Predicting Academic Risk in Computer Science and Engineering Courses a ML Model Evaluation

Daniel Getty, Mason Turner, Matthew Nickols, Halil Bisgin PH.D
Department of Computer Science
University of Michigan-Flint
Email: {dgetty, machtu, mnickols, bisgin}@umich.edu

Abstract*—*Contribution: **Student dropout has been a major concern for higher education as it can lead to multi-faceted issues including financial problems and low academic performance for both students and institutions. Dropouts pose an existential threat, especially for institutions which offer Massive Open Online Courses (MOOCs) [1]. Therefore, detecting the early signals of academic risk either at course or college level and studying the underlying factors towards failure and success have become a crucial task. While existing studies have successfully elucidated certain factors within specific contexts, there remain unidentified contributors that warrant further research. In this paper we aim to investigate this challenging problem specifically for the computer science and engineering students at a regional university by leveraging institutional sources such as the Canvas Learning Management System (LMS) [2], and the Banner Enterprise Resource Planning (ERP) [3] data.**

-*Background:* **LMS data offers valuable insights into student engagement and performance as it can record all interactions. Recent studies have shown the utility of LMS logs, which can include student activities such as syllabus view, recording views, assignment attempts, assignment scores, and faculty comments for identifying at-risk online students [4]. The LMS offers time variant course assignment activity data which may also vary depending on the modality of the courses, i.e., online, mixed-mode, etc. On the other hand, the ERP system provides a wealth of past time-invariant data, including demographic, standardized test scores, incoming high school and previous college GPA scores. It further records time-variant Semester Activity data from students' prior academic outcomes and makes several vectors like current college GPA's, and registration holds ,final course grade, Drop/Fail/Withdrawal (DFW), completed indicator, satisfactory unsatisfactory, and discredited grade distribution available.**

-*Research Question:* **What ML model(s) will be most accurate when compared; using the same student academic features determined to have the greatest feature importance?**

-*Methodology:* **An in-depth analysis of variables from aforementioned data sources, which we will combine, to identify the best performing ML model(s). These models include Decision Tree Classifier [5], K-Nearest Neighbor (KNN) [6], Random Forest Classifier [7], Support Vector Machine (SVM) [8] and Naive Bayes [9]. Namely, we will be using both time-varying course activity data and past time-invariant data to predict satisfactory and unsatisfactory outcomes for students.**

-*Findings:* **In this project, we explored various machine learning models to predict student academic risk. We used data from the University of Michigan-Flint to train and test the models. The dependent variable or class was "satisfactory unsatisfactory binary". The results show that the Random Forest model performed the best in terms of accuracy and precision. The testing features include Overall GPA, High School GPA, SAT Combined, Age, and Previous College GPA. These were chosen because the three models KNN, SVM, and Naive Bayes would not produce a confusion matrix with larger feature sets within 15 min. This may or may not have been due to the computational hardware available in the study.**

***Index Terms*—Academic Risk Prediction, Learning Management System, Enterprise Resource Planning, Machine Learning, Risk Score**

I. Introduction

Similar to the commonalities found in the body of literature studied. We will focus on many of the same processes. We will analyze feature importance from the student data extracted. This will become the basis for feature selection. We will evaluate five popular machine learning algorithms. These include Decision Tree Classifier [5], Random Forest Classifier [7], K-Nearest Neighbors (KNN) [6], Support Vector Machines [8], and Naive Bayes [9]. These five models will be compared based on Accuracy, Sensitivity, Specificity, and The Area Under Receiver Operating Characteristic (ROC) curve. Lastly we will explore knowledge gained to do a deeper dive into the chosen ML algorithm(s) and develop an expanded feature list for future research.

II. Related Work

-Feature Selection: Dalipi and colleagues discussed the importance of feature selection and engineering in dropout prediction for massive open online courses (MOOCs) [10]. They highlighted the need to identify relevant features from large datasets containing student information, engagement metrics, and performance indicators. Feature engineering techniques such as dimensionality reduction and transformation were employed to extract meaningful features that improve the predictive accuracy of dropout models. Costa et al. developed a predictive model for dropout in computer science courses, with an emphasis on feature selection and engineering [11]. They identified relevant features from student data, including demographics, academic performance, and engagement metrics. Feature engineering techniques such as normalization, scaling, and encoding were applied to preprocess the data and transform it into a suitable format for machine learning algorithms. Lykourentzou et al. conducted feature selection to identify predictors of dropout in e-learning courses [12]. They analyzed various student attributes, including demographics, academic history, and engagement metrics, to determine the most influential factors associated with dropout risk. Tamada and colleagues performed feature engineering on data from

learning management systems (LMS) logs to predict dropout in technical courses [4]. They extracted features such as login frequency, page views, and participation in discussions from the raw log data and engineered additional features to capture student engagement and behavior patterns. These engineered features enhanced the predictive accuracy of their dropout prediction model. Feature selection to analyze CS and EGR student data would leverage domain specific features like course grades and standardized tests related to these fields. For example CS and EGR studenets would need to be proficient in reading, mathematics, programming, and physics to name a few. In this study we plan to identify and emphasis these features as well as course performance from the LMS data. Aulck and colleagues discuss building several machine learning models using various feature sets, including demographic information, academic performance, financial aid records, and student behavior data [13].

-*Predictive Model Evaluation:* Predicting dropout in computer science courses, Costa et al. evaluated the performance of their predictive model using metrics such as accuracy, precision, recall, and area under the curve (AUC) [11]. These metrics provided insights into the model's ability to correctly classify students as either at-risk or not at-risk of dropping out. Additionally, the researchers conducted cross-validation to assess the model's generalizability across different datasets and settings. Lykourentzou et al. assessed the performance of their dropout prediction models in e-learning courses using similar evaluation metrics, including accuracy, precision, and recall [12]. They also examined the area under the receiver operating characteristic curve (AUC-ROC) to measure the model's discrimination ability. By evaluating the models against these metrics, the researchers gauged their effectiveness in identifying students at risk of dropping out. Prenkaj and colleagues conducted a survey of machine learning approaches for student dropout prediction in online courses [14]. They discussed various evaluation techniques, including holdout validation, k-fold cross-validation, and bootstrapping, which are commonly used to assess the performance of predictive models. The survey provided insights into the strengths and limitations of different evaluation methods and their applicability in dropout prediction research. Tamada and colleagues evaluated the performance of their dropout prediction model for technical courses using metrics such as accuracy, precision, and recall [4]. They also analyzed the model's receiver operating characteristic (ROC) curve to assess its trade-off between true positive and false positive rates. By examining these evaluation metrics, the researchers determined the model's effectiveness in identifying students at risk of dropping out. Aulck et al. evaluate logistic regression, random forest, and gradient boosting classifiers in predicting student dropout based on various metrics such as accuracy, precision, recall, and F1-score. [13]

-*Machine Learning Techniques:* Dropout prediction in e-learning courses, Lykourentzou et al. utilized machine learning algorithms such as decision trees, support vector machines, and artificial neural networks [12]. These algorithms were applied to features extracted from student data, including demographics, academic performance, and engagement metrics, to develop predictive models capable of identifying students at risk of dropping out. Dalipi and colleagues reviewed machine learning techniques for predicting dropout in MOOCs [10]. They discussed the application of various algorithms such as support vector machines, hidden Markov models, and ensemble methods in analyzing MOOC data to predict student dropout. The review highlighted the strengths and limitations of each technique and provided insights into their effectiveness in different contexts. Prenkaj et al. discussed techniques, including decision trees, logistic regression, random forests, and neural networks [14]. The survey highlighted the diversity of machine learning methods used in dropout prediction research and provided insights into their comparative effectiveness and applicability.

-*Early Intervention:* Dropout prediction in university-level courses, Sandoval-Palis et al. emphasized the importance of timely intervention to support at-risk students [15]. By predicting dropout early in the course, educators can implement targeted interventions such as academic support programs, mentoring, and counseling to address students' needs and prevent dropout. Tamada and colleagues focused on predicting dropout in technical courses using data from learning management systems (LMS) logs [4]. They highlighted the potential for early intervention based on predictive models that identify students at risk of dropping out. By intervening early, educators can provide timely support and guidance to at-risk students, increasing their chances of success and retention in the course. Costa et al. developed a predictive model for dropout in computer science courses, emphasizing the importance of early identification and intervention [11]. By identifying students at risk of dropping out early in the semester, educators can implement targeted interventions such as academic advising, tutoring, and peer mentoring to address students' challenges and improve their chances of completing the course successfully. Lykourentzou et al. discussed the potential for early intervention in e-learning courses based on predictive models for dropout [12]. By identifying at-risk students early in the course, educators can provide personalized support and resources to help them overcome challenges and stay engaged in their studies. By identifying students who are at risk of dropping out using machine learning models, institutions can take proactive steps to help them stay enrolled, such as providing academic support or financial assistance [13].

-*Personalized Interventions:* Del Bonifro and colleagues explored the potential for personalized interventions based on predictive models for student dropout [16]. They discussed how predictive models can identify individual students at risk of dropping out based on their unique characteristics, behaviors, and performance indicators. By tailoring interventions to meet the specific needs of these students, educators can provide targeted support and resources to improve their chances of success and retention. Tamada and colleagues focused on predicting dropout in technical courses using

data from learning management systems (LMS) logs [4]. They highlighted the potential for personalized interventions based on predictive models that identify students at risk of dropping out. By understanding the specific challenges and barriers faced by individual students, educators can develop customized support plans and resources to address their needs and improve their likelihood of completing the course successfully. Costa et al. developed a predictive model for dropout in computer science courses, emphasizing the importance of personalized interventions in supporting at-risk students [11]. By identifying students at risk of dropping out early in the semester, educators can implement targeted interventions such as academic advising, tutoring, and mentoring to address their individual needs and challenges. Lykourentzou et al. conducted a study on dropout prediction in e-learning courses, considering the potential for personalized interventions based on predictive models [12]. By identifying students at risk of dropping out and understanding the factors contributing to their disengagement, educators can develop customized support strategies and resources to re-engage these students and improve their chances of success in the course. Aulk et al. suggests that personalized interventions may be more effective in preventing student dropout compared to one-size-fits-all solutions. For example, if a model identifies academic performance as a significant factor contributing to student dropout for a particular group of students, institutions can provide targeted academic support and resources to those students to help them succeed. [13].

III. Methodology

Data was pulled from the University of Michigan-Flint's Canvas Learning Management System (LMS), extracted via the Unison Data Platform (UDP) and then loaded into UM-Flint's data warehouse. Data from the Banner Enterprise Resource Planning (ERP) System was also loaded into the data warehouse. This Canvas LMS and Banner ERP data is consolidated in the data Warehouse where it is transformed into de-normalized datasets. Some of the data pre-processing is done during this transformation process prior to the export of CSV files for analysis. As mentioned, blending of multi-dimensional Data was performed using SQL scripts in the data warehouse. We have blended Student Course, Demographic, Incoming Scores, Current GPA, and Holds. Dimensional Features were identified with the name ending in (bin, cat, ord) to aid in the data type identification process.

The table with the de-normalized data being reviewed is referred to as the "Time Variant Student Course Assignment Activity" (TVCAA.csv) table. The grain is "one row/student/term/course/assignment." Some values that are representative of the student or term level will repeat in the dataset. Other tables without as fine of a grain were derived for reference. "Time Varying Semester Activity" (TVSA.csv) contains features that change by semester (i.e. Age, Overall GPA, etc). "Past Time Invariant Student" (PTIS.csv) holds features that are immutable to the student (i.e. sex, ethnicity, etc).

The aforementioned de-identified CSV files were extracted from the UM-Flint data warehouse then read into Pandas dataframes using python scripts contained within RMarkdown (.rmd) files to manage literate statistical programming needs. After extraction, test scores and GPA NULL values were replaced with the mean values of their respective columns. Comment feature data points that were NULL were replaced with "string neutral" for sentiment analysis. A python function called "one-hot encoding", a categorical feature conversion tool, was applied to the categorical columns that were determined to be used as independent (X) variables in the feature importance selection. The dependent (Y) variable to be put in a python series was determined as "Satisfactory Unsatisfactory Bin"; A binary variable which identified whether a student had a satisfactory (D or higher) or unsatisfactory (D- or lower) coded as 1 and 0 respectively for the outcome of the student's course.

Five machine learning models were selected for comparison in this study - Decision Tree Classifier, K-Nearest Neighbor (KNN), Random Forest Classifier, Support Vector Machine (SVM) and Naive Bayes. The dataframes for X and Y were split into training and test data via python's train-test-split method. Feature importance was calculated for the X variables using the methods provided by the DecisionTreeClassifier() and RandomForestClassifier() objects in python. Variables with insignificant feature importance were removed from the model, and the process repeated and the X variables were then chosen. The selected features include Overall GPA, High School GPA, SAT Combined, Age, and Previous College GPA. These were chosen because the other three models including KNN, SVM, and NB models struggled to produce a confusion matrix with larger feature sets.

For each of the five models, a for-loop was run to determine the best random state to minimize the Mean Absolute Error (MAE). The Models were then constructed using these determined random states and fit on the training datasets, and predictions were made on the test dataset. ML Models were tested on multiple computers including windows 10 and 11 as well as a computer running a Debian 10 Linux distribution. The most robust PC is a windows 11 OS running on an Intel Core i7-10750H CPU @ 2.6GHZ with 12 logical cores, 64GB of DDR4 3.2GHz system memory, an NVIDIA GeForce RTX 2060 GPU running 240 tensor cores, and 6GB of DDR6 memory running at 1.75GHz. The number of true negative, false positive, false negative and true positive results in the test set were determined using the confusion-matrix method. Sensitivity and specificity of the models were determined using these numbers. For the Decision Tree and Random Forest models, visualizations were made of the trees using the graphviz python library method. Finally, the Receiver Operating Characteristics - Area Under the Curve (ROC-AUC) for each model was plotted, using false positive rates (FPR), true positive rates (TPR) and thresholds extracted. Using Accuracy and ROC-AUC as the most important determining metrics; the Random Forest Model was chosen.

IV. Results

The five models were evaluated on their accuracy, area under the curve (AUC), sensitivity and specificity. The Random Forest Classifier was determined to be the most accurate model out of the five, with 0.95 accuracy, 0.98 AUC, 0.97 sensitivity and 0.79 specificity.

Model	Accuracy	Sensitivity	Specificity	AUC
Decision Tree Classifier	0.95	0.98	0.77	0.89
Random Forest	0.95	0.97	0.79	0.98
Support Vector Machines	0.9	0.99	0.27	0.9
K-Nearest Neighbor	0.94	0.97	0.8	0.98
Naïve Bayes	0.85	0.86	0.72	0.87

Fig. 1: ML Model Results Chart

Given that the dependent variable "Satisfactory Unsatisfactory Bin" was binary and the model had 0.95 accuracy, the MAE was 0.05. With unsatisfactory classified as negative and satisfactory as positive, the confusion matrix was determined to have 1,631 true negatives, 425 false positives, 385 false negatives, and 13,479 true positives. The sensitivity and specificity were calculated using these numbers.

Confusion Matrix	Value
True Negative	1631
False Positive	425
False Negative	385
True Positive	13479

Fig. 2: Random Forest Confusion Matrix

Comparing multiple models required the reduction of feature size, and so five numeric variables with the most feature importance were chosen as the dependent variables. Those variables were, in descending order of feature importance, see table below:

	Feature	Importance
0	UMF_OVERALL_GPA	0.585195
1	HSCH_GPA	0.159754
2	SAT_TOTAL_COMBINED	0.094901
3	AGE	0.087309
4	PCOL_GPA	0.072840

Fig. 3: Feature Importance

The model's ROC-AUC of 0.98 is exceptionally high and indicates a highly accurate model as shown in the following graph.

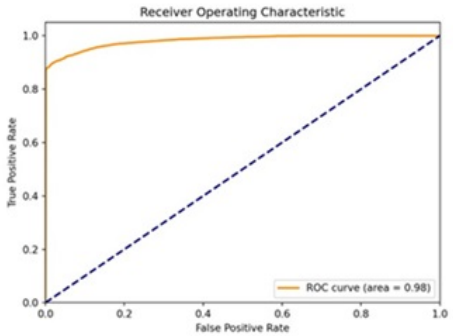

Fig. 4: ROC-AUC

V. Discussion

This study has shown that the Random Forest Classifier has exceptional performance in predicting academic outcomes. The findings suggest that this model is highly accurate, with an impressive (AUC) of 0.98. The model shows high sensitivity and specificity of 0.97 and 0.79, respectively. The model demonstrates its ability to accurately identify both true positives (satisfactory) and true negatives (unsatisfactory). This suggests that the model is not only accurate but also robust in its predictions.

The confusion matrix provided further insight into the model's performance. With a high number of true positives (13,479) and a relatively low number of false positives (425), it appears that the model is good at binary outcomes like satisfactory / un-satisfactory while minimizing misclassifications. This is particularly important in student outcome applications where accurate classification is crucial to the identification of students at risk.

We plan to build on these findings for further research into identify at risk students. Also, expanding the dataset to include many more features for analysis. The knowledge gained from this research will be used to develop new tools for Academic Advisors and Faculty to make early identification of students who are at risk. By intervening early, educators can provide timely support and guidance to at-risk students, increasing their chances of success and retention in the course. Educators and Advisors can implement targeted interventions such as academic advising, tutoring, and peer mentoring to address students' challenges and improve their chances of completing the course successfully. This also enables institutions to take proactive steps to help at risk students stay enrolled with potential need of financial assistance.

VI. Conclusions and Future Work

This project will be re-evaluated and improved in the future. In particular, feature size will be revisited for the Decision Tree and Random Forest Classifiers, because of the limitations of comparing five models. Decision Tree and Random Forest Models excel at interpretability and handling categorical features. Additionally, Random Forests are robust against overfitting and at handling high-dimensional datasets.

We will also re-evaluate the models removing the overall GPA and incorporating more features that may not score as high in feature importance. Sentiment analysis will be run on all comments by faculty to the student and coded for analysis by the Decision Tree and Random Forest classifiers.

VII. Acknowledgments

Research into related work created with the assistance of generative artificial intelligence [17].

Code templates for build the python machine learning models were initially derived from Kaggle.com [18].

Python Libraries Included: pandas [19] numpy [20] matplotlib [21] sklearn [22] Scipy [23] textblob [24]

Software Used: Anaconda Navigator - Python Environment Management [25] Python 3.12 - Programming Language Used for Analysis [26] R - Programming Language used by RMarkdown [27] R Studio - Building Markdown files for literate statistical programming [28] Microsoft Visual Studio Code - Python IDE [29]

Machine Learning Models: Decision Tree [5] Random Forest [7] Support Vector Machines [8] K-Nearest Neighbors [6] Naïve Bayes [9]

References

[1] S. Brown and M. Hossain, "Financial problems and academic performance in higher education institutions: A review of literature," *Higher Education Studies*, vol. 8, no. 2, pp. 45–58, 2018.

[2] Instructure, Inc., "Canvas lms," Learning Management System (LMS) by Instructure, Inc., Accessed 2024. [Online]. Available: https://www.instructure.com/canvas/

[3] Ellucian, "Banner erp," Enterprise Resource Planning (ERP) software by Ellucian, Accessed 2024. [Online]. Available: https://www.ellucian.com/solutions/banner-erp

[4] M. M. Tamada, R. Giusti, and J. F. d. M. Netto, "Predicting students at risk of dropout in technical course using lms logs," *Electronics*, vol. 11, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/3/468

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] T. Cover and P. Hart, "Nearest neighbor pattern classification," in *IEEE Transactions on Information Theory*, vol. 13, no. 1. IEEE, 1967, pp. 21–27.

[7] T. K. Ho, "Random decision forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. IEEE, 1995, pp. 278–282.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[9] J. D. Cook, "Naive bayes," *Encyclopedia of Machine Learning*, pp. 734–739, 2010.

[10] F. Dalipi, A. S. Imran, and Z. Kastrati, "Mooc dropout prediction using machine learning techniques: Review and research challenges," in *2018 IEEE global engineering education conference (EDUCON)*. IEEE, 2018, pp. 1007–1014.

[11] A. G. Costa, J. C. B. Mattos, T. T. Primo, C. Cechinel, and R. Muñoz, "Model for prediction of student dropout in a computer science course," in *2021 XVI Latin American Conference on Learning Technologies (LACLO)*, 2021, pp. 137–143.

[12] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, and V. Loumos, "Dropout prediction in e-learning courses through the combination of machine learning techniques," *Computers an Education*, vol. 53, no. 3, pp. 950–965, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131509001249

[13] L. Aulck, N. Velagapudi, J. Blumenstock, and J. West, "Predicting student dropout in higher education," 2017.

[14] B. Prenkaj, P. Velardi, G. Stilo, D. Distante, and S. Faralli, "A survey of machine learning approaches for student dropout prediction in online courses," *ACM Comput. Surv.*, vol. 53, no. 3, may 2020. [Online]. Available: https://doi.org/10.1145/3388792

[15] I. Sandoval-Palis, D. Naranjo, J. Vidal, and R. Gilar-Corbi, "Early dropout prediction model: A case study of university leveling course students," *Sustainability*, vol. 12, no. 22, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/22/9314

[16] F. Del Bonifro, M. Gabbrielli, G. Lisanti, and S. P. Zingaro, "Student dropout prediction," in *Artificial Intelligence in Education*, I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, and E. Millán, Eds. Cham: Springer International Publishing, 2020, pp. 129–140.

[17] OpenAI, "Openai's gpt-3 language model," Retrieved from https://openai.com/gpt-3, 2022.

[18] Kaggle. (Accessed 2024) Intro to machine learning. [Online]. Available: https://www.kaggle.com/learn/intro-to-machine-learning

[19] W. McKinney *et al.* (2022) pandas: powerful python data analysis toolkit. [Online]. Available: https://pandas.pydata.org/

[20] T. E. Oliphant *et al.* (2022) Numpy: fundamental package for scientific computing with python. [Online]. Available: https://numpy.org/

[21] J. D. Hunter *et al.* (2022) Matplotlib: plotting library for python. [Online]. Available: https://matplotlib.org/

[22] F. Pedregosa *et al.* (2022) scikit-learn: machine learning in python. [Online]. Available: https://scikit-learn.org/

[23] T. E. Oliphant *et al.* (2022) Scipy: scientific computing tools for python. [Online]. Available: https://www.scipy.org/

[24] S. Loria *et al.* (2022) Textblob: simplified text processing in python. [Online]. Available: https://textblob.readthedocs.io/

[25] I. Anaconda. (Accessed 2024) Anaconda navigator - python environment management. [Online]. Available: https://www.anaconda.com/products/individual

[26] P. online Foundation. (2024) Python 3.12 - programming language used for analysis. [Online]. Available: https://www.python.org/

[27] R. C. Team. (2024) R - programming language used by rmarkdown. [Online]. Available: https://www.r-project.org/

[28] P. RStudio. (Accessed 2024) R studio - building markdown files for literate statistical programming. [Online]. Available: https://rstudio.com/products/rstudio/

[29] M. Corporation. (Accessed 2024) Microsoft visual studio code - python ide. [Online]. Available: https://code.visualstudio.com/