Machine Learning: Predicting Graduation Rates of Virginia High Schools

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Lillian Cochrane

Fall, 2022

Technical Project Team Members

Matthew Gerace

Matt Koehler

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Daniel G. Graham, Department of Computer Science

Machine Learning: Predicting Graduation Rates of Virginia High Schools

CS 4991 Capstone Report, 2022

Lillian Cochrane
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
lmc2msm@virginia.edu

Abstract
Understanding that education inequality is a significant problem affecting high school dropout in Virginia, three computer science students at the University of Virginia wanted to know how schools can improve their on-time graduation rates. The team and I decided to use machine learning techniques to build a tool that predicts graduation rates of Virginia high schools based on data features such as free and reduced lunch eligibility rate and number of students enrolled in advanced programs. We trained and tested multiple regression models and compared them to see which model had the best performance based on their root mean square error (RMSE).

We compared linear regressor, decision trees, random forest, neural networks, and a voting regressor models and used Google Collab to collaborate on the Python program. For packages, we used pandas for data import and management, while we used models from the scikit-learn and tensorflow libraries. We achieved the lowest RMSE by combining the linear regressor, decision tree, and random forest models into a weighted voting regressor. Using this model, schools would be able to see how different factors might contribute to student performance, and decide where to allocate their resources. To improve this project, more data, both in terms of features and number of datapoints, is needed. Additionally, there are several more types of models that can be implemented and tested.

1 Introduction

How can certain factors contribute to overall student performance? This is what my teammates and I were asking each other when we decided to focus on building a tool that could be used to help improve student performance. Improving student performance and graduation rates is a longstanding concern, given the typical negative effects of ~~a~~ dropping out of high school. Negative outcomes related to dropping out of high school include a median income lower than for people who have a high school degree or above (Trends in High School Dropout and Completion Rates in the United States, n.d.)

Understanding how to prevent more students from dropping out of high school is important because removes some career obstacles, as well as an opportunity to pursue higher education.

2 Related Work

A research paper called "Predicting students' graduation outcomes through support vector machines" focused on describing and proposing a machine learning algorithm built using support vector machines in order to predict if students were to graduate higher education institutions on time (Pang et al., 2017). The model's intended use was as a tool to identify students who might need extra help. The model described in this paper is similar to the models that my team and I trained, as the focus is on predicting on-time graduation. However, their predictive model is a classifier, as it predicts whether or not a student is going to graduate on time. In contrast, the models that my team and I trained were regression models, predicting graduation rate. Additionally, the proposed model gives predictions for individual students, while the models that my team tested are predictions for entire high schools.

A research article called "Predicting first-time-in-college students' degree completion outcomes" detailed a machine learning algorithm that predicted if and when college students would graduate (Demeter et al., 2022). The algorithm that was chosen to accomplish this was random forest, one of the algorithms that my team had trained and tested. One difference from the models my team used is that the random forest model described in the paper was a classifier, as the model they ultimately chose predicted if a student would graduate, and if they would graduate on time. Additionally, the researchers' model is designed to predict the outcomes of single students. This is different from the models my team used, as the models we used were regressors that predicted on-time graduation rate of entire schools.

3 Process Design

At the very beginning of this project, we chose datasets that we wanted to work with, and then combined and cleaned that data. From there, we chose models that we were fairly familiar so that we could set a baseline to compare to more complicated models. Finally, we fine-tuned models, and combined a few to see if that would improve performance.

3.1 Choosing Datasets

After my group had decided we were going to build models that predict graduation rate, we had to choose what features we would be feeding into those models. All of the datasets we used were from the Virginia Department of Education website. We imported and used five different datasets from this website.

For one of our features, we had decided to use the free and reduced lunch eligibility rate of each high school, as we thought that was one way to assess student need. One dataset was imported because it included the four-year graduation rate of each high school in Virginia, which is the output variable that trained models output. Another dataset included a feature on the number of students enrolled in each school. This data was an important aspect in the feature engineering.

We decided it would be useful to understand how many students were involved in advanced programs. Features included in the advanced program dataset were the number of students enrolled in the Governor's School academy, as well as breakdowns of how many were enrolled in STEM or health academy. Other features included were the number of students taking AP classes, number of students taking AP exams, seniors enrolled in the IB program, and seniors awarded IB diplomas. We decided that each of the features included in the dataset would be valuable because it covered a variety of different advanced programs that a student could be enrolled in.

Finally, the fifth dataset we imported included data on suspension. The specific features we used were the total number of days missed due to suspension for male students, and the number of days missed due to suspension for female students.

3.2 Feature Engineering and Data Cleaning

The python program that includes all of the code for this project was written in Google Colab.

The features containing data on suspension, number of days missed due to suspension for males and number of days missed due to suspension for females, were added together to create another column: total number of days missed due to suspension.

Additionally, several features were divided by the total number of students enrolled to take into account how higher values could be due to a larger student population. All the features from the advanced program enrollment dataset and the suspension dataset were divided by number of students enrolled at the school for the same reason.

After this, the dataset was split into a training set and a testing set. The training set was then ready for data cleaning. This involved normalizing the data, as well using an imputer.

3.3 Model Training and Tuning

My team chose three models to use as a baseline: a linear regressor, a decision tree regressor, and a random forest regressor. All of the models we used were from the scikit learn library. We chose these models because we were already somewhat familiar with them from previous assignments. These were some of the first models we had worked with in class, and we thought it would be interesting to compare their performance with newer and more unfamiliar models.

Using the grid search provided with the scikit learn library, we tuned the hyperparameters of the decision tree and random forest regressors. We used the splitter and max features hyperparameters in the grid search for the decision tree regressor. We used the number of estimators and max features hyperparameters in the grid search for the decision tree regressor.

Then, we created and experimented with several neural networks that acted as regressors. We made these models using the keras API from the TensorFlow library. First we implemented a simple neural network with one hidden layer. Then, we experimented with number of layers, activation function, neurons in each layer, and the learning rate of the optimizer in an attempt to achieve a lower RMSE. We then experimented with wide and deep neural networks, altering activation function, number of layers and their neurons, optimizer learning rate, and number of epochs.

3.4 Combining Models

At this point, the best performing model was the random forest regressor. We decided to use a voting regressor that averaged the predictions of the initial three regressor models, the linear regressor, decision tree regressors, and random forest regressor. This is because they tend to perform slightly better than the best performing model. We used the voting regressor that is a part of the scikit learn library. We then experimented with applying different weights to each model's predictions, in an attempt to lower the RMSE even further.

4 Results

These results are based on the most recent time the program was run. The RMSE of the test set for the linear regressor, decision tree regressor, and random forest regressor are 4.78, 6.81, 3.95, respectively. The neural network that achieved the lowest RMSE, a 4.88, was a wide and deep neural network with 4 hidden layers. The lowest overall RMSE, a 3.70, was achieved by a voting regressor that applied a weight of 0.5 to the random forest regressor's predictions, and a weight of 0.25 to the predictions of both the linear regressor and the decision tree regressor.

5 Conclusion

This project is an example of how models could be trained and tested to provide schools with guidance on overall student performance. Having access to feature importance could help schools better understand what could be impacting the performance of the student population. However, it should be understood that because correlation does not equate to causation, significant action should not be made purely based on the models' predictions.

6 Future Work

To be used in the future, the source code for this project should be checked for errors. The data features that are currently being used should be evaluated to see if their removal would improve performance. The Virginia Department of Education website should be evaluated to see if any additional data would be useful to include in the overall dataset. Any one of the models or a combination of models should be chosen to create a software that is accessible for users.

7 UVA Evaluation

CS 1110, or Introduction to Programming, is where I first learned how to program in Python and was introduced to programming concepts that I used in this project. My Linear Algebra and Calculus classes prepared me for Machine Learning, as they are both foundational subjects.

8 Acknowledgments

Thank you to my teammates, Matthew Gerace and Matt Koehler. Thank you to Annie Cao, the CS 4774

teaching assistant who advised us while we were working on this project. Thank you to Professor Rich Nguyen, the instructor for CS 4774.

References

Demeter, E., Dorodchi, M., Al-Hossami, E., Benedict, A., Slattery Walker, L., & Smail, J. (2022). Predicting first-time-in-college students' degree completion outcomes. *Higher Education*. https://doi.org/10.1007/s10734-021-00790-9

Pang, Y., Judd, N., O'Brien, J., & Ben-Avie, M. (2017). Predicting students' graduation outcomes through support vector machines. *2017 IEEE Frontiers in Education Conference (FIE)*. https://doi.org/10.1109/fie.2017.8190666

Trends in High School Dropout and Completion Rates in the United States. (n.d.). National Center For Education Statistics. Retrieved February 26, 2022, from https://nces.ed.gov/programs/dropout/intro.asp#ref2