Leveraging Machine Learning Techniques to Evaluate Persistence in Engineering

Alexa Catherine Andershock, The University of Tennessee, Knoxville

Lexy Andershock is an incoming Engineering Education Ph.D. student at Virginia Tech. Her research interests include the influence of first-year engineering programs on engineering students, especially relating to major choice and future academic performance. She earned her B.S. in Computer Science from the University of Tennessee, Knoxville.

Baker A. Martin, University of Tennessee at Knoxville

Baker Martin is a Teaching Assistant Professor in Engineering Fundamentals at the University of Tennessee, Knoxville, where he teaches in the first-year engineering program. His research interests include choice and decision making, especially relating to first-year engineering students' major selection. He earned his Ph.D. in Engineering and Science Education from Clemson University, his M.S. in Chemical Engineering from the University of Tennessee, Knoxville, and his B.S. in Chemical Engineering from Virginia Tech.

Rachael Bevill Burns, University of Tennessee at Knoxville

Rachael Bevill Burns is an Assistant Professor of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville. She earned her Ph.D. in Computer Science from the Max Planck Institute for Intelligent Systems (MPI-IS) in Stuttgart, Germany, and her B.S. and M.S. in Biomedical Engineering from the George Washington University in Washington, D.C. Her research lab, STARI, is one of five Max Planck Partner Groups in the US, a prestigious affiliation with the Max Planck Society in Germany promoting cutting-edge, collaborative international research.

Leveraging Machine Learning Techniques to Evaluate Persistence in Engineering

Abstract

Understanding student persistence in college is a valuable resource for creating possible interventions to increase retention, especially in engineering, where attrition rates are high. While much research has been conducted about persistence and graduation in engineering, less is known about how the influences on persistence vary for first-year students. By determining which factors have the strongest correlation with persistence to graduation and the next academic year, we can identify warnings of non-persistence early in students' academic careers. This research utilized the MIDFIELD (Multiple Institution Database for Investigating Engineering Longitudinal Development) dataset, employing machine learning methods to identify the factors that best predict graduation and persistence in the second year. Persistence in the university, engineering program, and degree program were observed. Our results suggest that factors directly affected by a student's decisions in college were stronger predictors of persistence than factors determined before college or demographic characteristics.

Keywords: success, persistence, machine learning

Introduction

Historically, the national shortage of engineers [1] and low student retention in undergraduate engineering programs [2] have motivated research on persistence in engineering degree programs in order to increase the number of engineering graduates. While the factors that affect persistence in engineering have been investigated extensively [3], little is known about how these influences vary for first-year students. Studying these factors will provide students with valuable information early in their engineering coursework and help increase persistence in engineering majors. To measure the relationship between dozens of factors, such as academic and demographic variables, this study utilized machine learning (ML) techniques. Machine learning algorithms are able to learn generalizable patterns from data that would not be possible with traditional statistical-based methods. ML techniques are also more flexible and better suited to capture complex interactions between variables than strict statistical analysis [4]. With the implementation of machine learning algorithms, this study sought to address two questions:

RQ1: How effectively, accurately, and precisely can machine learning techniques be used to identify the most important features influencing the persistence of first year students?

RQ2: How does the feature importance change depending on the projected graduation goal: (1) from the university, (2) in engineering, or (3) in the same major as declared at matriculation?

Understanding how feature importance changes based on each graduation goal can help identify key factors that lead to persistence in engineering, as well as provide students with information that better aligns with their specific academic goals in their undergraduate careers.

Literature Review

The factors that affect persistence in engineering have been widely studied. A study by Bir and Ahn found that pre-entry attributes like high school grade-point average (GPA), academic behaviors during college, and academic integration were significant predictors of persistence in aerospace engineering majors, whereas social integration was not [5]. These findings emphasized the importance of academic performance both before and during college to increase persistence in engineering. Demographic variables, such as sex and race, have also been shown to have impacts on the reasons for leaving [6]. Gender has been shown to account for some differences in persistence, but this is heavily dependent on each institution [7]. However, both male and female persistors were shown to have more academic skills and mathematical reasoning ability than non-persistors [8]. Levels of attrition in science majors due to ethnicity have been shown to be non-significant, whereas preadmission variables, such as high school GPA and SAT scores, accounted for a significant fraction of variance of persistence decisions [9].

Previous research has also utilized machine learning techniques to predict engineering students' persistence to graduation. Machine learning has become an increasingly popular field due to the widespread use of artificial intelligence in predictive modeling. Machine learning models also demonstrate significant improvements in accuracy and scalability compared to traditional predictive models, such as decision trees [10]. Zahedi et al. [11] utilized the Multi-Institutional Database for Investigating Engineering Longitudinal Development, or MIDFIELD [12], to study the persistence of computing students. Many supervised ML algorithms were compared based on accuracy, precision, recall, and F1 score, with the Random Forest (RF) model exhibiting the highest accuracy, precision, and F1 scores. When ranking the predictive variables based on the mean decrease in accuracy of the RF model due to the removal of a variable, cumulative GPA had the most importance on predicting graduation. Similarly, in Osunbumni et al. [13], which explored predictors of persistence to graduation in a college of engineering, many different models were compared, with the RF model having the highest precision, F1, and AUC scores. The study also used three different tests to measure variable influence: Shapley Additive Explanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME), and Mutual Information (MI). In all three tests, the first- and second-term GPAs were the strongest predictors of graduation for engineering students.

This study aims to add to the literature by considering understudied academic and demographic variables, such as academic standing and zip code, evaluating different predictive outcomes and measurements of persistence, and including more institutional data to increase the generalizability of the predictive model.

Theoretical Framework

This study is framed in Astin's Input-Environment-Output (I-E-O) model [14]. This framework argues that proper education assessments must include data that includes "student inputs, student outcomes, and the educational environment to which the student is exposed" [15]. Inputs refer to the qualities or the initial level of developed talent that a student has at the time of entry; outputs refer to desired outcome the researcher is trying to develop within an education program; environment refers to the actual experiences during the educational program. This framework aims to further explain the relationship between environment and student outcomes by necessarily considering student inputs. In Figure 1, the image shows how student inputs can affect both the environment that the student chooses (path A), and that students with similar inputs have correlated effects (path C), meaning that inputs affect the observed relationship between environments and outputs.

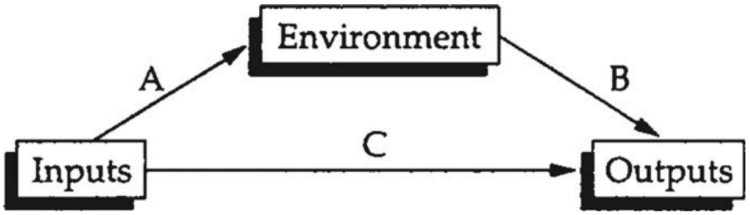


Figure 1: Astin's I-E-O Model

In this study, we classified the variables used in the study according to the definitions above. Input variables are qualities, characteristics, or statistics that students bring with them from high school or prior, such as demographic variables, high school GPA, and SAT scores. Environment variables are variables that are established in college, such as term and cumulative GPA, major choice, and academic standing. Output variables are the outcomes that the machine learning model will predict, including persistence in college, engineering, initial major, and the second year. Table 2 describes each variable used, as well as its classification in the I-E-O model.

Variable Name	Category	Description
age	Input	Age at first matriculation
entry_cip6	Input	First declared major (after first-year program, if applicable)
high_school_gpa	Input	High school final grade point average of last high school attended
home_zip	Input	Five-digit home zip code at time of entry
race	Input	Race as self-reported by the student (Asian, Black, Hispanic/Latinx, International, Native American, Other/Unknown, White)
sat_comp	Input	Composite SAT score (recentered pre-1996)
sex	Input	Sex (Male, Female, Other/Unknown) as self-reported by the student
transfer	Input	Did the student transfer into the institution from another institution (First-Time Transfer, First-Time in College, Transfer from Florida Community College)
us_citizen	Input	Is the student a citizen of the United States of America (Yes, No)
cgpa	Environment	Last reported cumulative GPA for each year of enrollment
hours	Environment	Last reported cumulative number of hours for each year of enrollment
same_cip6	Environment	Indicator that a student has switched from the major declared upon entry
standing	Environment	Last reported academic standing (Good Standing, Academic Warning, Academic Probation, Academic Dismissal, No Credit)
tgpa	Environment	Last reported term GPA for each year of enrollment
egrad	Output	Indicator that the student persisted in engineering
grad	Output	Indicator that the student persisted at the institution
mgrad	Output	Indicator that the student persisted in their first declared engineering major

Table 2: Variable Classification

Data and Methodology

Dataset, Sample Filtering & Demographic Results

This study utilized MIDFIELD to supply the training data for the machine learning models. MIDFIELD is a national dataset containing the student-level records of 1.7 million undergraduate, degree-seeking students at 19 institutions in the U.S. from 1987 through 2018 [12]. The "fix9a" version of MIDFIELD used in this study was compiled on September 15th, 2022. Classification of Instructional Program (CIP) codes are a national taxonomy created by the U.S. Department of Education to classify academic majors. To achieve the final population sample, a filtering process was applied to ensure complete and accurate data. Since the goal of this research was to track persistence of first-year engineering students, any student who had ever formally declared engineering as a major in their first year was initially included. Additional filtering was done to ensure each student had enough data in MIDFIELD to calculate if they could graduate within six years, and any student with data discrepancies between the student, term, and degree tables was excluded. If the student had term data from enrollment in majors other than engineering, this data was dropped. This resulted in a final sample population of 144,507 first-year undergraduate students, representing 16 institutions and 32 engineering majors. Descriptive statistics of the final population are detailed in Tables 3, 4 and 5.

Asian	Black	Hispanic /Latinx	Inter-national	Native American	Other /Unknown	White	Total
9,137	11,589	4,717	11,043	709	3,229	104,083	144,507

Table 3: Sample Demographic by Race

Female	Male	Other /Unknown
27,834	116,667	6

Table 4: Sample Demographic by Sex

First Time in College	First-Time Transfer	Florida CC Transfer
108,414	32,579	3,514

Table 5: Sample Demographic by Transfer Status

The final sample is 72.03% White, 8.02% Black, 7.64% International, 6.32% Asian, 3.26% Hispanic/Latinx, 2.23% Other/Unknown, and 0.49% Native American. The transfer status of the students included in the sample is 75.02% First Time in College, 22.54% First-Time Transfer, and 2.43% Florida Community College Transfer. Additionally, 80.73% of the sample is Male, 19.26% Female, and 0.01% is Unknown.

Variable Inclusion

The goal of this study was to include known predictors of persistence, such as GPA and SAT score, demographic variables, and new variables that have limited representation in previous research, such as home zip code, academic standing, and indicators of whether a student switches their major. These factors were included to determine whether these variables might have more predictive power than traditionally observed academic milestones, with the goal of providing information to students and their supporters to further aid their decision making.

Exploratory Data Analysis

Through initial exploratory data analysis, it was determined that the final dataset was imbalanced. A 50/50 split between persisters and non-persisters would indicate a perfectly balanced dataset. From the first-year student population, 71% graduated college, 70% graduated with an engineering degree, 58% graduated in the same engineering major declared at matriculation, and 88% persisted to the second year, which indicated various degrees of imbalance. Synthetic Minority Over-sampling Technique, or SMOTE [16] was initially implemented to overcome bias that resulted from an unbalanced dataset. SMOTE creates even class distribution by injecting synthetic minority class examples into the dataset; however, it was not ultimately included, due to other model fine-tuning methods producing better model results.

To further understand the dataset, Principal Component Analysis (PCA) [17] was conducted on the training data. PCA aims to reduce the dimensionality of the dataset while preserving as much statistical information as possible. PCA helps to visually reduce correlated variables into distinct sets. In Figure 2, persisters and non-persisters are not visually distinct classes, indicating that the current variables are not major contributors to variance in either group, or that the dataset is not linearly separable. This suggests that even with perfect model implementation, persisters and non-persisters are hard to separate with the current data, and that model shortcomings do not solely rely on adequate fine-tuning or model selection alone.

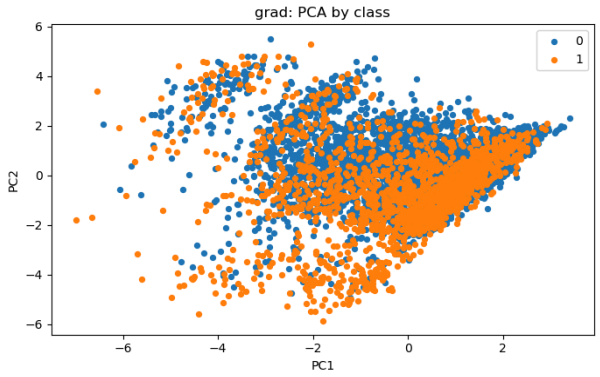


Figure 2: PCA of Training Data for College Graduates. Non-persisting students are represented as orange dots, and persisting students are represented as blue dots.

Imputation & Preparation for ML Model

Not every variable in the final dataset was natively suited as input for an ML model. Some ML models, such as Random Forests, require that for every sample, each feature must have a value; i.e.,, no data can be missing. Since having perfectly complete datasets is unlikely, researchers have proposed several techniques to handle missing data [18]. Imputation is a popular and proven technique used to replace missing data with estimated values. Between pairwise deletion and imputing data, imputation generally yields better classification results [19].

However, a recent study found that in cases where more than 50% of the data is imputed, the root mean squared error of the model increased, indicating a deviation between the imputed data and the original data's values. As the amount of missing data reached 70%, significant disparities between imputed and complete non-imputed data emerged [20]. Therefore, to avoid the potential biases and decreased performance that may result from imputing data with too many missing values, any variable that was missing reports from 50% or more of the sample was eliminated. This included disability status, veteran status, high school, high school rank, high school size, and state indicators. This left four input variables to impute: entry CIP code (missing due to students never declaring a major outside of general engineering), high school GPA, home ZIP code, and composite SAT score (missing due to failure to collect or report). The other four input variables (entry CIP code, home ZIP code, composite SAT score, and high school GPA) had no missing data.

Multiple methods of imputation exist, with the most common categories being single-imputation, multiple imputation, and imputation through machine learning techniques. Multiple imputation is encouraged over single-imputation methods [21]. Machine learning imputation techniques, which rely on classification algorithms and regression trees, such as K-Nearest Neighbors and RF models, fail to account for the uncertainty caused by missing data. However, multivariate imputation by chained equations (MICE) does account for this uncertainty by imputing the data multiple times, then taking an average or random sample from multiple complete datasets [22]. Because of this, MICE was used for imputation.

In order to decide which statistical method to use within MICE, the overall distribution of each variable was determined. Both the entry CIP code and home ZIP code variables are sparse categorical data. When graphing the population of each engineering major, as seen in Figure 3, it exhibits long-tail distribution, with a few majors having high populations while several unique majors have low population counts. For the distribution of zip codes, as seen in Figure 4, several zip codes have low population concentrations, while relatively few zip codes have higher populations. Even when graphed logarithmically, high population counts are difficult to

distinguish from low population counts. These results indicate that predictive means matching should be used for imputing these variables [22].

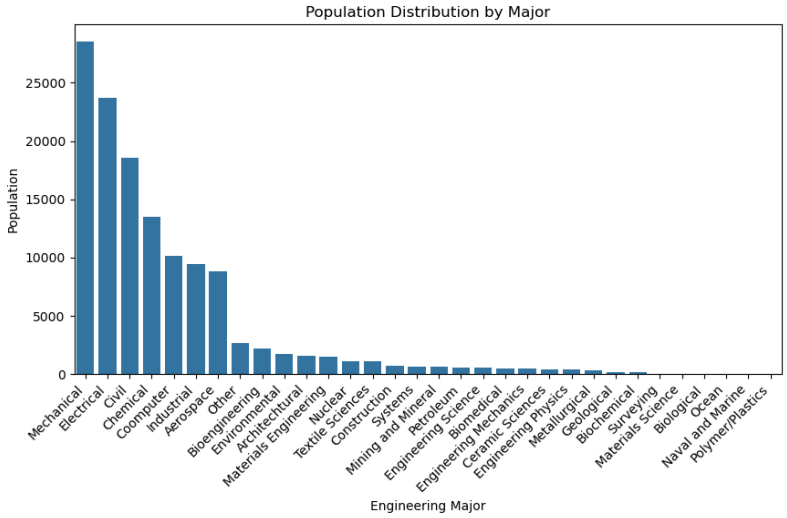


Figure 3: Engineering Population by Major

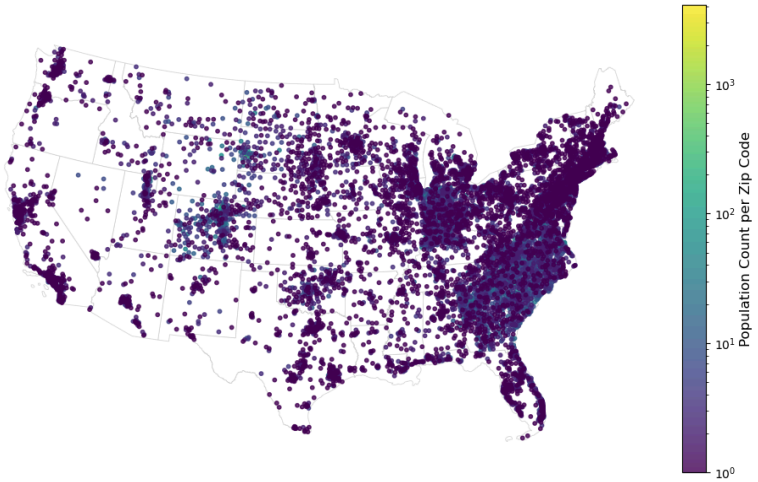


Figure 4: Engineering Population by ZIP Code

For the continuous variables, composite SAT score and high school GPA, Anderson-Darling tests [23] were used to see whether the variables were normally distributed, in which case Bayesian Linear Regression would be used for imputation. For Anderson-Darling tests, the Anderson-Darling statistic (A^2) indicates how much the sample deviates from a specific distribution. To test normal distribution, significance levels at 1%, 2.5%, 5%, 10%, and 15% are reported. At each level, A^2 is compared to its corresponding critical value. If A^2 is less than the critical level, there is not enough evidence to conclude the data does not follow a normal distribution, and the null hypothesis fails to be rejected. However, if A^2 is greater than the critical value, the data likely does not follow a normally distributed pattern, and the null hypothesis is rejected. For composite SAT scores and high school GPA, the null hypothesis is rejected at every

significance level, as seen in Tables 6 and 7, indicating that neither of these variables are normally distributed. Thus, predictive means matching is used for imputation. When corroborating with Quantile-Quantile (Q-Q) plots [24] in Figures 5 and 6, the data clearly does not fall along the straight, diagonal normal distribution line, but instead curves downward at the top and bottom, indicating a non-normal distribution.

Anderson-Darling Critical Values	
Significance Level	Critical Value
15%	0.5760
10%	0.6560
5%	0.7870
2.5%	0.9180
1%	1.0920

Table 6: Anderson-Darling Critical Values for Data Sample

Anderson-Darling Test Results			
Variable	Complete Samples	A^2	Null Hypothesis
sat_comp	123,505	265.5167	Reject
high_school_gpa	115,130	962.1329	Reject

Table 7: Anderson-Darling Results for Continuous Variables

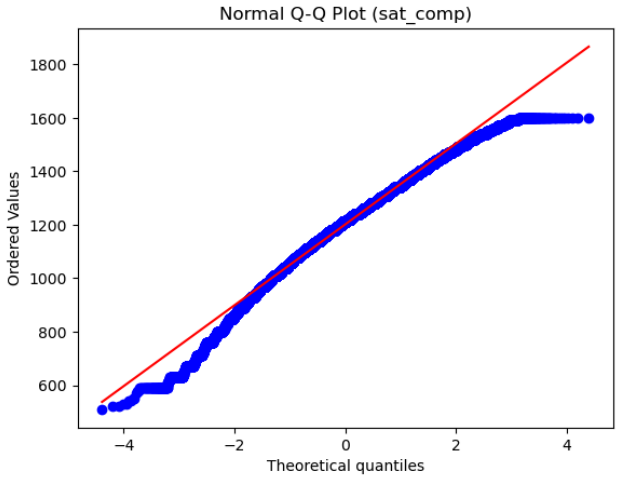


Figure 5: Q-Q Plot for Composite SAT Score

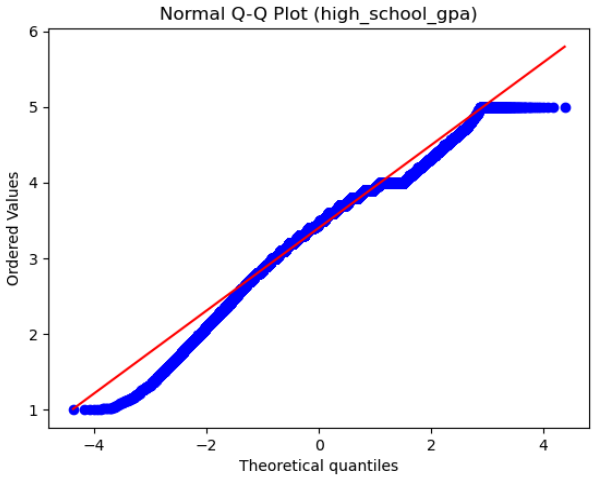


Figure 6: Q-Q Plot for High School GPA

Label Encodings

The machine learning model created for this study was implemented through the popular Python package scikit-learn [25]. In order for non-numeric, i.e., categorical, variables to be compatible with the model, they must first be encoded into numerical values. Ordinal encoding, or assigning numerical values to each variable, is appropriate for data that is intrinsically ordered, such as academic standing. For other variables, such as race, sex, or transfer status, there is no such

inherent order to this data type; thus, one-hot encoding was used for these variables. This helps avoid biases with incorrectly treating categorical variables as ordinal [26]. For academic standing, good standing is considered a high feature value, and academic dismissal is a low feature value, as described in Table 8. For binary encoding, a 'yes' or true value is encoded as a one, and a 'no' or false value is a zero.

Academic Standing Encoding				
Academic Dismissal	Academic Probation	Academic Warning	No Credit Attempted	Good Standing
0	1	2	3	4

Table 8: Academic Standing Encoding Labels

Model Selection & Tuning

As previous studies found that RF models had consistently high accuracy and precision scores when compared to other models [11], [13], this study originally implemented a RF model. However, since the Random Forest model failed to perform well with the unbalanced dataset, an XGBoost [27] model was utilized. XGBoost is an ensemble tree-based method, combining decision trees sequentially, so each new tree corrects the errors of the previous one. It performs particularly well on large, complex datasets, and provides robust techniques to address imbalanced data [28].

Several techniques were applied to tune the model: mainly, class weighting and threshold tuning. Class weighting, as implemented in scikit-learn's XGBoost model [25], uses a ratio of the minority to majority class to assign a higher weight to the minority class. For example, if there are 50 negative samples and 10 positive ones, misclassification of positive samples will be penalized $\frac{50}{10} = 5$ times more than the misclassification of negative samples. In order to increase penalties for mislabeling the minority class, the majority class must be labelled as a zero and the minority class as one in the data. Therefore, persisters had a zero label, and non-persistors had a one label. This is important to keep in mind when analyzing model performance, as it will predict class one, or non-persistence.

Generalized Threshold Shifting (GHOST) threshold tuning [29] was also implemented to increase model performance. Threshold tuning helps address imbalance datasets, as the default threshold of 0.5 for a binary classifier may not always be optimal for the model. For each training set, 200 evenly spaced thresholds ranging from 0.01 to 0.99 were tested. The threshold that maximized Matthew's correlation constant (MCC) [30] on the training set was used for prediction on the test set. The thresholds for the four persistence training sets (predicting institutional graduation, engineering graduation, same-major graduation, and second-year status) were 0.58, 0.59, 0.60, and 0.76.

Analysis

To quantify our results from the XGBoost model, multiple feature importance metrics were used. Information gain was used to measure the reduction in uncertainty when the dataset is split using a specific attribute [31], i.e., how useful a variable is for separating persisters from non-persisters. SHAP plots [32] were used to illustrate how each attribute affects the model's prediction, both in magnitude and direction. Mean SHAP plots show the average impact of a variable on prediction, beeswarm SHAP plots indicate the directionality and shape of feature impact, and waterfall SHAP plots were used to visualize the predictive SHAP values of a single sample [33].

The accuracy, precision, recall, F1 score, ROC-AUC [34] and MCC scores of each model were also recorded, with an emphasis being placed on the MCC score as it is a more reliable predictor of model performance on unbalanced datasets [35], [36]. These tests are further described in Table 9. MCC score ranges from -1 to 1, with -1 indicating complete misclassification and 1 indicating complete classification. Values close to 0.3 are considered weak, and values above 0.5 are considered strong. Confusion matrices, which visualize the total percentage of true positives, true negatives, false positives, and false negatives, were also utilized.

Metrics for Measuring ML Performance			
Metric	Description	Range	Mathematical Representation
Accuracy	Proportion of *correct* predictions (out of all predictions)	$[0, 1]$	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	Measures how many *positive* predictions were correct	$[0, 1]$	$\frac{TP}{TP+FP}$
Recall	Number of actual *positive* cases correctly identified	$[0, 1]$	$\frac{TP}{TP+FN}$
F1 Score	Harmonic mean of precision and recall	$[0, 1]$	$2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$
ROC-AUC	Probability the model ranks a random *positive* example higher than a *negative* one	$[0, 1]$	$\sum_{i=0}^{n-1} (FPR_{i+1} - FPR_i) \cdot \frac{TPR_{i+1}+TPR_i}{2}$
MCC	Evaluates performance of binary classifiers	$[-1, 1]$	$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Table 9: Model Performance Metrics

Results and Discussion

Overall model accuracy, as seen in Table 10 and Figure 8, was moderate. However, accuracy is not a reliable indicator in unbalanced training sets, since high performance can be achieved by simply predicting the majority class. F1 score also has a major drawback in unbalanced datasets, as it does not consider the number of samples correctly classified as negative. ROC-AUC scores are also sensitive to class imbalance. However, MCC overcomes these issues by only producing a high score if the model correctly predicts a majority of positive and negative classes [37]. Thus, only MCC will be considered when discussing model performance in this study.

Model Performance Measurements (Predicting Non-Persistence)				
variable	grad	egrad	mgrad	second-year status
MCC	0.40	0.39	0.39	0.38
Accuracy	0.76	0.76	0.71	0.89
Precision	0.60	0.62	0.74	0.56
Recall	0.53	0.51	0.46	0.35
F1 Score	0.56	0.56	0.57	0.43
ROC-AUC	0.78	0.77	0.76	0.80

Table 10: Model Performance

The MCC score for each target predictor was moderate, indicating the basic rates of the confusion matrix (sensitivity, specificity, precision, and negative predictive value) also had moderate scores. This can be quantified by Figure 8, with the model achieving a true positive score of 0.35 and a true negative rate of 0.96 when classifying non-persistence in college.

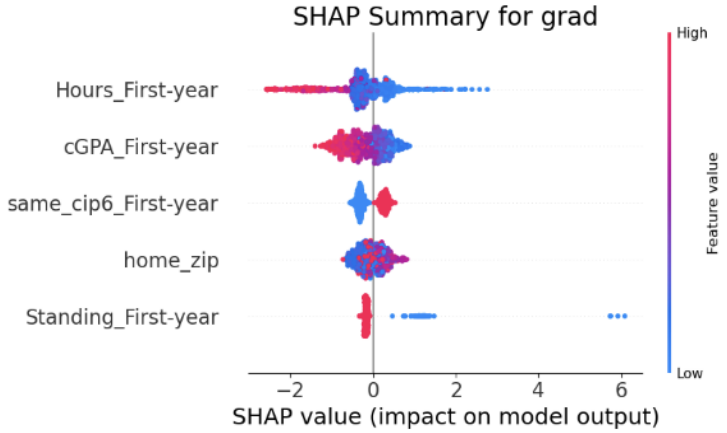


Figure 7: SHAP Plot for Top 5 Indicators of grad Non-Persistence

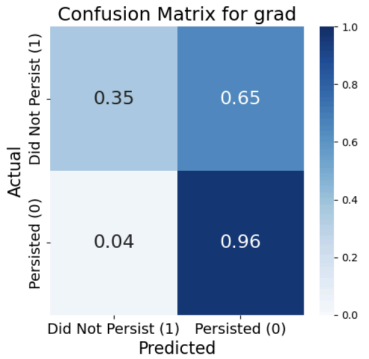


Figure 8: Confusion Matrix for grad

When analyzing the aggregated importance gain of the features, academic standing, race, and transfer status were consistently ranked the most important for class separation across all target predictions, as seen in Figure 9. This is not surprising, as information gain is biased towards features with high cardinality (i.e., few unique values). Higher cardinality allows for more nuanced splits, which helps create more pure classes. As a result, these variables have a disproportionate positive importance compared to variables with smaller dimensions. However, high aggregated gain did not indicate that a variable had high importance in identifying persistence.

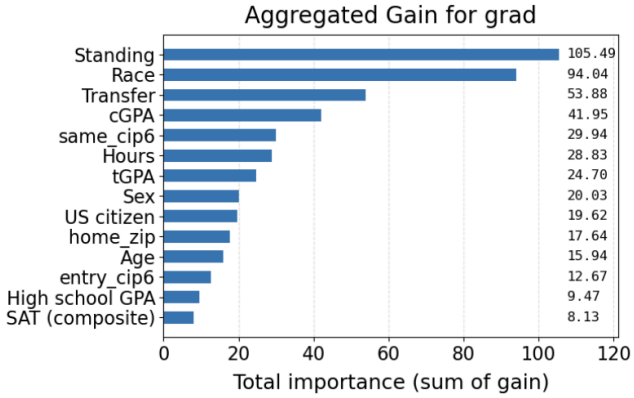


Figure 9: Information Gain for grad

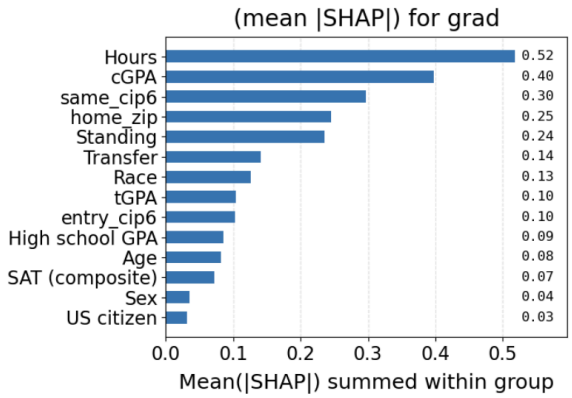


Figure 10: Mean Absolute SHAP for grad

Figures 7 and 10 help explain how important each variable is to the model and how it affects the model's predictions. In Figure 10, cumulative GPA and number of hours all have large impacts on the model's decision-making for each of the four persistence outcomes. Less impactful, but still important variables, include home zip code, transfer status, and race. Composite SAT score, sex, and citizenship status were consistently ranked the least influential variables, as well as high school GPA, to an extent.

There are also considerable differences in each variable's influence when comparing the four persistence outcomes. When predicting persistence in college, students who switched majors by the end of their first year or were still enrolled in a first-year engineering program were more likely to persist to graduation. Zip code was moderately important for determining persistence in both engineering and in a specific major. When predicting persistence in the second year of undergraduate study, the CIP6 code declared at matriculation was very influential, but determining which of the 32 majors were more likely to persist was not able to be clearly represented by the SHAP graph.

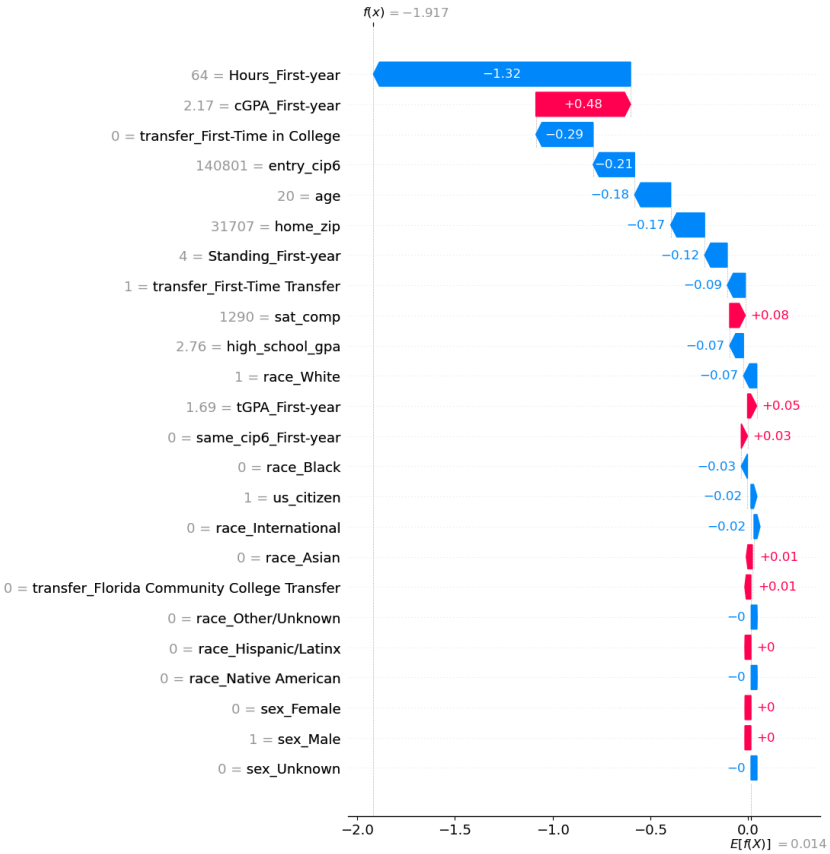


Figure 11: Waterfall plot for grad, showcasing all features

Figure 11 shows how one student's features influence model prediction. The negative values, colored in blue, indicate the features that influenced the model to classify this student as a persister (class 0), whereas the positive values, colored in pink, influenced the model to classify this student as a non-persister (class 1). For this student, their high number of hours makes them more likely to persist to graduation, but having a low cumulative GPA makes them less likely to persist. Some features like race, sex, and citizenship status have almost no impact on the model's decision.

Conclusions

This research reveals that persistence in engineering is not about picking the "best" students out of high school, as high school GPA and composite SAT scores are low predictors of persistence, but about providing students with opportunities to succeed during their first year in college. In reviewing the importance of the home zip code variable, resources could be allocated for students coming from low-income areas to assist them from the moment they get to college. Additionally, for those with an interest in monitoring retention, cumulative GPA is a good indicator to judge the likelihood of persistence in general, which agrees with previous studies [11].

Limitations

This research is useful to determine which variables are important, but not *why* they are important; more information at the institutional level is required to separate correlation from causation in variable analysis. The machine learning model and imputation methods, due to bootstrapping, introduce variability into the results. This is nullified with a consistent seed; however, readers must be discouraged to interpret results with staunch finality.

Future Work

Immediate improvements can be implemented by introducing more granularity and separation in certain variables. For example, a variable that indicates whether students that are enrolled in direct-to-major-matriculation institutions or first-year general engineering programs should be introduced to understand the influence of each, especially since the same CIP6 variable was important for predicting persistence to graduation. Additionally, visualizations should be made to understand how particular engineering majors and zip codes affect persistence, as these were also influential in several models. Further research should also focus on evaluating these effects across multiple years rather than focusing on the first-year alone. Future applications can benefit from utilizing machine learning techniques by creating persistence plans specific to each school, major, and student, which would provide early academic intervention and customized support throughout each student's career.

References

[1] H. Steenkamp, A. L. Nel, and J. Carroll, "Retention of engineering students," in *IEEE Global Engineering Education Conference, EDUCON*, 2017. doi: 10.1109/EDUCON.2017.7942922.

[2] B. N. Geisinger and D. R. Raman, "Why they leave: Understanding student attrition from engineering majors," in *International Journal of Engineering Education*, 2013.

[3] M. W. Ohland, S. D. Sheppard, G. Lichtenstein, O. Eris, D. Chachra, and R. A. Layton, "Persistence, engagement, and migration in engineering programs," *Journal of Engineering Education*, vol. 97, no. 3, 2008, doi: 10.1002/j.2168-9830.2008.tb00978.x.

[4] B. Khosravi *et al.*, "Demystifying Statistics and Machine Learning in Analysis of Structured Tabular Data," *Journal of Arthroplasty*, vol. 38, no. 10, 2023, doi: 10.1016/j.arth.2023.08.045.

[5] D. D. Bir and B. Ahn, "Factors predicting students' persistence and academic success in an aerospace engineering program," 2019.

[6] S. González-Pérez, M. Martínez-Martínez, V. Rey-Paredes, and E. Cifre, "I am done with this! Women dropping out of engineering majors," *Front. Psychol.*, vol. 13, 2022, doi: 10.3389/fpsyg.2022.918439.

[7] M. W. Ohland *et al.*, “Race, gender, and measures of success in engineering education,” *Journal of Engineering Education*, vol. 100, no. 2, 2011, doi: 10.1002/j.2168-9830.2011.tb00012.x.

[8] S. Takahira, D. J. Goodings, and J. P. Byrnes, “Retention and performance of male and female engineering students: An examination of academic and environmental variables,” *Journal of Engineering Education*, vol. 87, no. 3, 1998, doi: 10.1002/j.2168-9830.1998.tb00357.x.

[9] R. Elliott, A. C. Strenta, R. Adair, M. Matier, and J. Scott, “The role of ethnicity in choosing and leaving science in highly selective institutions,” *Res. High. Educ.*, vol. 37, no. 6, 1996, doi: 10.1007/BF01792952.

[10] M. Hatta, W. N. Wahid, F. Yusuf, F. Hidayat, N. A. Santoso, and Q. Aini, “Enhancing Predictive Models in System Development Using Machine Learning Algorithms,” *International Journal of Cyber and IT Service Management*, vol. 4, no. 2, pp. 80–87, Oct. 2024, doi: 10.34306/ijcitsm.v4i2.159.

[11] L. Zahedi, S. J. Lunn, S. Pouyanfar, M. S. Ross, and M. W. Ohland, “Leveraging machine learning techniques to analyze computing persistence in undergraduate programs,” in *ASEE Annual Conference and Exposition, Conference Proceedings*, 2020. doi: 10.18260/1-2--34921.

[12] M. W. Ohland and R. A. Long, “The multiple-institution database for investigating engineering longitudinal development: An experiential case study of data sharing and Reuse,” *Adv. Eng. Educ.*, vol. 5, no. 2, 2016.

[13] I. Osunbunmi *et al.*, “Artificial intelligence in engineering education research: Using machine learning models to predict undergraduate engineering students’ persistence to graduation,” *Journal of Engineering Education*, vol. 114, no. 4, 2025, doi: 10.1002/jee.70034.

[14] A. W. Astin, “The Methodology of Research on College Impact, Part One,” *Sociol. Educ.*, vol. 43, no. 3, 1970, doi: 10.2307/2112065.

[15] A. W. Astin and A. L. Antonio, “A Conceptual Model for Assessment,” in *Assessment for Excellence*, 2025. doi: 10.5040/9798216383673.ch-002.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, 2002, doi: 10.1613/jair.953.

[17] I. T. Jollife and J. Cadima, “Principal component analysis: A review and recent developments,” 2016. doi: 10.1098/rsta.2015.0202.

[18] H. Kang, “The prevention and handling of the missing data,” 2013. doi: 10.4097/kjae.2013.64.5.402.

[19] A. Farhangfar, L. Kurgan, and J. Dy, “Impact of imputation of missing values on classification error for discrete data,” *Pattern Recognit.*, vol. 41, no. 12, 2008, doi: 10.1016/j.patcog.2008.05.019.

[20] K. P. Junaid, T. Kiran, M. Gupta, K. Kishore, and S. Siwatch, “How much missing data is too much to impute for longitudinal health indicators? A preliminary guideline for the choice of the extent of missing proportion to impute with multiple imputation by chained equations,” *Popul. Health Metr.*, vol. 23, no. 1, 2025, doi: 10.1186/s12963-025-00364-2.

[21] J. R. Dettori, D. C. Norvell, and J. R. Chapman, “The Sin of Missing Data: Is All Forgiven by Way of Imputation?,” *Global Spine J.*, vol. 8, no. 8, 2018, doi: 10.1177/2192568218811922.

[22] S. van Buuren, *Flexible Imputation of Missing Data*. 2012. doi: 10.1201/b11826.

[23] T. W. Anderson and D. A. Darling, “A Test of Goodness of Fit,” *J. Am. Stat. Assoc.*, vol. 49, no. 268, 1954, doi: 10.1080/01621459.1954.10501232.

[24] M. B. Wilk and R. Gnanadesikan, “Probability plotting methods for the analysis of data.,” *Biometrika*, vol. 55, no. 1, 1968, doi: 10.1093/biomet/55.1.1.

[25] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, 2011.

[26] J. A. Samuels, “One-Hot Encoding and Two-Hot Encoding: An Introduction,” *Imperial College*, 2024.

[27] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. doi: 10.1145/2939672.2939785.

[28] M. Wiens, A. Verone-Boyle, N. Henscheid, J. T. Podichetty, and J. Burton, “A Tutorial and Use Case Example of the eXtreme Gradient Boosting (XGBoost) Artificial Intelligence Algorithm for Drug Development Applications,” *Clin. Transl. Sci.*, vol. 18, no. 3, 2025, doi: 10.1111/cts.70172.

[29] C. Esposito, G. A. Landrum, N. Schneider, N. Stiefl, and S. Riniker, “GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning,” 2021. doi: 10.1021/acs.jcim.1c00160.

[30] B. W. Matthews, “Comparison of the predicted and observed secondary structure of T4 phage lysozyme,” *BBA - Protein Structure*, vol. 405, no. 2, 1975, doi: 10.1016/0005-2795(75)90109-9.

[31] E. Epstein, N. Nallapareddy, and S. Ray, “On the Relationship between Feature Selection Metrics and Accuracy,” *Entropy*, vol. 25, no. 12, 2023, doi: 10.3390/e25121646.

[32] S. M. Lundberg and S. I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, 2017.

[33] A. V. Ponce-Bobadilla, V. Schmitt, C. S. Maier, S. Mensing, and S. Stodtmann, “Practical guide to SHAP analysis: Explaining supervised machine learning model predictions in drug development,” *Clin. Transl. Sci.*, vol. 17, no. 11, 2024, doi: 10.1111/cts.70056.

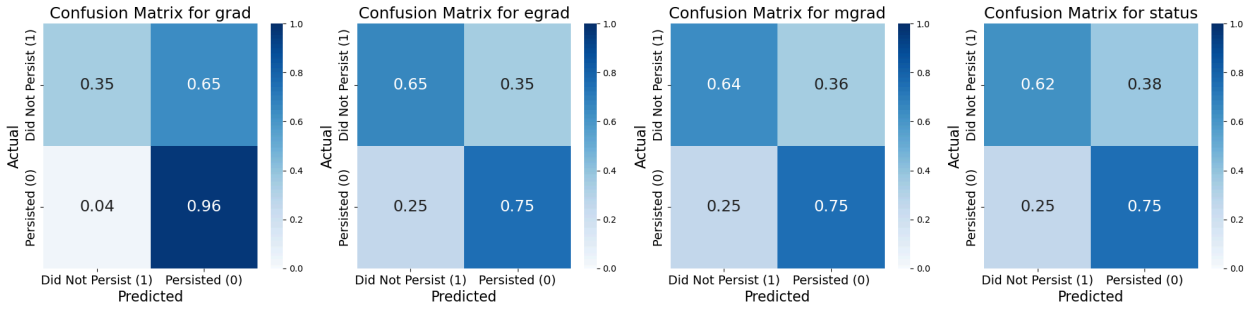
[34] D. Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,” *Mach. Learn. Technol.*, vol. 2, Jan. 2008.

[35] D. Chicco, N. Tötsch, and G. Jurman, “The matthews correlation coefficient (Mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation,” *BioData Min.*, vol. 14, 2021, doi: 10.1186/s13040-021-00244-z.

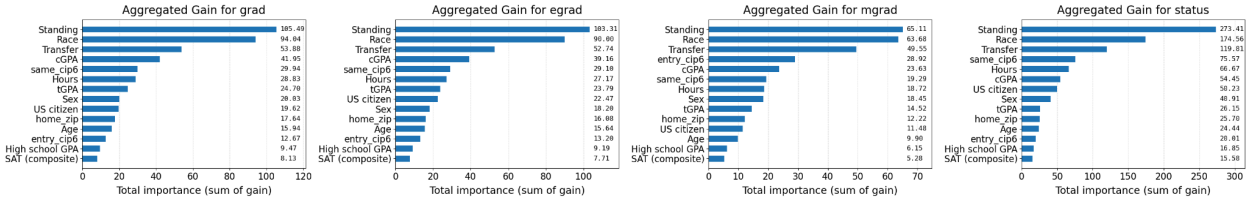
[36] D. Chicco and G. Jurman, “The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification,” *BioData Min.*, vol. 16, no. 1, 2023, doi: 10.1186/s13040-023-00322-4.

[37] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, 2020, doi: 10.1186/s12864-019-6413-7.

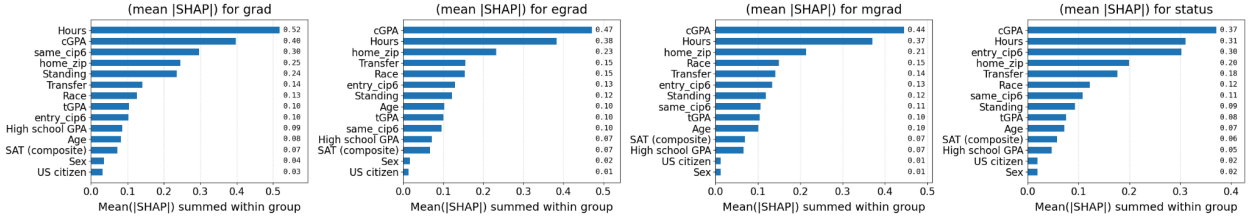
Appendix



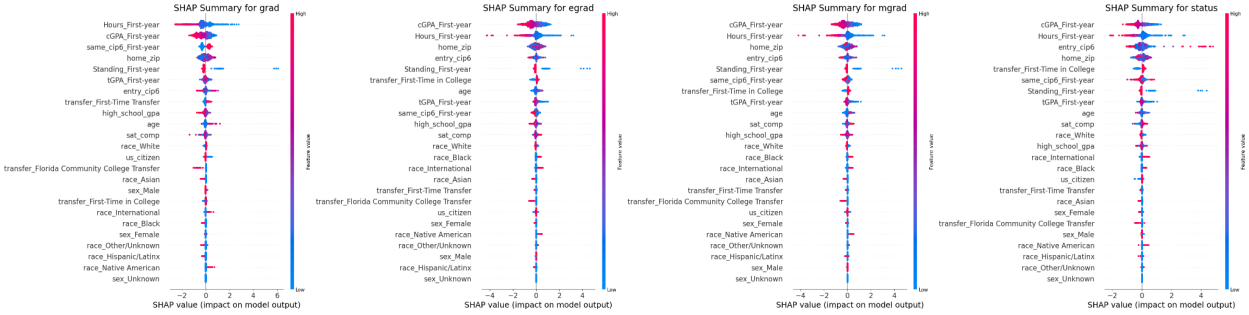
Appendix Figure 1: Confusion Matrices for Target Variables



Appendix Figure 2: Feature Importance Variables



Appendix Figure 3: Mean Absolute SHAP Values



Appendix Figure 4: SHAP Magnitude Plots