ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #36590

Analyzing Student Procrastination to Identify At-Risk Behavior

Dr. Mihai Boicu, George Mason University

Mihai Boicu, Ph.D., is Associate Professor of Information Sciences and Technology at George Mason University, Associate Director of the Learning Agents Center (http://lac.gmu.edu), and Co-Director of Personalized Learning in Applied Information Technology Laboratory (http://plait.gmu.edu/).

Mr. Jay Lalwani, Thomas Jefferson High School for Science and Technology

Jay Lalwani is a student and aspiring Computer Scientist attending Thomas Jefferson High School for Science and Technology (Grade 12).

Aditya Daga, Edison Academy Magnet School

Aditya Daga is a Senior in high school at the Edison Academy Magnet School (Formerly Middlesex County Academy for Science Mathematics and Engineering Technologies) and is interested in data science, machine learning, and artificial intelligence. These interests cultivated after Aditya explored the intersection of statistics and computer science for his capstone project in his AP Statistics class. Aditya hopes to one day be a Data Scientist and leverage his skill sets to make informed business decisions using the vast amount of data available in today's world.

Analyzing Student Procrastination to Identify At-Risk Behavior

Mihai Boicu, Aditya Daga, Jay Lalwani
George Mason University

Abstract

Researchers have long considered procrastination as a form of self-destructive behavior and a key factor in university students' failure. However, procrastination may positively impact some students, so more factors must be considered to identify at-risk behavior. Early identification and intervention may mitigate the effects of procrastination and avoid failure. The goal of this study is to determine if a neural network model is appropriate to identify failure based on procrastination features and to determine if these procrastination features have a positive impact in increasing the model's basic metrics. Data is based on all sections offered in a term for a core database course containing 345 students and more than 6000 data points. Using a 10-fold cross validation, we obtained 10 neural networks with an average accuracy of 89% and a 86% recall. An ablation experiment for procrastination features shows a 47% decrease in recall.

Keywords

Procrastination, Early Failure Identification, Neural Networks, Undergraduate Education, Engineering Education

Introduction

In the educational context, procrastination is a known student behavior which was studied for a long time. Almost all students will exhibit procrastination behavior in various learning situations. According to Steel[1], 80%-95% of college students engage in procrastination, from which nearly 50% with some negative consequences. As we will present in the next section, many studies are looking at the effects of procrastination on student learning and grades. While most of the students are negatively affected by procrastination, there are cases in which it might have a positive effect. In this study, we investigated how to build a neural network model that will consider procrastination features and will be able to predict early in the course if the procrastination will have a negative impact. Moreover, we will analyze if the procrastination features impact the model metrics. Providing a customized way to build a procrastination model for a course is very useful because it allows the instructor to identify and intervene early to avoid long-term negative effects of procrastination.

We will address two main hypotheses. The first hypothesis is looking if a prediction model is possible:

> Given student data that can be extracted from a typical learning management system (LMS), it is possible to build a neural network model based on procrastination features that will predict if a student is unsuccessful in a course with a reasonable accuracy of 90%.

We selected 90% as the accuracy threshold because we expect some borderline errors, but we want to minimize such errors. For example, imagine a student who failed or passed with just 0.1% difference in the course grade. Such a student will easily be missed by any model. Moreover, 90% will miss categorize around 3 students in a class of 30 students, which is reasonable for early intervention.

The second hypothesis that we will address is that the procrastination features positively impact the model metrics, especially regarding accuracy and recall. The recall is important because it measures the false negatives, representing the students who will fail the course but were not identified.

> Given a neural network model using procrastination features, a new model constructed without the procrastination features will have a significant lower precision or recall.

While we address these hypotheses for a particular course, we designed the experiment to be easily adaptable for any course, allowing the proposed methods to be reused.

We start by describing current research studies on the effects of procrastination on student learning and course grade and how a procrastination model can be built to predict such effects. Then, we discuss what features can be extracted to assess procrastination based on a typical learning management system. We continue with a description of the course being studied and the collected data available. To prove that a model can be built, we perform a tenfold cross-validation experiment, building and evaluating ten different neural networks. The results were analyzed, and the average metrics obtained (around 90% accuracy and recall) demonstrate that the proposed model architecture will provide reasonable results independent of the training data. Next, we confirmed that the procrastination features have a determinant role in the developed models by performing an ablation experiment in which we built neural network models without the procrastination features and compared the metrics of the corresponding models. We conclude with a discussion of limitations and relevance of the obtained results and propose future steps. Also, information on how to obtain the code and its documentation is provided, allowing the experiment to be easily replicated on different course data.

Procrastination: Definition, Detection and Effects

In previous studies, there are various but similar definitions of procrastination. Lay defines the procrastination as "the tendency to postpone that which is necessary to reach some goal.[2,3]" Solomon et al.[4] define procrastination as "the act of needlessly delaying tasks to the point of experiencing subjective discomfort." Ellis and Knaus[5] consider that "it usually stems from and includes several emotional difficulties […] [you] choose needlessly and foolishly to harm yourself or them [others][5 (p. 14)]." Silver and Sabini[6] also regard it as an "irrational delay". We do not advocate for one definition or another but have an inclusive view trying to look at the procrastination features from various perspectives.

While our subjective observation saw an increase in procrastination attitude, we did not identify recent studies to assess the prevalence of procrastination in college students and how it evolved during COVID years. However, previous research indicates a prevalence of procrastination in the general population with chronic values for college students. Harriot and Ferrari[7], in a study with 211 adults, determined that 20% claimed to be chronic procrastinators. Ozer et al.[8], in a 2009 study among Turkish undergraduate students, found 52% of students self-reported procrastination.

A study to compare self-reported procrastination with automatic detection and their relationship with temporal discounting (the behavior to reduce the value of distant rewards) was performed by Howell et al.[9] with 95 introductory psychology students. They used computer-based assignments and examined the pattern of submissions to see if they accelerated before the due date. The performed statistical analysis determined that the rate of submission accelerated before the deadline. Also, they used a survey for the students in which they recorded students self-reported assignment procrastination, their intentions to perform the assignment, say-do correspondence, and their perceived academic control. They determined a correlation between these self-reported features and their detected submission procrastination. This study suggests that extracted features from a learning management system may be used instead of self-reported procrastination.

A similar study with our research, also using a neural network model, is "Mining Educational Data to Predict Students' Performance through Procrastination Behavior,[10]" in which researchers studied behavior patterns from students' assignment submission data to evaluate the extent of procrastination and classify them as non-procrastinators, procrastination candidates, and procrastinators to predict their class performance represented by their grade. After using both continuous and categorical data, the researchers found that using categorical data was a better predictor than using continuous data in the neural network. In addition, their neural network (hidden layer size = 100, activation function = 'relu,' adam optimization, and learning rate ε = 0.001) was reported to have a 96% accurate student performance prediction based on their level of procrastination. The critical difference in the researchers' data collection method was calculating spare time and idle time in assignment submissions in addition to the submission date to create more features about student procrastination behavior for their clustering method. A main difference with our proposal is the use of data over the entire course, while we propose to use only data collected in the second week to obtain an early warning.

In "Predicting Students' Academic Procrastination in Blended Learning Course Using Homework Submission Data,[11]" researchers presented an algorithm called "students' academic performance enhancement through homework late/non-submission detection (SAPE)," which first clustered students as procrastinators or non-procrastinators using k-means clustering (clustering data was collected only by checking if assignments were submitted on-time or late), then compared ten different classification methods (ZeroR, OneR, ID3, J48, random forest, decision stump, JRip, PART, NBTree, and Prism) to predict the performance of these clustered students best. They concluded that the Random Forest method consistently resulted in the greatest classification accuracy in predicting the performance of students, represented by its kappa statistic of 0.023.

Procrastination Features

An essential aspect of the study's design was maintaining the same workload for the students to ensure a good representation of their procrastination behavior (e.g., when an assignment was sent with respect to the due date of the assignment). However, if the workload increases, the procrastination behavior will likely change because of the additional work. Therefore, we used the following criteria to select the procrastination features:

(1) does not require extra work for the student (i.e., it is extracted as part of the normal student learning experience and does not require an additional survey or data entry by the student)

(2) can be relatively easily extracted from a learning management system (e.g., Blackboard)

(3) it is objective, factual and does not require additional interpretation (e.g., the submission date of an assignment, the grade obtained)

(4) reflects the working pattern of the student during the assignment lifetime (from the moment it is published until the moment is no longer accepted – with a due date somewhere in between).

One way to obtain procrastination information satisfying the above criteria is to design a short test assignment (i.e., 10 – 15 minutes in duration) that can be taken an unlimited number of times or have multiple submissions (each submission constitutes an attempt for the student). The questions must be randomly selected from a large pool, so it will not allow direct replication of results in a new attempt. Analyzing the attempts and their corresponding grades will create a learning pattern for the student without the need to inquire about their learning habits. Procrastination features that can be extracted include: the time of each attempt, the grade of each attempt, and the number of all attempts. However, because the number of attempts will vary from student to student, analyzing and evaluating individual attempt data may generate complex models that will require substantial training data (which is not readily available for a typical course). A solution is to use aggregated attempt information, like the date of the first attempt, the date of the last attempt, the number of attempts, the grade of the first attempt, the grade of the last attempt, the maximum grade obtained, the maximum grade obtained before the due date (or another expected date), and so on.

The limitation of the above approach is that students tend to perform the assignments sequentially; therefore, the data collected will reflect the work for this assignment rather than the entire module (week). A possible remediation is to include two or more small assignments in the module structure, for instance, one at the beginning and one at the end of the module. This way, the work pattern will likely be captured for the entire module.

Course Description

The course used in the experiment is an introductory database core course IT214 Database Fundamentals. The course has many sections per term, over 10 sections with over 300 students each term, providing a rich data source. The course is challenging, requiring both the conceptual design of database models and their Structured Query Language (SQL) implementation, and has a relatively high failure rate (DFW 20-30%). Early detection of failure and corresponding intervention will be critical in decreasing the failure rate.

The experiment was designed in the course's second module (second week). The structure of the module and the corresponding student tasks were:

(1) attend or view the course lecture either in-person, virtual, or recorded based on the course type

(2) study the learning materials provided (textbook readings, short videos, and short readings)

(3) take the first test (identified as A01) with unlimited attempts and containing 10 random questions in 10 minutes related to this module and previous module materials

(4) a video presentation of a hands-on design activity

(5) a step-by-step test guiding the student in the most important steps of similar hands-on activity; this is the second test used, having 10 pseudo-random questions generating many alternative scenarios (each time the test is taken a different scenario will be generated) (identified as A02)

(6) a different hands-on assignment that the students will perform and submit

(7) in the end, a short end-of-chapter quiz (20 random questions) that can be taken only 2 times (identified as A03).

The three tests A01, A02, and A03 will be used to record the working pattern of the students. While they will require around 40 minutes of work, they may take longer as more attempts are performed. The three tests were integrated into the learning process to reduce the workload and contribute to the module objectives, replacing or simplifying other activities. Test A01 is a formative test to check the memorization, understanding and application of the presented concepts. Test A02 is a formative test to check application and analysis of the presented concepts. Test A03 is a summative test for the previous objectives.

Data Collection and Normalized Input Files

The course learning management system (*i.e., Blackboard)* provides the following raw data related to the assignments: for each assignment, a list with all the attempts containing the student information, the attempt number per assignment, the date the attempt was started, the duration of the attempt, the grade obtained (if any) and other information that is not relevant to procrastination. Also, the learning management system may provide the entire grade book, which will contain the final grade in the course.

To allow our method to apply to various courses and course structures we created a generic input structure for the performed experiments. There are two input files: the attempts file and the grades file. The attempts file will contain the following information: student id (e.g., 1242), assignment name (e.g., "A01"), the attempt number (e.g., 2), the grade percent, from 0 to 100 (i.e., for a grade of 7 out of 10 points, the normalized grade is 70), the percent of remaining time to the due date, from -10000 to 100 where 100 will represent submission when the test was posted, 50 will represent submission at the middle of the working period, 0 submissions when the test was due, and negative values overdue submissions (if the course allows grace periods or overdue submissions). Sure, we do not expect a student to get 100% for this feature, but it is used to measure and normalize the timing of submissions. This will easily model early submissions, last-moment submissions and late submissions uniformly for all the assignments. The final grades file will contain each student's id and their final grade as a percentage from 0 to 100.

This normalization of data for the input files is easy to generate and assures the generality of the proposed methods that can be used for any other course for which such data can be extracted (i.e., the LMS must support data collection for attempts).

Neural Network Model

This section will briefly discuss the proposed neural network model and its rational.

Assignments. The model accepts one or more assignments as input. The number of assignments will be automatically computed based on the attempt data. Allowing one or more assignments allows easy customization of the model to various courses or various modules (i.e., one model after the second week and another model after the third week).

Features. The model will accept one or more of the following features for each assignment. Allowing the user to select what features are used in the model provides flexibility to run various experiments. This was used in the ablation experiment described later in the paper. The aggregated features that are considered in the model are:

(1) The *assignment start time*, representing the time when first attempt of an assignment is started. The percent value from the input file is normalized to a value between -1 and 1. Attempts less than -1, representing attempts that started way after the due date, are ignored. If no valid attempt (i.e., an attempt that was submitted and graded) is identified, a single default attempt will be considered with the starting time -1 and grade 0. This allows to process assignments that were not performed by the students.

(2) The *assignment end time*, representing the time when the last attempt of an assignment is started. The percent value from the input file is normalized to a value between -1 and 1. Attempts less than -1, representing attempts that started way after the due date, are ignored. If no valid attempt is identified, a single default attempt will be considered with the starting time -1 and grade 0.

(3) The *assignment start grade*, representing the grade obtained for the first attempt, normalized to a value between 0 and 1. If no attempt is found, a default value of 0 is considered.

(4) The *assignment end grade*, representing the grade obtained for the last attempt, normalized to a value between 0 and 1. If no attempt is found, a default value of 0 is considered. The *assignment max grade*, representing the best grade obtained for all valid attempts, normalized to a value between 0 and 1. If no attempt is found, a default value of 0 is considered.

(5) The *number of attempts*, representing the number of all valid attempts. The number of attempts is also normalized based on the maximum number of attempts in the data set.

Architecture. The proposed model is a classic multi-layer network having multiple inputs (the selected features for each assignment), several hidden layers, and one output layer. The inputs layer contains *numberOfAssignments x numberOfFeaturesPerAssignment* inputs. The architecture of the hidden layers was established during a previous experiment with random generated data[Error! Reference source not found.] and consists of six dense hidden layers with the following number of nodes: 256, 128, 64, 32, 16, 8. They use the *relu* activation function. The output layer has one node and uses the *sigmoid* activation function.

Implementation. The neural model is implemented in Python using Tensor Flow library[13]. The code for the model and the experiment is posted on the GitHub[14].

Example Label. We will use a binary label. If the final grade is under a given threshold (e.g., in this study we considered C+ or less), we will identify the student as being at risk because of procrastination and label the example with 1. Otherwise, the example is labeled 0.

Model Validation

A tenfold cross-validation experiment was performed to validate that the above model is appropriate to identify procrastination that leads to failure. The labeled data was randomly divided into ten equal groups of students. For each of the 10 experiments, one group was used as test data while the other nine were for training the neural model. The training data had around 310 students, and the test was computed for 35 students.

In this way, we showed that the model still performed well on the corresponding test data, independent of what training data was used. Figure 1 shows the box and whisker graph of the ten values obtained for accuracy, recall, and precision. There was a single outlier for recall at 0.56.

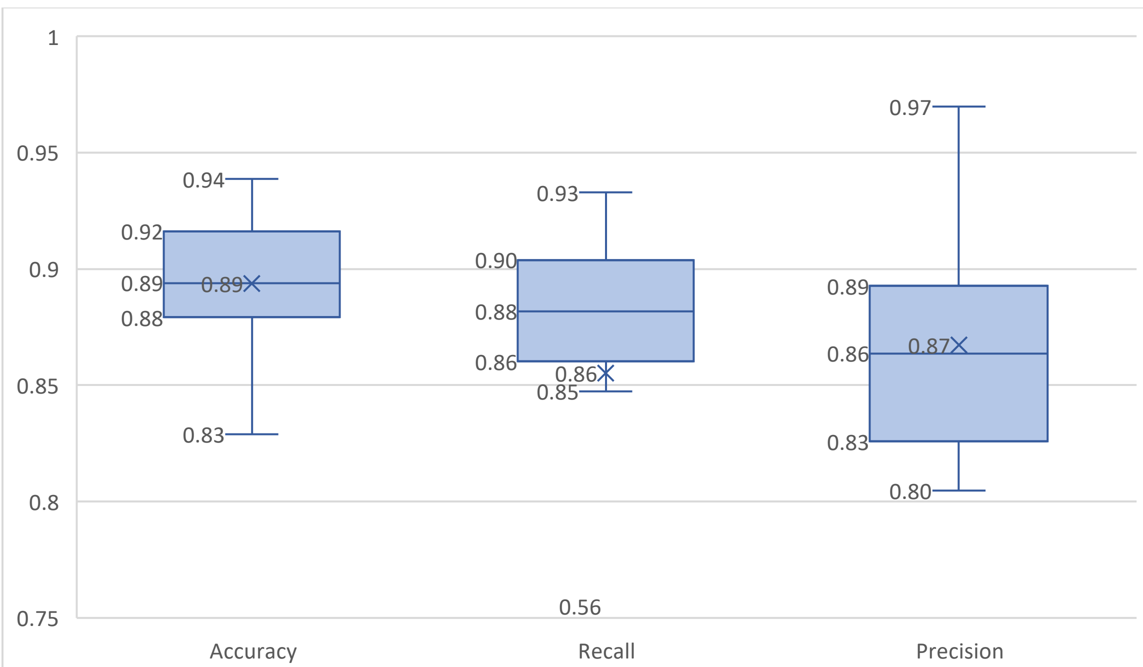

Figure 1 Box and Whisker Graph of Accuracy, Recall and Precision for 10 Neural Models

Relevance of Procrastination Features

To determine the impact of the procrastination features on the neural models, we performed an ablation experiment, repeating the experiment without using the procrastination features, and compared the results. In the repeated experiment, we kept the grading information for the assignments but no timing information. In Figure 2, we compare the two experiments' accuracy, recall, and precision.

The most significant difference is the recall value, which is much lower when procrastination features are not used. This shows the importance of procrastination features to avoid false negatives. From an educational perspective, the false negatives are very important because they reflect students who will underperform and were not identified. Another interesting result is that the precision of the models is similar, which means that if the models identify a student at risk, that student is indeed at risk. Combining the two results, we observe that if someone looks only at the grades at the end of week 2, they will be able to identify some of the failing students correctly, but not all. For instance, there might be students with good grades whose procrastination behavior will later affect them in the course. This is a compelling observation because it shows that the typical grade inspection done by an instructor will not be enough to identify the students at risk.

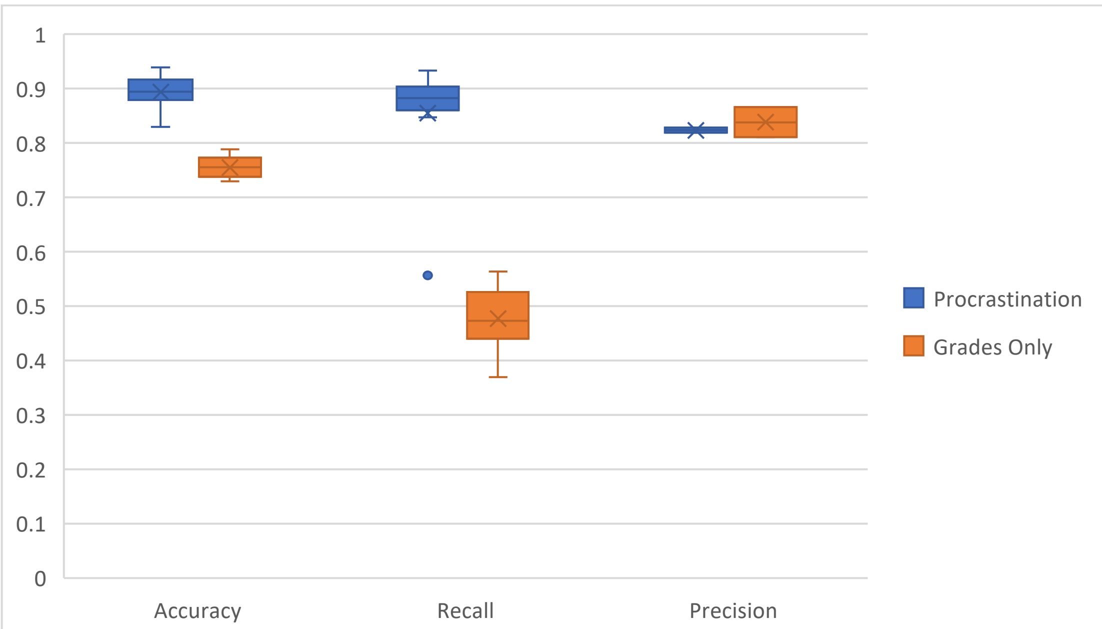

Figure 2 Accuracy, Recall and Precision Statistics Comparison in Ablation Experiment

Conclusion

There are several limitations related to our study. First, the study was performed with students from only one course in one term. To prove the model's generality, we plan to extend the study to several courses and multiple terms. The study investigated a single neural network model for procrastination. More types of neural network and other models must be developed and analyzed to identify the best model for a given context. The training and test data used in the study was relatively low with respect to the structure of the proposed neural network having the risk of model overfitting. To be sure that overfitting is avoided, a much larger dataset must be built.

Another limitation is that the computed features do not reflect the individual times of each attempt or the rate of attempts over time. Two extensions are possible: adding more features and computing the relevance of each feature through ablation experiments. Also, a limitation is that we considered the behavior at the end of week 2. We plan to create models looking at behavior later in the course, being able to monitor the evolution of procrastination behavior.

Despite these limitations, the performed experiment shows that a neural network model may be built to identify students whose performance will be affected by procrastination with around 89% accuracy and 89% recall. This confirms that procrastination behavior early in the course (in week 2) impacts the course results. Moreover, we determined that only grade analysis has a lower prediction power, having a very low recall. Therefore, adding procrastination features in any formal model that predicts student results in a course seems necessary.

We plan to extend this research in two complementary directions: build a better model to early predict student failure, and develop a system based on this model that can be easily used by instructors to predict and try to remediate student failure. Part of the proposed research is to study the benefits of this system on student learning.

Acknowledgements

This research was started as part of the 2022 Aspiring Scientist Summer Internship Program (https://science.gmu.edu/assip). We would like to express our deep gratitude to the organizers and especially to Dr. Andrea Karen Cobb for her dedication and leadership of the program.

References

1. Steel, Piers, The Nature of Procrastination: A Meta-Analytic and Theoretical Review of Quintessential Self-Regulatory Failure, in *Psychological Bulletin*, American Psychological Association, Washington, DC, vol. 133, pp. 65-94. DOI: 10.1037/0033-2909.133.1.65.
2. Lay, Clarry H., At Last my reseearch report on procrastination. Research Report No. 150, York University, Toronto, Canada.
3. Lay, Clarry H., At Last, My Research Article on Procrastination, Journal of Research in Personality 20, 1986, pg. 474-495.
4. Solomon, Laura J., Esther D. Rothblum, Academic Procrastination: Frequency and Cognitive-Behavioral Correlates, Journal of Counselling Psychology, American Psychological Association, 1984, vol. 31, no. 4, pg. 503-509.
5. Ellis, Albert, and Willliam J. Knaus, Overcoming procrastination, New York, Signet Books, 1977.
6. Silver, Mauri, and John Sabini, Procrastinating, Journal for the Theory of Social Behaviour, 1981, vol. 11, no. 2, pg. 207–221. https://doi.org/10.1111/j.1468-5914.1981.tb00033.x.
7. Harriott, Jesse, and Joseph R. Ferrari, Prevalence of procrastination among samples of adults, Psychological Reports, 1996, vol. 78, no. 2, pg. 611–616, https://doi.org/10.2466/pr0.1996.78.2.611.
8. Ozer, Bilge Uzun, Ayhan Demir, and Joseph R. Ferrari, Exploring academic procrastination among Turkish students: Possible gender differences in prevalence and reasons, Journal of Social Psychology, 2009, vol. 149, pg. 241-257.
9. Howell, Andrew J., David C. Watson, Russell A. Powell, Karen Buro, Academic procrastination: The pattern and correlates of behavioural postponement, in Personality and Individual Differences, volume 40, issue 8, 2006, pages 1519-1530, ISSN 0191-8869, https://doi.org/10.1016/j.paid.2005.11.023, available online at: https://www.sciencedirect.com/science/article/pii/S0191886906000055
10. Hooshyar, Danial, Margus Pedaste, Yeongwook Yang, Mining Educational Data to Predict Students' Performance through Procrastination Behavior in Entropy, 2020, 22, 12, doi:10.3390/e22010012
11. Akram, Aftab, Chengzhou Fu, Yuyao Li, Muhammad Yaqoob Javed, Ronghua Lin, Yuncheng Jiang, Yong Tang, Predicting Students' Academic Procrastination in Blended Learning Course Using Homework Submission Data in IEEE Access, 2019, Volume 7, doi: 10.1109/ACCESS.2019.2930867

12. Lalwani, Jay, Aditya Daga, Mihai Boicu, Using Machine Learning to Predict University Student Failure Based on Recurring Assignment Procrastination, in Mason Journal of Students' Scientist, 2022, 4, available online at: https://journals.gmu.edu/index.php/jssr/article/view/3487
13. TensorFlow, Library Overview, https://www.tensorflow.org/overview, accessed on 11/3/2022.
14. GitHub Procrastination Project, available online at: https://github.com/MihaiBoicu/procrastination

Mihai Boicu

Dr. Boicu is an Associate Professor in the Department of Information Sciences and Technology at George Mason University. He published over 120 peer-reviewed publications, including 4 books. He performs theoretical and applied research in Artificial Intelligence, Machine Learning, Probabilistic Reasoning and Engineering Education. He received more than 3M in funding from NSF, DARPA, IARPA, AFOSR, IC and other government agencies.

Aditya Daga

Mr. Daga is a senior at the Edison Academy Magnet School in Edison, NJ and is interested in pursuing statistics. He aspires to one day work as an actuary and apply his expertise in mathematics, statistics, and financial theory to assess risk and provide valuable insight to his enterprise. He also hopes to get his work published by the Casualty Actuarial Society. His interests outside of school span debate, civic engagement, and international affairs.

Jay Lalwani

Mr. Lalwani is a Senior at Thomas Jefferson High School for Science and Technology in Alexandria, VA. He has a strong background in computer science, with experience in machine learning, web development, and mobile app development. Working alongside teachers and professors to transform education with Artificial Intelligence, he has published in the Journal of Student-Scientists' Research. His passion for AI extends to further applications in Game Theory and Bioinformatics.