At-Risk Students Prediction Using Machine Learning

Reginald Ledain Gentillon
Master in Computer Science
Advisor: Dr. Jeffrey Duffany
Electrical and Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract — This article intends to discover how machine learning can be used to predict at-risk students during the school year. Different algorithms were tested within a common framework to compare their accuracy and their interpretability. Using some education expert knowledge, we examined each model relevance in relation to the most important features they used. Attendance, language proficiency and interim test completion were found to be very deterministic in the models prediction capabilities; not a surprise but a validation of the adequacy of the technology for this difficult task.

Key Terms *— Decision Trees, Deep Learning, Education, Machine Learning.*

INTRODUCTION

The school districts in the K-12 education domain usually rely on descriptive after-the-fact analytics to take actions on students' performance. Those actions come usually too late for many students and the hope is to implement corrective changes for the next cohort. Beyond obtaining updated reports during the school year, channeling the most urgent information to school leaders and teachers to intervene and help the students at-risk of failing in the end-of-year standardized tests, would be ideal.

While formative and interim assessments are a good way of measuring the students learning process [1], they usually don't offer that 360 view that can predict the actual student performance on their standardized tests. For that reason, we consider that pairing a mid-term, fall or winter, scores for the different strands with all the other indicators will create a more deterministic dataset for our intended goal.

Existing work in this field mostly try to predict broader impact in terms of the district graduation rates for instance. We will look into using unit testing in math in combination with a variety of well-known indicators like attendance, behavior and demographics, to create a possibly early warning for those students susceptible to fail at the end of the year for a particular subject.

FRAMEWORK AND METHODOLOGY

Various machine learning models can be used for classifying students at-risk. We will explore the efficacy and convenience of three popular models:

- Multi-layer perceptron or neural network
- Classification and Regression Tree
- Random Forest

Python has the popular scikit-learn library [2] which implements a large variety of those models and is the main framework for this project. To help identify the optimal configurations of those models we will use some specialized functions to find those parameters that yield the best accuracy starting with a generic model as our baseline. While we compare the three models' performance, we will also identify the most important features as rated by each one. We will also investigate each model interpretability and capacity to provide insight into the subject matter to help identify the root causes.

The data for this work will be comprised of 7th-graders of all the schools in a mid-size district in the United States using their fall interim math assessment along with their corresponding end-of-year performance on their state assessment. The data will also include attendance, behavior and socio-

demographics variables to provide a desired broader range of influenceable features.

Data Preparation

The multiple data sources were combined into one main dataset. Enabling the data for machine interpretation required additional transformations. Most importantly our target variable was converted to suit a binary classification model. The target variable is the student achievement level which was summarized to a binary class as follows in table 1.

Table 1
Class Definition

Target Variable	Binary Class
Standard Exceeded	Proficient (1)
Standard Met	
Standard Nearly Met	Not Proficient (0)
Standard Not Met	

Additional transforms were applied, turning variables to one hot encoding like gender and other categorical values.

Cluster Analysis

An initial exploration of the dataset was performed to identify possible clusters and correlations with the target variable. The Weka tool was used for this task. On the y-axis, our binary class with the two possible values, are plotted against each variable to reveal the clusters obtained by cluster analysis over the whole training set.

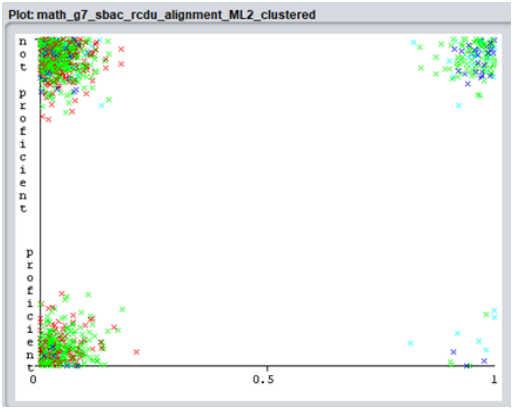

Figure 1
Hasdisability Clusters

The "hasdisability" variable clearly reveals a cluster of students (figure 1) associated with the variable value of 1 (True) for students that were "not proficient". Whereas very few instances were associated with this value as "proficient."

A similar pattern can be observed for the "lepstatus" variable, representing students with Limited English Proficiency (figure 2).

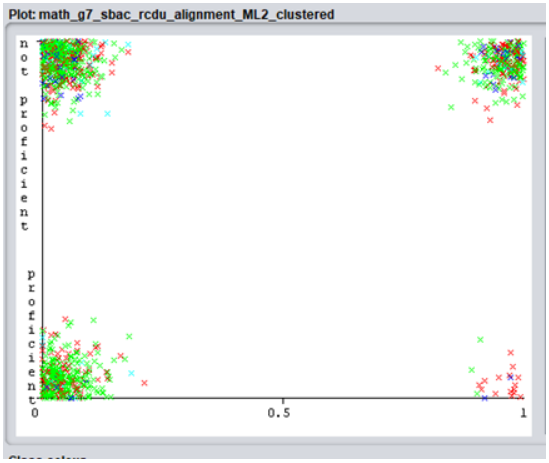

Figure 2
Lepstatus Clusters

However, unit tests results don't seem to be as well defined. The -1 value representing students without a score for a particular strand who could also be considered to not have taken the test tend to cluster in larger numbers as not proficient (figure 3).

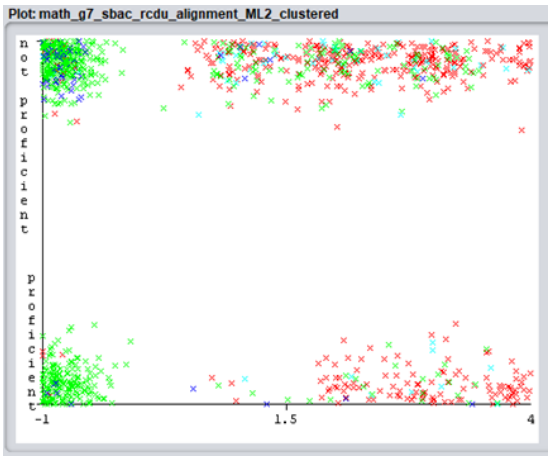

Figure 3
Unit 1 Statistics Clusters

Clusters unsupervised learning gave us a good glimpse into some interesting pattern in the data. We

will discover whether those patterns hold in the supervised models.

MACHINE LEARNING PREDICTION

While there are numerous Machine Learning algorithms, some behave better than others for each particular use. For our binary classification we use a dummy classifier as our baseline and compare with the other algorithms, Multi-Layer Neural Network, Decision Tree and Random Forest. We will use accuracy as the evaluation metric and derive features' importance ranking for each model.

After the basic data preparation was performed on the dataset, a training and a test subset was established to be used by all models. The test set had 233 records (20%) and the training set 929 records (80%) with a seed = 1 to perform a random split into the two.

Baseline Classifier

To judge the different models performance, establishing a baseline will help measure their effectivity in predicting students' performance. Scikit-learn python library has the DummyClassifier class that uses simple rules to provide this baseline.

The best accuracy was obtained by using the strategy value of "most_frequent" which always predicts the most frequent label in the training set:

Accuracy score = 0.712

Neural Network Classifier

The scikit-learn package for a neural network implements the log-loss function using Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) [3] or stochastic gradient descent (SGD) [4] as optimizers. The following hyper parameters where evaluated using a thorough search of the specified parameter values to determine the most optimal accuracy of the model with our dataset.

- Hidden_layer_sizes:
  - From 1 to 3 layers of perceptrons of different sizes [(50,50,50), (50,100,50), (14,), (100,)]
- Activation:
  - the hyperbolic tan function (tanh)
  - the rectified linear unit function (relu)
- Solver:
  - stochastic gradient descent (sgd)
  - limited-memory BFGS (lbfgs)
  - optimized SGD (adam)
- Alpha:
  - regularization parameter to prevent over-fitting (0.0001, 0.05)
- Learning_rate:
  - schedule for weight updates ('constant','adaptive')

The best parameters combination found by the grid search were:

Activation = tanh
Alpha = 0.05
hidden_layer_sizes = 100 (1)
learning_rate = constant
solver = adam

Using those parameters, the neural network classifier was trained on the data with a prediction accuracy that improved the baseline:

Accuracy score = 0.78

To help with the interpretability of the model, the features' importance based on the weights assigned by the algorithm can be derived as shown in table 2.

Table 2
NN Features Importance Ranking

#	feature	weight
1	gr07_u1_statistics_fall_level	0.0015
2	days_attended	0.0013
3	days_absent_excused_non_suspension	0.0011
4	gr07_u2c_mult_dividing_fall_level	0.0011
5	gr07_u2a_the_number_line_fall_level	0.0011
6	gr07_u6_probability_fall_level	0.0009
7	gr07_u5_unit_rates_etc_fall_level	0.0006
8	gender_female	0.0006
9	gr07_u2b_comb_quantities_fall_level	0.0006
10	total_action_duration_days	0.0004
11	days_absent_unexcused_non_suspension	0.0004
12	lep_status	0.0000
13	gender_undefined	0.0000
14	hasdisability	0.0000
15	migrantstatus	0.0000
16	gr07_u7_geometry_fall_level	0.0000

17	days_absent_out_of_school_suspension	0.0000
18	days_in_attendance_in_school_suspension	0.0000
19	gr07_u3_equat_ineq_fall_level	0.0000
20	gender_male	0.0000
21	economicdisadvantagestatus	-0.0004
22	disciplinary_incidents	-0.0011

Interestingly, if we compare the weights observed in this table to the initial cluster analysis, we can see that the "unit 1 statistics" is weighing heavily in the model decision making, whereas the "lep_status" and the "hasdisability" don't seem to have much influence on it.

Decision Tree Classifier

A decision tree classifier is a rule-based algorithm that can predict whether a student is proficient or not based on the rules inferred from the data features. Scikit-learn employs an optimized version of the Classification and Regression Tree (CART) algorithm with binary trees where features and thresholds used at the nodes depend on the largest gain obtained.

The classifier hyper parameters were determined based on the following ranges:

- max_depth:
  - maximum depth of tree (3, 4, 5, 6)
- min_samples_leaf:
  - minimum samples fraction required to be at a split node (0.04, 0.06, 0.08)
- min_samples_split:
  - minimum samples required to split node (2, 3, 10)
- max_features:
  - features fraction to consider to obtain best split = (0.2, 0.4,0.6, 0.8)

The best parameters found by grid search were:
max_depth = 6
max_features fraction = 0.8
min_samples_leaf fraction = 0.04
min_samples_split = 2

Using those parameters, the decision tree was also able to improve the baseline prediction accuracy:

Accuracy score = 0.78

As performed with the neural network analysis, we can extract the features importance in order to better validate our assumptions and compare the algorithms. Table 3 displays those features ranked from highest to lowest.

Table 3
Decision Tree Features Importance Ranking

#	feature	weight
1	lep_status	0.32
2	economicdisadvantagestatus	0.17
3	days_attended	0.17
4	gr07_u1_statistics_fall_level	0.14
5	gr07_u3_equat_ineq_fall_level	0.09
6	days_absent_unexcused_non_suspension	0.07
7	gr07_u2c_mult_dividing_fall_level	0.03
8	gr07_u2b_comb_quantities_fall_level	0.02
9	gender_male	0
10	gender_female	0
11	gender_undefined	0
12	hasdisability	0
13	migrantstatus	0
14	days_absent_out_of_school_suspension	0
15	days_in_attendance_in_school_suspension	0
16	days_absent_excused_non_suspension	0
17	disciplinary_incidents	0
18	total_action_duration_days	0
19	gr07_u2a_the_number_line_fall_level	0
20	gr07_u5_unit_rates_etc_fall_level	0
21	gr07_u6_probability_fall_level	0
22	gr07_u7_geometry_fall_level	0

We can see in this occasion that the top ranked feature is the "lep_status" in accordance with a similar striking observation in the clustering of this feature versus the target variable.

Figure 4 displays one of the major benefits of a decision tree which is the ability to visualize the decisions made at each node to classify a set of features in an instance of the dataset [5].

Random Forest Classifier

A Random Forest classifier uses Ensemble Learning to fit various decision tree classifiers applied to subsets of the dataset and averages them to improve the prediction and avoid over-fitting [4].

The classifier hyper parameters were determined based on the following ranges:

- max_depth:

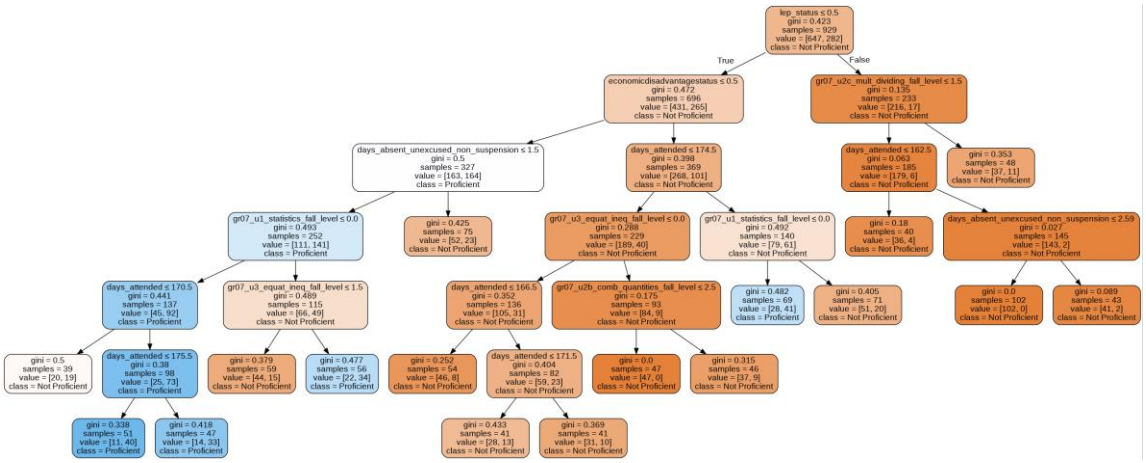

Figure 4

Decision Tree Classifier Visualization

- maximum depth of tree (3, None)
- features to consider to obtain best split (1, 3, 10)

- max_features:
- min_samples_split:
  - minimum samples required to split node (2, 3, 10)
- bootstrap:
  - whether samples are drawn with replacement (True, False)
- criterion:
  - function to measure the quality of the split ("gini", "entropy")

The best parameters found by grid search for the Random Forest Classifier were:

max_depth = None

max_features = 1

min_samples_split = 10

bootstrap = True

criterion = entropy

Using those parameters, the random forest classifier surpassed all the others in accuracy:

Accuracy score = 0.83

The features importance given by the random forest classifier with this increase accuracy are listed in table 4.

Table 4

Random Forest Classifier Features Importance Ranking

#	feature	weight
1	days_attended	0.15
2	days_absent_excused_non_suspension	0.12
3	lep_status	0.11
4	gr07_u1_statistics_fall_level	0.09
5	days_absent_unexcused_non_suspension	0.08
6	economicdisadvantagestatus	0.07
7	gr07_u3_equat_ineq_fall_level	0.05
8	hasdisability	0.04
9	disciplinary_incidents	0.04
10	gr07_u2a_the_number_line_fall_level	0.04
11	gr07_u2b_comb_quantities_fall_level	0.04
12	gr07_u2c_mult_dividing_fall_level	0.04
13	gr07_u5_unit_rates_etc_fall_level	0.03
14	days_absent_out_of_school_suspension	0.02
15	total_action_duration_days	0.02
16	gr07_u6_probability_fall_level	0.02
17	gr07_u7_geometry_fall_level	0.02
18	gender_male	0.01
19	gender_female	0.01
20	gender_undefined	0
21	migrantstatus	0
22	days_in_attendance_in_school_suspension	0

With the increased accuracy we can attest that some of the evidence-based [5] indicators are ranking higher as expected. For instance, the number of days attended weighs the highest and could be intuitively assumed, as absent students are more likely to fail their end-of-year test.

OVERALL RESULTS

The "days_attended" as well as the "lep_status" features lie consistently among the top 3 features in terms of importance when we look at how the three models weigh them. This observation is aligned with the expected importance in practice where attendance is of utmost importance and limited English proficiency of English language learners adversely impact their performance.

The Decision Tree matched the neural network performance but offered the benefit to visualize the resulting tree, making the model more easily interpretable.

The Random Forest model on the other hand had the highest accuracy. Nevertheless, Random Forest is considered a black box algorithm as it's not possible to visualize it in a single tree representation because it's an ensemble learning that averages its subtrees to make the predictions. Table 4 summarizes those results.

Table 4
Models Performance

#	Model	Accuracy
1	Baseline	0.71
2	Neural Network	0.78
3	Decision Tree	0.78
4	Random Forest	0.83

Yet, we were able to extract the features importance for all the models which helps with their interpretability and to assess our assumptions on the subject matter.

FUTURE WORK

Additional data preparation to scale some features like the tests score may greatly improve the model's accuracy. More features, as well as a larger dataset spanning multiple years can also contribute to better train the models.

Future work is planned to leverage the model's interpretability into individualized students' prediction as a warning system, while providing the teachers and administrators with the specific details about the weights or the rules that are used in the classification. End users enabled with this information will be able to take more specific actions to help the students succeed instead of relying on a black box prediction.

CONCLUSION

We have seen how the initial exploration of the data through clustering identified marked patterns for some features. Those features importance was also confirmed in their use by the 3 models we tested, where the Random Forest model was the most accurate.

Despite the use of different types of algorithms with varying techniques, we were able to validate some general known assumptions about the data and obtain each model accuracy and its adequacy to generate warnings for students at-risk of not passing their end of year test.

REFERENCES

[1] M. Perie, S. Marion and B. Gong, "Moving toward a comprehensive assessment system: A framework for considering interim assessments," *Educational Measurement: Issues and Practice*, vol. 28, no. 3, pp. 5-13, Feb. 2009. [Online]. Available: https://www.ncaase.com/docs/PerieMarionGong2009.pdf.

[2] F. Pedregosa et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, 12, pp. 2825–2830, Oct. 2011. [Online]. Available: https://arxiv.org/pdf/1201.0490.pdf.

[3] A. Haghighi. (2014, Dec. 2). *Numerical Optimization: Understanding L-BFGS* [Online]. Available: http://aria42.com/blog/2014/12/understanding-lbfgs.

[4] R. Roy. (n. d.). *ML: Stochastic Gradient Descent (SGD)* [Online]. Available: https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/.

[5] S. Koon, Y. Petscher and B. Foorman, "Using evidence-based decision trees instead of formulas to identify at-risk readers," *Institute of Education Sciences (IES)*, REL 2014-036, Washington, DC, USA, June 2014. [Online]. Available: https://ies.ed.gov/ncee/edlabs/regions/southeast/pdf/REL_2014036.pdf