the number of nodes as the one before it. The first layer had 19 nodes, one for each feature, and the last layer had a single node for the single variable being predicted, whether or not the student passed. The network was trained using stochastic gradient descent with binary cross entropy as its loss function.

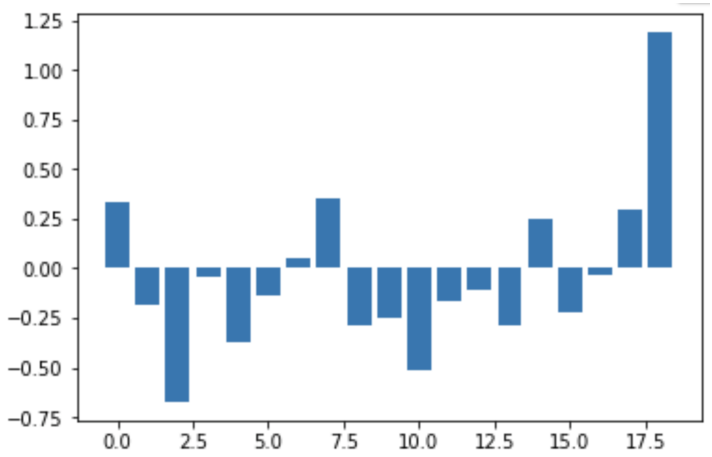
3 Expiremental Results

The first notable result from the SVM was the resulting ideal hyperparameters generated by GridSearchCV. The resulting hyperparameters were the kernel being polynomial, the degree being 1, and the regularization parameter being 10. This essentially means that the ideal kernel was linear due to the polynomial degree being 1, and that the regularization was slightly strong due to C being 10, a smaller value, meaning a larger regularization according to how the regularization is inversely proportional. The other notable result we looked at was the accuracy of the SVM in conducting predictions on the testing data. When we used the .score() function from GridSearchCV on our x_test and y_test, the accuracy was 90.91%. The following graph shows the importance of each feature determined by the SVM:



From left to right, the bars are for school, sex, age, Medu, Fedu, Mjob, Fjob, traveltime, studytime, failures, schoolsup, famsup, higher, internet, famrel, goout, absences, G1, and G2.

Initially, from our almost default logistic regression, we reached an accuracy of about 88.88%. From the tuning of hyperparameters, we produce a logistic regression with no penalty function using the SAG solver running for a maximum of 5000 iterations. In addition, as there is no penalty function, the parameter C is ignored within the regression. We receive an accuracy of almost 92%.



From left to right, the bars are for school, sex, age, Medu, Fedu, Mjob, Fjob, traveltime, studytime, failures, schoolsup, famsup, higher, internet, famrel, goout, absences, G1, and G2.

Without tuning the hyperparameters, the accuracy of the neural network was about 69.70%. The decreased accuracy in comparison to the other models is likely due to the fact that training neural networks requires many data points and ours only had 395. The network was tuned using GridSearchCV in order to optimize the number of epochs and learning rate. It found that the optimal learning rate is .2 and the optimal number of epochs is 50, which is the max amount that was tested.

4 Conclusion

We mostly achieved what we set out to do. The subgoal we had was to see what were the most relevant factors for determining a child's future grade, and indirectly, their future success. To determine the importance of each factor in determining a child's future grade we first trained a Logistic Regression Classifier on the features, paired with their label. We then retrieved the weights associated with each future. We assumed the importance of a feature was associated with how largely positive or negative the weight it received was (absolute value greater than 1). Notable takeaways from this were how a child's past performance, especially in their grade for semester 2, intuitively had a strong positive correlation with the final grade. More surprising information was contained in details such as how the number of absences and parental occupations had barely any effect on their final grade while features such as travel time, study time, and number of classes failed previously had strong effects. The most significant decline is with G1, as it possesses the weight with the least effect on the logistic regression, although for some of the setups Cortez and Silva had G1 as the feature with the maximum importance.

Factors with strong positive weights including school, age, sex, family

relations, and the aforementioned G2 were simple to comprehend: the school one goes to significantly affects what they learn and the standards they perform up to; as these are secondary schools, students who are older than 18 have almost guaranteed been held back and therefore less likely to perform well; more stable home life leads to better mental health and the ability to focus more on school.

However, not all the highly weighted features were not as easy to grasp with the offenders being the desire to attend higher level education and the amount of time studying. Both have high negative weights although by nature one would believe if you were to study for school and have the motivation to continue school (college), a student would put more effort and thus perform better in school. Notably, the weights regarding parents were significantly lower compared to other features, especially education and occupations. Therefore, the logistic regression implies these have less impact than individual traits/activities.

In respect to our overall final goal which was to see whether or not we were able to accurately predict a child's future grades based on their background information, we achieved this goal to varying success. With our SVM, and Logistic Regression Classifier we were able to achieve rather high testing accuracy of 90.91% and 92% respectively with optimal hypertuning. However, with our Neural Network we weren't able to achieve nearly as high of an accuracy, likely due to the fact that we had a very small set of data, only around 150 pieces of data for each class. This was likely not sufficient due to neural networks generally requiring a very large set of training data for optimal performance.

5 Participant Contribution

All three of us worked together on Zoom to process the data and build the models, but we split up work between us after that. Joe Huang tuned the hyperparameters for the SVM and wrote some of the abstract, results, and conclusions sections. Zakir Makhani tuned the hyperparameters of the logisitic regression model and also wrote some of the abstract, results, and conclusions sections. John Mathew tuned the hyperparameters of the neural network, wrote the introduction and contributed to the approach and conclusions sections.