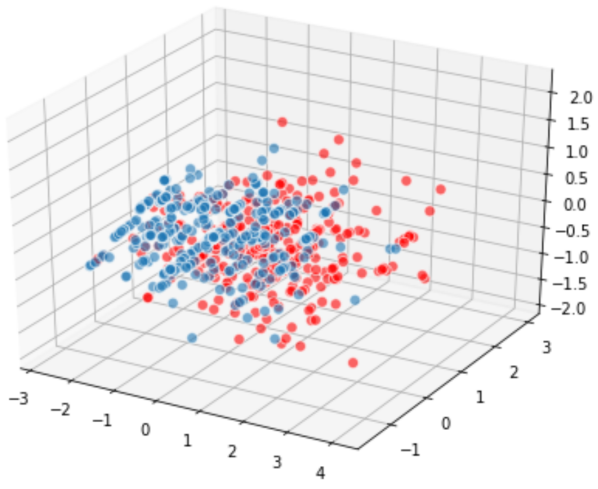
Predicting Student Success

ICS 635: Machine Learning

Final Project Report

Katrina (Johnson) Turner
May 10, 2019



Introduction

The Problem

The current method of registering students for Algebra 2 in local high schools is not effective. Currently, students are registered based on previous math grades and teacher recommendation. Counselors can also a register a student for the course based on any criteria they see fit. High dropout and failure rates indicate that these methods are not optimal.

The Approach

I was able to get data from a local high school I used to teach at and will be testing various classifier models to find the one that best predicts a student's success. We are defining success as finishing the course with a C or better. If a high enough accuracy is achieved, teachers can use this model to aid their decisions at registration time.

The Data Set

The data set contains the last 10 years of math grades for all students at that school as well as a variety of other data such as grade level, GPA, feeder school, teacher, etc. The data points are the students and the features are the data collected on them.

The data wrangling process included dropping unnecessary or unusable columns, imputing missing values from a correlated feature and dropping rows with too many missing values. In the end I was able to keep 1000 data points and the following features:

1. Algebra 1 Grade
2. Geometry Grade
3. Math GPA
4. Overall GPA
5. Grade Level

I also kept the student ID number for the index and of course, Algebra 2 grade as the Target Column. The data was randomly split into 80% training data and 20% test set.

Preprocessing

- Algebra 2 grades of A-C are considered successful and were labeled as 1, while grades below a C, as well as W (Withdrew after deadline) were labeled 0.
- All other grades were converted to number labels. A=4, B=3, C=2, D=1, F/W=0
- Grade level was mapped from [9, 10, 11, 12] to [0, 1, 2, 3] respectively

Methods & Results

Based on some visualizations of the data, and that fact that the data is both categorical and continuous, there were several models I hoped would produce good results. To narrow down my list, I used sklearn to do a GridSearch with 4-fold cross validation to find the best parameters for each, then found the accuracy to quickly compare them. The reason I chose 4-fold is because this would split my 80% training data into 20% bins, so each iteration would train on 60% and validate on 20%. The data set is relatively small so I felt it was good to play it safe. Here are the models and the parameters I ran GridSearch on:

1. K-NearestNeighbors
    - n neighbors
2. Linear Discriminant Analysis
    - solver
    - n components
    - tol (Threshold used for rank estimation in SVD solver.)
3. Decision Tree Classifier
    - criterion
    - splitter
    - max depth
    - min sample split
4. Random Forest Classifier
    - criterion
    - n estimators
    - max depth
    - min sample split

GridSearch Results

I did not expect K Nearest Neighbors to do well, but it is simple and fast to run so I tried it just in case. As expected, it had the lowest GridSearch accuracy at 0.71, so I dropped it.

I expected LDA and RandomForest to perform the best, which they did, but what was surprising to me was that the Decision Tree Classifier had almost the exact same accuracy as Random Forest. Perhaps because I didn't have that many features, having multiple trees did not make much of a difference. LDA performed the best with an initial accuracy of 0.80 with an svd solver, 1 component and 0.001 as the tol. Random Forest was close behind with an accuracy of 0.79 with gini for the criterion, 5 estimators, and 7 as both the max depth and min sample split.

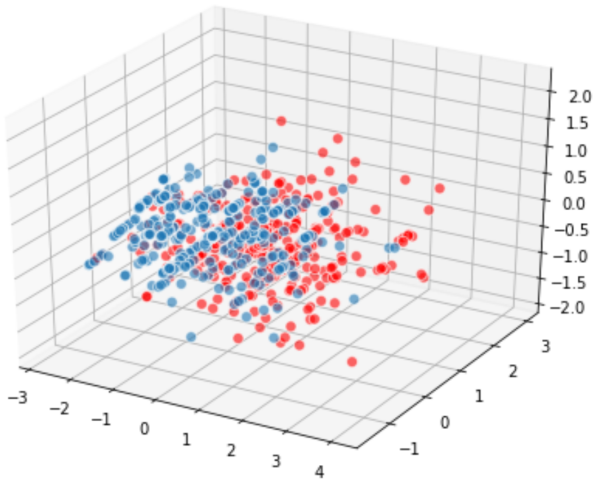


Figure 1: 3D PCA graph of data. Blue=success, Red = Fail

LDA

The plot in Figure 2 does not look quite linearly separable, but you can still see two general clusters. Also, since the dimensionality has been reduced for visualization, it is plausible they are more separable than this. So I decided to focus on using Linear Discriminant Analysis because it's initial accuracy was higher and it seemed an overall good fit for my data. LDA uses a 0-1 loss function for optimization.

I do not have very high dimensional data, with only 5 features and 2 classes, and I used Cross-Validation to optimize the parameters so overfitting should not be a problem.

After fitting and LDA model to my entire training set with the parameters mentioned above, I was able to get an AUROC score of 0.886 on my test set. This already exceeded my expectations, but I wanted to see if I could do better. Looking at the weight vectors and the correlations between features, Geometry and Math GPA had a correlation of 0.89 and their weights were always opposite, one positive, one negative over multiple runs of LDA. Geometry is obviously a contributing factor to Math GPA, so I decided to drop it as a feature because it didn't seem to be helping my model. Also, the topic of Geometry isn't directly related to Algebra 2, so I felt comfortable just keeping Alegbra 1 and Math GPA as the math grades.

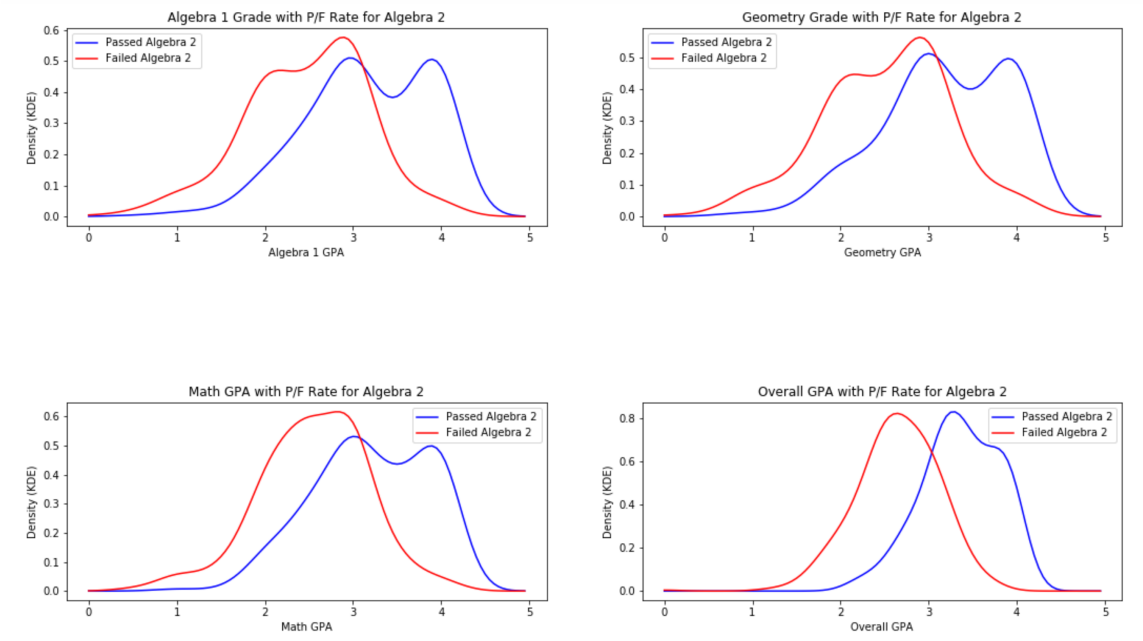


Figure 2: Kernel Density Estimates for various features, separated by Target Success/Fail

Results

I ran Grid Search Cross Validation again on this new data set, but just for LDA. I got the same parameters, but a better accuracy of 0.81 and sure enough, when I repeated the train-test process mentioned above on the entire set, I got an AUROC score of 0.8924. Considering how much the individual features overlap (see Figure 2), I am very happy with this score.

Conclusion

Based on my results, I would say the LDA model is a good predictor of student success in Algebra 2. I was hoping for at least 80% AUROC and I got almost 90%, so I will be presenting my findings to the Math Department at the school I got the data from.

An interesting observation is that Overall GPA had the highest weight in the weight vector, but that feature has never been taken into consideration when registering students before. The previous thinking was that it was only how they performed in math that mattered, but it seems that for Algebra 2, their overall performance as a student is a contributing factor. Which does make sense because it is the first of the more rigorous high school math courses and so good study habits and overall academic skills are important to success.

Another observation that supports a previous theory of mine is that the success rate of students taking Algebra 2 declines as they go up in grade level. The weight for this feature is even negative, about -0.5. Not necessarily because they're younger, but because if the teachers feel they aren't ready, they send them on another path that has them take Algebra 2 at a later time. From past experience, the screening process for students to take Algebra 2 in much more rigorous when they're younger and so a lot of them are more successful. Whereas, once students are in 12th grade, there is no "later time" to take it in high school, so a lot more of them are trying to take it, even if they aren't ready.

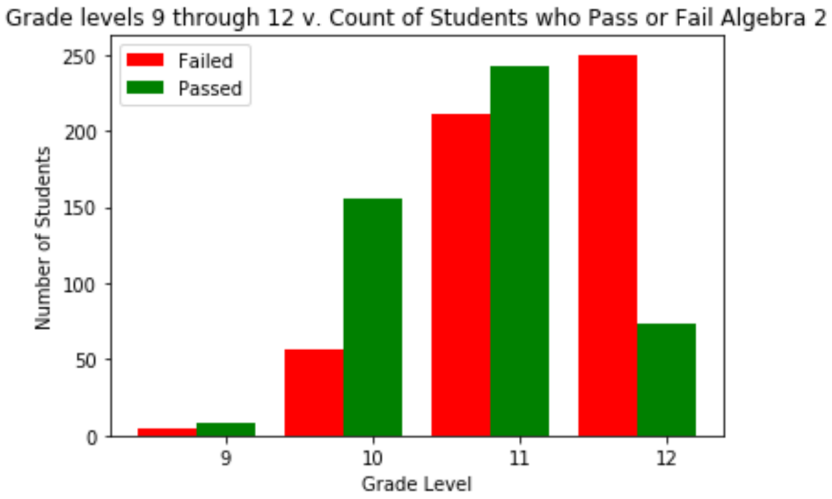


Figure 3: Pass/Fail Counts for Algebra 2 across Grade Level

Next Steps

I would like to develop an easy template for teachers to use to continuously add data to train the model, as well as predict a student's success in Algebra 2. I will of course advise them that this is a tool for registration, not an ultimatum for allowing students to take the course. I would also like to generalize this code so teachers from other schools and disciplines could utilize it.

References

- **Castle High School:** The registrars office provided student data for the project. The Math Department provided insight and will assist with further testing and data collection.
- **Mahdi Belcaid:** Consult for Data Science methods. I also presented the data cleaning and visualization portions of this project for his class.
- **Troy Macris:** Assisted with some of the data visualizations.