EUROPEAN JOURNAL OF EMERGING ARTIFICIAL INTELLIGENCE

Predictive Modeling of Programming Anxiety in University Students: A Logistic Regression Approach for Early Identification

Dr. Talia N. Marentis
Department of Educational Psychology Midlands Institute of Learning Sciences, Birmingham, UK

V0LUME02 ISSUE02 (2025)
Published Date: 24 December 2025 // Page no.: - 24-36

ABSTRACT

The proliferation of programming as a fundamental skill across academic and professional domains has amplified the need to address pedagogical challenges that hinder student success. Among these, programming anxiety—a specific form of situational anxiety characterized by fear, apprehension, and cognitive interference during programming tasks—has emerged as a significant barrier to learning and retention in computing education. This study addresses the critical need for scalable, automated methods to identify students at risk of experiencing high levels of programming anxiety. We developed and evaluated a machine learning classification model using a dataset of 1,732 undergraduate students from computing-related degree programs. The study was structured using the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework, encompassing comprehensive data preprocessing, feature engineering, and class imbalance mitigation through the Synthetic Minority Over-sampling Technique (SMOTE). Five supervised learning algorithms were systematically compared: Logistic Regression, Support Vector Machine, Naïve Bayes, Random Forest, and a Decision Tree classifier. Performance was assessed using a suite of metrics, including accuracy, precision, recall, F-measure, and Cohen's kappa, with model robustness confirmed via stratified 10-fold cross-validation. The Logistic Regression model demonstrated superior performance, achieving an accuracy of 97.75%, a precision of 96.88%, a recall of 96.70%, an F-measure of 96.77%, and a Cohen's kappa of 0.950. Key predictors identified by the model included previous academic performance in foundational programming courses, high school academic track, working student status, and sleep patterns. The resulting model provides a highly accurate and interpretable tool for educational institutions. Its potential for integration into learning management systems and academic advising platforms offers a proactive mechanism for delivering timely, targeted interventions, thereby fostering a more supportive learning environment and enhancing student success in the digital age.

Keywords: Programming Anxiety, Machine Learning, Educational Data Mining, Student Mental Health, Logistic Regression, Predictive Analytics, Higher Education

1. Introduction

1.1 Broad Background and Historical Context

In the landscape of 21st-century education and industry, computational thinking and programming proficiency have transitioned from niche specializations to foundational competencies essential across a vast spectrum of disciplines [1]. This paradigm shift is driven by the digital transformation of economies and the increasing demand for a workforce capable of innovating and problem-solving through technology. Consequently, higher education institutions worldwide have expanded their computing-related degree programs and integrated programming courses into curricula far beyond traditional computer science, including business, arts, and social sciences [3, 4]. This has led to a significant and sustained growth in student enrollment in these courses [6]. While this educational trend is critical for developing future-ready graduates, it has concurrently surfaced a significant pedagogical and psychological challenge: programming anxiety.

Programming anxiety is defined as a domain-specific anxiety characterized by feelings of tension, apprehension, and fear specifically related to the task of computer programming. It is distinct from more generalized forms of anxiety, such as math anxiety or test anxiety, though it may share some overlapping characteristics. Students experiencing programming anxiety often report a sense of cognitive overload, a fear of making errors, and intense self-doubt when faced with coding tasks. These psychological responses can manifest in detrimental behaviors, including procrastination, task avoidance, and disengagement from coursework [7]. The consequences of unchecked programming anxiety are severe, contributing to poor academic performance, elevated course withdrawal rates, and ultimately, higher attrition from computing-related

fields [9]. This phenomenon not only undermines individual student success and well-being but also poses a threat to the pipeline of skilled professionals needed to fuel technological innovation. As the demand for programming skills continues to surge [1, 5], understanding, identifying, and mitigating programming anxiety has become a critical priority for educators, administrators, and researchers. The challenge lies in moving beyond reactive measures, which often address student difficulties only after they have resulted in academic failure, toward proactive systems that can identify at-risk students early in their academic journey.

1.2 Critical Literature Review

The formal study of programming anxiety has led to the development of specialized psychometric instruments designed to measure its prevalence and severity. A notable example is the Programming Anxiety Scale, a validated tool that assesses dimensions such as peer-related stress and self-confidence in coding contexts [10]. While such scales are invaluable for research and targeted clinical assessment, their reliance on manual administration, scoring, and interpretation limits their scalability and utility for real-time, large-scale institutional monitoring. The logistical overhead makes it impractical to continuously screen entire student populations, creating a delay between the onset of anxiety and its detection.

In parallel, the field of educational data mining has seen a rapid expansion in the application of machine learning (ML) techniques to predict various student outcomes and identify at-risk individuals [8]. A significant body of research has focused on predicting general mental health conditions among students. For instance, studies have employed algorithms to assess and classify levels of depression, anxiety, and stress using data from standardized psychometric tools, with some models reporting exceptionally high accuracy [11, 14]. This research has extended to diverse populations beyond education, such as predicting anxiety among seafarers [12] and elderly patients [13], demonstrating the versatility of ML approaches. Furthermore, the unique pressures of global events like the COVID-19 pandemic spurred research into models that could predict anxiety based on context-specific factors, such as sociodemographic and exposure variables, highlighting the importance of situational context in predictive modeling [15, 16, 23].

Despite these advances, a critical review of this literature reveals several prevailing limitations. Many studies depend heavily on self-reported data, which can be prone to subjectivity and response bias, potentially limiting the generalizability of the findings [11, 28]. Moreover, methodological challenges such as class imbalance—where one class (e.g., low anxiety) is far more prevalent than another—can skew model performance if not properly addressed, an issue that remains underexplored in some research [27].

Within the specific context of higher education, researchers have begun to apply ML to predict student mental health [18] and academic stress [22]. These studies have successfully identified key predictors, including academic workload and social variables, and have demonstrated the effectiveness of algorithms like Support Vector Machine (SVM) and Logistic Regression [21]. For instance, one study in the Philippines used SVM to classify academic stress with 95% accuracy, identifying unrealistic expectations and heavy workloads as primary stressors during the pandemic [22]. Similarly, researchers have leveraged electronic health records and demographic data to build interpretable models of anxiety and depression in undergraduates, emphasizing the need for transparent feature selection [19]. This focus has even been extended to younger populations, identifying factors like family income and academic performance as significant predictors of anxiety in schoolchildren [20]. A systematic review confirmed the efficacy of SVM and Logistic Regression in this domain, also noting the widespread use of accuracy and precision as primary evaluation metrics [21]. However, a persistent theme across this body of work is its tendency to focus on generalized mental health conditions like depression or broad academic stress.

1.3 Research Gap

Despite the clear and documented negative impact of programming anxiety on student learning and retention [7, 9] and the parallel advancements in applying machine learning to educational and mental health data [18, 21], a significant research gap persists. There is a marked scarcity of studies dedicated to the development and validation of automated, data-driven models specifically designed for the early identification of students at risk of *programming anxiety*. The existing body of literature predominantly concentrates on either predicting broad academic outcomes like overall performance [8] or classifying general mental health issues such as anxiety and depression [11, 14, 19]. While this work is valuable, it fails to address the unique cognitive and emotional challenges inherent to learning computer programming. The specific antecedents and manifestations of programming anxiety are distinct from those of general academic stress, and models trained on generic mental health data may not be sensitive enough to detect this domain-specific construct. Therefore, there is a pressing need for research that develops and rigorously evaluates predictive models using features directly relevant to the student experience in computing education.

1.4 Objectives and Hypotheses

This study aims to bridge the identified research gap by focusing on the development of a specialized classification model for programming anxiety. The research is guided by a primary objective, two secondary objectives, and two core hypotheses.

Primary Objective: To develop and evaluate a robust machine learning classification model capable of accurately identifying university students experiencing high levels of programming anxiety based on a range of demographic, academic, and behavioral attributes.

Secondary Objectives:

1. To systematically compare the performance of five distinct supervised classification algorithms—Logistic Regression, Support Vector Machine, Naïve Bayes, Random Forest, and a Decision Tree classifier—to identify the most effective and reliable model for the classification task.

2. To identify and interpret the key predictor variables that significantly contribute to the classification of programming anxiety, thereby providing actionable insights for educators and academic advisors.

Hypotheses:

- **Hypothesis 1 (H1):** A supervised machine learning model, particularly a well-tuned Logistic Regression classifier, can predict student programming anxiety levels with a high degree of accuracy, precision, and recall (i.e., performance metrics exceeding 90%).

- **Hypothesis 2 (H2):** A multifaceted feature set, combining academic history variables (e.g., grades in introductory programming courses, previous semester GPA), learning-related behaviors (e.g., weekly study hours, nightly sleep hours), and key demographic and socioeconomic factors (e.g., working student status, access to technology), will contain significant predictors of programming anxiety.

By addressing these objectives, this research seeks to provide a practical, data-driven tool for educational institutions to proactively support student success in the increasingly critical field of computer programming.

2. Methods

2.1 Research Design

This study employed a quantitative, developmental research methodology [24] structured within a descriptive research framework [25]. The core objective was to construct and validate a predictive model, following a systematic process guided by the Cross-Industry Standard Process for Data Mining (CRISP-DM) [26]. The CRISP-DM framework provides a structured, multi-phase approach that ensures a rigorous and comprehensive development lifecycle, from initial problem conceptualization to final model deployment. The phases implemented in this study were: (1) Problem and Data Understanding, (2) Data Preprocessing, (3) Model Engineering, (4) Model Evaluation, and (5) Model Deployment. The overall approach was grounded in the supervised machine learning paradigm, wherein a labeled target variable (programming anxiety level) was used to train, validate, and test a series of classification algorithms. This design was chosen for its suitability in creating a predictive tool based on existing, historical data with a known outcome, with the ultimate goal of applying the resulting model to new, unseen data for early identification of at-risk students.

2.2 Participants / Sample

The data for this study consisted of a cross-sectional dataset obtained from a cohort of 1,732 undergraduate students. These participants were enrolled in various computing-related degree programs—including Bachelor of Science in Information Technology (BSIT), Bachelor of Science in Information Systems (BSIS), Bachelor of Science in Computer Science (BSCS), and Bachelor of Science in Entertainment and Multimedia Computing (BSEMC)—at a large public university in the Philippines. The data was collected during the first semester of the 2023–2024 academic year. The sample represented a diverse cross-section of students across different year levels, from first-year to fourth-year, providing a comprehensive view of the student experience in computing education. The institutional context is that of a large, urban university with a significant student population in its technology and computer studies departments, making it a suitable environment for studying the factors contributing to programming anxiety. All data was fully anonymized by the source department prior to its provision for this research to ensure the privacy and confidentiality of the student participants.

2.3 Materials and Apparatus

Data Source and Instrumentation:

The dataset used in this research was a secondary, anonymized administrative dataset provided by the

university's Computer Studies Department. It integrated data from multiple sources, including student academic records, enrollment information, and survey responses.

The primary dependent variable, **Programming Anxiety Level**, was a binary categorical variable ('High' vs. 'Low'). This classification was derived by the department from student scores on a validated psychometric instrument, the Programming Anxiety Scale [10]. This scale is designed to measure domain-specific anxiety related to coding, capturing dimensions like peer comparison stress and self-confidence in problem-solving and code comprehension. To protect student privacy and confidentiality, the raw numerical scores were not provided. Instead, the department aggregated the scores and released only the final classifications. In the dataset of 1,732 instances, 1,126 were labeled as 'High' programming anxiety and 606 were labeled as 'Low'.

The independent variables, or attributes, consisted of a wide range of demographic, socioeconomic, academic, and behavioral factors. A comprehensive list of these attributes is provided in the Markdown table below.

Attribute	Description	Values
Gender	Identifies the student's gender.	(Male/Female)
Age	Student's age in completed years.	Numerical
Working Student	Whether the student is employed while studying.	(Yes/No)
Parents with Higher Education	Whether at least one parent completed college education.	(Yes/No)
Family Monthly Income	Whether the family earns a certain threshold or more monthly.	(Yes/No)
Number of Siblings	Total number of the student's siblings.	Numerical
Course Enrolled	Degree program the student is currently enrolled in.	(BSIT, BSIS, BSCS, BSEMC)
Current Year Level	Student's year level in college.	Numerical (1-4)
Previous Semester GPA Remark	Academic remark based on previous semester's GPA.	(Above Average, Average, Below Average)
Final Comp. Prog. 1 Grade	Academic remark based on the final grade in Programming 1.	(Above Average, Average, Below Average)
Final Comp. Prog. 2 Grade	Academic remark based on the final grade in Programming 2.	(Above Average, Average, Below Average)
Senior High School Track	Whether the student took the ICT or STEM track in SHS.	(Yes/No)
Multiple ICT Equipment Access	Whether the student owns more than one ICT device.	(Yes/No)
Uses Mobile Data	Whether mobile data is the student's primary internet source.	(Yes/No)

Multi-modal Learner	Whether the student prefers a multi-modal learning style.	(Yes/No)
Study Hours per Week	Whether the student studies 10+ hours per week on average.	(Yes/No)
Sleep Hours per Night	Whether the student sleeps 6+ hours per night on average.	(Yes/No)
In a Relationship	Whether the student is currently in a romantic relationship.	(Yes/No)

Software and Hardware:

All data preprocessing, analysis, model engineering, and evaluation were conducted using the Python programming language (version 3.9) within a cloud-based computational environment, specifically Google Colaboratory. This platform provided access to necessary computing resources and facilitated efficient data handling and simulations. The following key Python libraries were utilized:

- **Pandas:** For data manipulation, cleaning, and exploratory data analysis.
- **NumPy:** For numerical operations and data structuring.
- **Scikit-learn:** For implementing machine learning algorithms, preprocessing steps (e.g., encoding, scaling), feature selection, and model evaluation metrics.
- **Matplotlib & Seaborn:** For data visualization to support exploratory analysis and presentation of results.
- **Imbalanced-learn:** Specifically for implementing the Synthetic Minority Over-sampling Technique (SMOTE).

2.4 Data Collection Procedure

The research followed the structured phases of the CRISP-DM model.

Phase 1: Problem and Data Understanding: This initial phase involved defining the research problem as a binary classification task: to predict whether a student exhibits 'High' or 'Low' programming anxiety. The anonymized dataset was acquired from the university's Computer Studies Department. A thorough Exploratory Data Analysis (EDA) was conducted to understand the fundamental characteristics of the data. This included examining the distribution of each variable, identifying the frequency of categorical values, checking for missing data, and calculating initial correlations between variables to gain preliminary insights into potential relationships. This phase confirmed the class imbalance, with nearly twice as many instances of 'High' anxiety as 'Low' anxiety, flagging this as a critical issue to address in the subsequent phase.

Phase 2: Data Pre-processing: This phase focused on cleaning and transforming the raw data into a format suitable for machine learning algorithms. Several critical steps were performed:

- **Handling Inconsistencies and Missing Values:** The dataset was meticulously checked for any inaccuracies or missing entries. Given the completeness of the administrative dataset, no significant missing values were found that required imputation.
- **Encoding Categorical Variables:** All non-numeric features were converted into a machine-readable format. Nominal variables with two categories (e.g., 'Gender', 'Working Student') were label-encoded into 0 and 1. Nominal variables with more than two categories ('Course Enrolled') and ordinal variables ('Previous Semester GPA Remark') were transformed using One-Hot Encoding. This process creates new binary columns for each category, preventing the model from assuming a false ordinal relationship between categories.
- **Addressing Class Imbalance:** To counteract the potential for the model to be biased towards the majority class ('High' anxiety), the **Synthetic Minority Over-sampling Technique (SMOTE)** was applied to the training data. SMOTE works by creating synthetic samples of the minority class ('Low' anxiety) by interpolating between existing minority class

instances in the feature space. This results in a balanced dataset for model training, which is crucial for improving the model's ability to correctly identify instances of the minority class, a key requirement for an effective early warning system [27]. The dataset was first split into training and testing sets, and SMOTE was applied *only* to the training set to prevent data leakage.

- **Feature Selection:** To enhance model performance, reduce overfitting, and improve interpretability, a systematic feature selection process was conducted. Two techniques were employed: (1) **Random Forest Feature Importances**, which ranks features based on their contribution to the model's decision-making process, and (2) **Sequential Feature Selection (SFS)**, an iterative method that selects a subset of features that maximizes model performance. The intersection of features identified as important by both methods was chosen for the final model. This process resulted in the selection of nine significant attributes: Working Status, Course, Current Year Level, Previous Semester GPA, Computer Programming 1 Final Grade, Senior High School Track, Multiple ICT Equipment Access, Multi-modal Learner status, and Sleep Hours per Night.

2.5 Data Analysis

Model Training and Selection:

Five distinct supervised classification algorithms were selected for evaluation, chosen based on their proven effectiveness in similar predictive tasks in educational and mental health research [21]. The algorithms were:

1. **Logistic Regression:** A linear model valued for its interpretability, computational efficiency, and strong performance in binary classification tasks.
2. **Support Vector Machine (SVM):** A powerful algorithm that finds an optimal hyperplane to separate classes, effective in high-dimensional spaces.
3. **Naïve Bayes (NB):** A probabilistic classifier based on Bayes' theorem with a "naïve" assumption of feature independence. It is known for its simplicity and speed.
4. **Random Forest (RF):** An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification, known for its robustness and ability to handle complex interactions.
5. **Decision Tree (J48/CART):** A non-parametric model that creates a tree-like structure of decisions, valued for its transparency and ease of interpretation.

Model Validation:

To ensure the reliability and generalizability of the model's performance, a Stratified 10-Fold Cross-Validation technique was employed. The dataset was partitioned into 10 equal-sized folds, maintaining the original percentage of samples for each class in each fold. In each of the 10 iterations, one fold was held out as the validation set, while the other nine folds were used for training. The performance metrics were then averaged across all 10 iterations. This rigorous method provides a more stable and less biased estimate of the model's performance on unseen data compared to a simple train-test split, significantly reducing the risk of overfitting.

Performance Metrics:

A comprehensive suite of metrics was used to conduct a multi-faceted evaluation of each model's performance.

- **Classification Metrics:**
  - **Accuracy:** The proportion of total predictions that were correct.
  - **Precision:** The proportion of positive identifications that were actually correct (minimizes false positives).
  - **Recall (Sensitivity):** The proportion of actual positives that were correctly identified (minimizes false negatives). This is particularly important for an early warning system.
  - **F-Measure (F1-Score):** The harmonic mean of precision and recall, providing a single score that balances both concerns.
  - **Cohen's Kappa:** A statistic that measures inter-rater agreement for categorical items, indicating how much better the classifier performs compared to a random chance classifier.
- **Error Metrics:**
  - **Mean Absolute Error (MAE):** The average of the absolute differences between predicted and actual values.
  - **Root Mean Squared Error (RMSE):** The square root of the average of squared differences, which penalizes larger errors more heavily.

- **Relative Absolute Error (RAE) & Root Relative Squared Error (RRSE):** Normalized error metrics that compare the model's error to that of a simple predictor (e.g., the mean), providing a standardized measure of performance.

- **Goodness-of-Fit (for Logistic Regression):** The performance of the final Logistic Regression model was further assessed using the **Receiver Operating Characteristic (ROC) curve** and the **Area Under the Curve (AUC)**. The AUC represents the model's ability to discriminate between the 'High' and 'Low' anxiety classes across all possible classification thresholds, with a value of 1.0 indicating a perfect classifier.

3. Results

3.1 Preliminary Analyses

Following the data pre-processing phase, the final dataset used for model engineering consisted of 1,732 instances and the nine significant features identified through the selection process. The original class distribution in the dataset was imbalanced, with 1,126 instances (65.0%) labeled as 'High' programming anxiety and 606 instances (35.0%) labeled as 'Low'. After splitting the data into an 80/20 train/test partition, the SMOTE procedure was applied to the training set. This successfully balanced the class distribution in the training data, creating an equal number of instances for both 'High' and 'Low' anxiety classes, thereby mitigating the risk of model bias towards the majority class. The held-out test set retained its original imbalanced distribution to provide a realistic evaluation of the model's performance on real-world data.

3.2 Main Findings

Model Performance Comparison:

The five supervised learning algorithms were trained and evaluated using stratified 10-fold cross-validation. The average performance across the folds for each classification metric is presented in Table 1. The results clearly indicate that the Logistic Regression model outperformed all other competing algorithms across every key metric. It achieved the highest F-measure (96.77%), accuracy (97.75%), precision (96.88%), recall (96.70%), and Cohen's Kappa (0.950). The Support Vector Machine (SVM) model demonstrated the second-best performance, with an accuracy of 97.69% and a Cohen's Kappa of 0.949, indicating it was also a highly effective classifier. The Naïve Bayes, Random Forest, and Decision Tree models, while still performing reasonably well, showed a distinct drop in performance, with accuracies clustering between 94.57% and 94.98% and Cohen's Kappa values ranging from 0.879 to 0.889. The superior F-measure of the Logistic Regression model is particularly noteworthy as it reflects a strong balance between precision and recall, which is crucial for an effective student-at-risk detection system.

Table 1. Performance Comparison of Classification Algorithms

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Cohen's Kappa
Logistic Regression	**97.75**	**96.88**	**96.70**	**96.77**	**0.950**
Support Vector Machine	97.69	96.42	97.03	96.72	0.949
J48 Decision Tree	94.98	93.97	91.59	92.72	0.889
Random Forest	94.57	92.93	91.58	92.17	0.880
Naive Bayes	94.57	94.29	90.11	92.03	0.879

Error Analysis Comparison:

To further assess the models' predictive stability and deviation, four error metrics were calculated. The results, summarized in Table 2, reinforce the findings from the classification metrics. The Logistic Regression model consistently exhibited the lowest error rates, with a Mean Absolute Error (MAE) of 0.0225, a Root Mean Squared Error

(RMSE) of 0.1464, a Relative Absolute Error (RAE) of 0.03%, and a Root Relative Squared Error (RRSE) of 30.71%. These minimal error values indicate that the model's predictions had the smallest average deviation from the actual outcomes and were the most consistent. The SVM model again showed strong performance with very low error rates, closely trailing Logistic Regression. In contrast, the Decision Tree, Random Forest, and Naïve Bayes models all produced higher error values, particularly for RMSE and RRSE, suggesting greater variance and less stability in their predictions.

Table 2. Error Analysis of Classification Algorithms

Algorithm	MAE	RMSE	RAE (%)	RRSE (%)
Logistic Regression	**0.0225**	**0.1464**	**0.03**	**30.71**
Support Vector Machine	0.0227	0.1477	0.10	31.05
J48 Decision Tree	0.0501	0.2238	5.08	46.91
Random Forest	0.0541	0.2325	5.49	48.87
Naive Bayes	0.0541	0.2327	5.50	48.92

In-depth Analysis of the Best Model (Logistic Regression):

Given its superior performance, the Logistic Regression model was selected for further analysis. The stability of the model was confirmed through the 10-fold cross-validation process, where both accuracy and F-measure consistently remained between 97% and 99% across all ten folds, indicating high reliability when tested on different subsets of the data.

The model's excellent discriminatory power was quantified by the Area Under the Receiver Operating Characteristic Curve (AUC). The Logistic Regression model achieved an **AUC of 0.98**, with a 95% confidence interval of (0.960, 0.995) and a p-value < 0.001. A value so close to 1.0 signifies that the model is outstanding at distinguishing between students with 'High' and 'Low' programming anxiety.

Finally, an analysis of the model's coefficients provided insight into the factors driving its predictions. The coefficients reveal the relative importance and the direction of influence of each feature on the probability of a student being classified with high anxiety.

Table 3. Logistic Regression Model Coefficients for Selected Features

Feature	Coefficient	Interpretation
Previous Semester GPA (Below Average)	3.020	Strong positive association with high anxiety.
Working Student (Yes)	2.527	Strong positive association with high anxiety.
Computer Prog. 1 Grade (Below Average)	4.124	Strongest positive association with high anxiety.

Current Year Level (e.g., Year 2)	2.888	Positive association; anxiety increases after year 1.
Course (BSCS)	2.113	Positive association relative to the reference course.
Senior High School Track (Non-ICT)	2.084	Positive association with high anxiety.
Sleep Hours per Night (No, <6 hours)	-2.768	Strong negative association (more sleep, less anxiety).
Multi-modal Learner (Yes)	-2.631	Negative association (preference for multiple learning styles is linked to lower anxiety).
Multiple ICT Equipment Access (No)	-2.126	Negative association (access to more devices is linked to lower anxiety).

The coefficients reveal that academic struggles, particularly a "Below Average" grade in the foundational "Computer Programming 1" course, is the most potent predictor of high anxiety. Being a working student and having a low GPA in the previous semester are also strong contributors. Conversely, factors like getting adequate sleep, having a preference for multi-modal learning, and having access to multiple ICT devices were associated with a lower likelihood of experiencing high programming anxiety.

3.3 Exploratory Findings

An exploratory interaction analysis suggested that the negative impact of certain stressors may be compounded. For instance, the positive coefficient for 'Working Student (Yes)' was observed to be significantly larger for students who also had a 'Below Average' grade in 'Computer Programming 1'. This suggests that the pressures of employment and academic difficulty interact, creating a synergistic effect that dramatically increases the likelihood of programming anxiety. This finding, while not a primary objective, underscores the complex, multifactorial nature of the student experience and highlights the importance of considering how different life and academic factors intersect when designing support systems.

4. Discussion

4.1 Interpretation

The primary outcome of this research is the successful development and validation of a highly accurate machine learning model for the automated classification of student programming anxiety. The Logistic Regression model, achieving 97.75% accuracy and an F-measure of 96.77%, demonstrates that it is possible to reliably identify students at risk using readily available institutional data. This finding provides strong support for our first hypothesis (H1). The practical implication is profound: educational institutions are no longer limited to reactive, post-hoc interventions. Instead, they can implement a proactive, data-driven system to flag students who are likely struggling with this specific psychological barrier, enabling support services to intervene *before* anxiety leads to course failure or withdrawal.

The interpretation of the model's coefficients provides deeper, actionable insights and supports our second hypothesis (H2). The most significant predictor of high programming anxiety was a "Below Average" grade in the introductory "Computer Programming 1" course. This is intuitively logical; early struggles in a foundational subject can shatter a student's confidence and create a cycle of fear and avoidance in subsequent courses. This finding pinpoints a critical intervention window: the end of the first programming course. Similarly, factors like being a working student and having a poor GPA from the previous semester point to the powerful influence of external pressures and cumulative academic difficulty. These students may be juggling competing priorities and have less time and cognitive resources to dedicate to a demanding subject, thereby increasing their anxiety.

Conversely, the model identified several protective factors. Students who reported getting adequate sleep (6+ hours per night) and those who identified as multi-modal learners were less likely to be classified with high anxiety. This suggests that student well-being (sleep) and learning preferences are not trivial factors. An institution that promotes healthy habits and employs varied pedagogical strategies that cater to different learning styles may inadvertently be building resilience against programming anxiety. The negative coefficient associated with not having access to multiple ICT devices further suggests that resource availability plays a role, possibly by reducing technical friction and frustration in the learning process.

4.2 Comparison with Literature

The performance of our Logistic Regression model compares favorably with the existing body of work on mental health prediction using machine learning. The achieved accuracy of 97.75% is notably high, surpassing the accuracy levels reported in several studies focused on predicting general anxiety or stress in various populations, which often fall in the 68-88% range [12, 18]. Our results are more aligned with studies that reported exceptionally high performance, although some of those may have faced different methodological constraints [11]. The robust performance of our model can be attributed to several factors articulated in the literature: a well-curated feature set, a large sample size, and the critical step of addressing class imbalance using SMOTE, a technique known to improve model performance on imbalanced datasets [27].

The specific predictors identified in our model resonate strongly with findings from previous educational and psychological research. The paramount importance of early academic performance as a predictor of subsequent outcomes is a well-established principle in educational data mining [8]. Our finding that poor performance in an initial programming course is the top predictor of anxiety aligns directly with studies that have identified academic workload and unrealistic expectations as key stressors for computing students [7, 22]. Furthermore, the significance of socioeconomic factors like 'Working Student' status echoes research that highlights the impact of social and demographic variables on student mental health [19, 23].

Our study's primary contribution, however, lies in its specific focus. While a systematic review by Vergaray et al. [21] confirmed the effectiveness of algorithms like SVM and Logistic Regression for predicting general stress in college students, our work applies these methods to the niche but critical construct of programming anxiety. By doing so, we address a specific gap where students may not screen positive for generalized anxiety but still suffer from a debilitating, domain-specific variant. The structured methodology, following the CRISP-DM framework, aligns our work with best practices for applied data mining projects as reviewed by Schröer et al. [26], adding to the methodological rigor of our findings.

4.3 Strengths and Limitations

This study possesses several key strengths that bolster the confidence in its findings. First, its high ecological validity, derived from the use of a large, real-world administrative dataset of 1,732 students, ensures the model is grounded in the authentic student experience. Second, the adoption of a rigorous, structured methodology based on the CRISP-DM framework [26], combined with robust validation using stratified 10-fold cross-validation, minimizes the risks of bias and overfitting and enhances the reliability of the performance estimates. Third, the study successfully developed a highly accurate (97.75%) yet highly interpretable model. The choice of Logistic Regression as the final model provides clear, quantifiable insights into the factors driving the predictions, making the results directly actionable for non-technical stakeholders like educators and advisors. Fourth, by proactively addressing the common methodological pitfall of class imbalance through SMOTE [27], the model's ability to correctly identify the minority class (students with 'Low' anxiety) was strengthened, leading to a more balanced and trustworthy classifier. Finally, the novel focus on programming anxiety, as opposed to general mental health, represents a significant contribution to the field of educational data mining and computing education research.

Despite these strengths, it is crucial to acknowledge the study's limitations. First, the data was sourced from a single public university in the Philippines. While this provides a deep and consistent dataset, it may limit the generalizability of the specific model coefficients to other institutions with different student demographics, curricula, or cultural contexts. The model would likely need to be recalibrated or retrained on local data before deployment elsewhere. Second, the dependent variable was a pre-categorized binary label ('High'/'Low'). This simplification, while necessary for privacy, results in a loss of granularity compared to the original continuous anxiety scores, preventing analysis of moderate anxiety levels. Third, the cross-sectional nature of the data provides a snapshot of anxiety at a single point in time. It cannot establish causality or track the trajectory of a student's anxiety over the course of their degree program. A longitudinal study would be required to understand how these factors evolve. Fourth, the model relies on administrative and self-reported data, which may be subject to inherent inaccuracies or biases. Lastly, while our study evaluated a range of effective traditional algorithms, it did not explore more complex, non-linear models such as deep learning neural networks, which have shown promise in other areas of anxiety prediction [29, 30].

4.4 Implications

The findings of this study have significant practical and theoretical implications.

Practical Implications: The most direct implication is the potential for the developed model to be operationalized as an early warning system within higher education institutions. Integrated into a university's Learning Management System (LMS) or student information system, the model could automatically analyze student data at the end of each semester and generate a risk score for programming anxiety. This would enable academic advisors, faculty members, and student support services to move from a reactive to a proactive support paradigm. Instead of waiting for a student to fail, they could receive an alert prompting them to reach out with targeted interventions. Such interventions could include offers of specialized tutoring, enrollment in peer-led study groups, academic coaching on time management and study skills, or confidential referrals to counseling services. This data-informed approach allows for the efficient allocation of limited support resources to the students who need them most.

Theoretical Implications: This research contributes to the theoretical understanding of programming anxiety by empirically identifying and ranking its key predictors. It validates the construct as a predictable phenomenon, distinct from general anxiety, with a clear set of academic and behavioral antecedents. The finding that early academic struggles are the most potent predictor reinforces theories of self-efficacy, where initial failures can create a negative feedback loop that erodes a student's belief in their ability to succeed. Furthermore, the study demonstrates the power of applying data science and machine learning methodologies to complex pedagogical problems, providing a methodological framework for future research in educational psychology and learning analytics.

4.5 Conclusion and Future Directions

In conclusion, this study successfully demonstrated that a Logistic Regression model, developed using a structured data mining process and trained on real-world institutional data, can serve as a highly effective and automated tool for identifying university students at risk of programming anxiety. The model achieved outstanding performance, and the analysis of its features provides clear, actionable insights for educators. This work represents a critical step towards creating more supportive, data-informed learning environments that can mitigate the negative impacts of anxiety and improve student well-being and academic success in computing disciplines.

To build upon this research, several future directions are recommended. First, it is essential to **validate the model's performance and generalizability** by testing it on diverse datasets from other universities, including those in different countries and with different institutional characteristics. Second, future work should prioritize **longitudinal studies** that track student anxiety and its predictors over the entire duration of their academic program. This would provide invaluable insights into the developmental trajectory of programming anxiety and identify the most critical intervention points. Third, researchers should **explore the application of more advanced modeling techniques**, including deep learning approaches [29, 30], which may be able to capture more complex, non-linear relationships within the data and potentially improve predictive accuracy further. Fourth, the ultimate goal of this work is to improve student outcomes, which necessitates research that moves **from prediction to intervention**. Future studies should focus on developing and rigorously evaluating the effectiveness of targeted support programs designed for students identified as at-risk by the model. Finally, the feature set could be expanded to include more granular and real-time data, such as analysis of students' coding process data (e.g., number of compilation errors, time between keystrokes) or even psycho-physiological data from wearable sensors, which could provide even earlier and more sensitive indicators of anxiety.

References

[1] R. Yazdanian, R. L. Davis, X. Guo, F. Lim, P. Dillenbourg, and M.-Y. Kan, 'On the radar: Predicting nearfuture surges in skills' hiring demand to provide early warning to educators', Computers and Education: Artificial Intelligence, vol. 3, p. 100043, Jan. 2022. doi.org/10.1016/j.caeai.2021.100043

[2] K. M. N. Rebuta, I. M. P. Cabaron, R. J. C. Pucong, J. M. C. Bisquera, R. T. Llerado, and M. V. M. Buladaco, 'Relationship of programming skills and perceived value of learning programming among Information Technology education students in Davao Del Sur', Int. J. Res. Innov. Soc. Sci., vol. 6, no. 6, pp. 882–887, 2022. doi.org/10.47772/ijriss.2022.6633

[3] Philippine Commission on Higher Education, CMO No. 25, Series of 2015: Policies, standards, and guidelines for the Bachelor of Science in Information Technology program, 2015. [Online]. Available: https://ched.gov.ph/wp-content/uploads/2017/10/CMO-no.-25-s.-2015.pdf

[4] Commission on Higher Education, "CHED Memorandum Order No. 24, series of 2015: Policies, Standards and Guidelines for the Bachelor of Library and Information Science (BLIS) Program," Oct. 12, 2015. [Online]. Available: https://ched.gov.ph/wp-content/uploads/2017/10/CMO-no.-24-s.-2015.pdf

[5] L. Hu, 'Programming and 21st century skill development in K-12 schools: A multidimensional meta-analysis', Journal of Computer Assisted Learning, vol. 40, no. 2, pp. 610–636, Nov. 2023. doi.org/10.1111/jcal.12904

[6] J. Zheng, M. Duffy, and G. Zhu, 'Predictors of university students' intentions to enroll in computer programming courses: a mixed-method investigation', Discover Education, vol. 3, no. 1, Sep. 2024. doi.org/10.1007/s44217-024-00232-5

[7] C. N. P. Olipas, R. F. Leona, A. C. A. Villegas, A. I. Cunanan Jr., and C. L. P. Javate, 'The academic performance and the computer programming anxiety of BSIT students: A basis for instructional strategy improvement', Int. J. Adv. Eng. Manag. Sci., vol. 7, no. 6, pp. 125–129, 2021. dx.doi.org/10.22161/ijaems.76.15

[8] Ahmed, 'Student performance prediction using machine learning algorithms', Applied Computational Intelligence and Soft Computing, 2024. doi.org/10.1155/2024/4067721

[9] C. Connolly, E. Murphy, and S. Moore, 'Programming Anxiety Amongst Computing Students—A key in the retention debate? ', IEEE Transactions on Education, vol. 52, no. 1, pp. 52–56, Aug. 2008. doi.org/10.1109/te.2008.917193

[10] Yildirim, O. G., & Özdener, N. (2022), 'Development and validation of the Programming Anxiety Scale', International Journal of Computer Science Education in Schools, 5(3), 17–34, 2022. doi.org/10.21585/ijcses.v5i3.140

[11] P. Kumar, S. Garg, and A. Garg, 'Assessment of anxiety, depression and stress using machine learning models', Procedia Computer Science, vol. 171, pp. 1989–1998, 2020. doi.org/10.1016/j.procs.2020.04.213

[12] A. Sau and I. Bhakta, 'Screening of anxiety and depression among seafarers using machine learning technology', Informatics in Medicine Unlocked, vol. 16, p. 100228, 2019. doi.org/10.1016/j.imu.2019.100228

[13] A. Sau and I. Bhakta, 'Predicting anxiety and depression in elderly patients using machine learning technology', Healthcare Technology Letters, vol. 4, no. 6, pp. 238–243, Nov. 2017. doi.org/10.1049/htl.2016.0096

[14] A. Priya, S. Garg, and N. P. Tigga, 'Predicting anxiety, depression and stress in modern life using machine learning algorithms', Procedia Computer Science, vol. 167, pp. 1258–1267, 2020. doi.org/10.1016/j.procs.2020.03.442

[15] J. D. Elhai, H. Yang, D. McKay, G. J. G. Asmundson, and C. Montag, 'Modeling anxiety and fear of COVID-19 using machine learning in a sample of Chinese adults: associations with psychopathology, sociodemographic, and exposure variables', Anxiety, Stress, & Coping, vol. 34, no. 2, pp. 130–144, 2021. doi.org/10.1080/10615806.2021.1878158

[16] Albagmi, F. M., Alansari, A., Shawan, D. S. A., AlNujaidi, H., & Olatunji, S. O., 'Prediction of generalized anxiety levels during the Covid-19 pandemic: A machine learning-based modeling approach', Informatics in Medicine Unlocked, 28, 100854, 2022. doi.org/10.1016/j.imu.2022.100854

[17] S. A. Farooq, O. Konda, A. Kunwar, and N. Rajeev, 'Anxiety prediction and analysis - A machine learning based approach', 4th International Conference for Emerging Technology (INCET), 2023. doi.org/10.1109/incet57972.2023.10170115

[18] S. Mutalib, 'Mental health prediction models using machine learning in higher education institutions', Turkish Journal of Computer and Mathematics Education, vol. 12, no. 5, pp. 1782–1792, 2021. doi.org/10.17762/turcomat.v12i5.2181

[19] M. D. Nemesure, M. V. Heinz, R. Huang, and N. C. Jacobson, 'Predictive modeling of depression and anxiety using electronic health records and a novel machine learning approach with artificial intelligence', Scientific Reports, vol. 11, no. 1, 2021. doi.org/10.1038/s41598-021-81368-4

[20] R. Qasrawi, S. VicunaPolo, D. A. Al-Halawa, S. Hallaq, and Z. Abdeen, 'Assessment and prediction of depression and anxiety risk factors in schoolchildren: Machine learning techniques performance analysis', JMIR Formative Research, vol. 6, no. 8, e32736, 2022. doi.org/10.2196/32736

[21] A. D. Vergaray, N. S. Ríos, J. I. Necochea-Chamorro, K. Z. Ramos, and Y. Del Rosario Vásquez Valencia, 'Systematic review of machine learning techniques to predict anxiety and stress in college students', Informatics in Medicine Unlocked, vol. 43, p. 101391, 2023. doi.org/10.1016/j.imu.2023.101391

[22] Geronimo, S. M., Hernandez, A. A., Abisado, M. B., Rodriguez, R. L., Nova, A. C., Caluya, S. S., & Blancaflor, E. B., 'Understanding Perceived Academic Stress among Filipino Students during COVID19 using Machine Learning', SIGITE '23: Proceedings of the 24th Annual Conference on Information Technology Education, 4, 54–59, 2023. doi.org/10.1145/3585059.3611412

[23] N. B. Mendoza, R. B. King, and J. Y. Haw, 'The mental health and well-being of students and teachers during the

COVID-19 pandemic: combining classical statistics and machine learning approaches', Educational Psychology, vol. 43, no. 5, pp. 430–451, 2023. doi.org/10.1080/01443410.2023.2226846

[24] Ibrahim, A., 'Definition, purpose, and procedure of developmental research: An analytical review', Asian Research Journal of Arts & Social Sciences, 1(6), 1–6, 2016. doi.org/10.9734/arjass/2016/30478

[25] M. Vale, 'Descriptive research design and its myriad uses', Elsevier Author Services - Articles, Dec. 27, 2023. [Online]. Available: https://scientific-publishing.webshop.elsevier.com/researchprocess/descriptive-research-design-and-its-myriad-uses/. [Accessed: Apr. 12, 2025].

[26] C. Schröer, F. Kruse, and J. C. M. Gómez, 'A systematic literature review on applying CRISP-DM process model', Procedia Computer Science, vol. 181, pp. 526–534, Jan. 2021. doi.org/10.1016/j.procs.2021.01.199

[27] Alija, S., Beqiri, E., Gaafar, A. S., & Hamoud, A. K., 'Predicting students' performance using supervised machine learning based on imbalanced dataset and wrapper feature selection', Informatica, 47(1), 2023. doi.org/10.31449/inf.v47i1.4519

[28] W. L. Ku and H. Min, 'Evaluating machine learning stability in predicting depression and anxiety amidst subjective response errors', Healthcare, vol. 12, no. 6, p. 625, 2024. doi.org/10.3390/healthcare12060625

[29] Bendebane, L., Laboudi, Z., Saighi, A., Al-Tarawneh, H., Ouannas, A., & Grassi, G., 'A Multi-Class Deep Learning Approach for Early Detection of Depressive and Anxiety Disorders Using Twitter Data', Algorithms, 16(12), 543, 2023. doi.org/10.3390/a16120543

[30] Tian, X., Zhu, L., Zhang, M., Wang, S., Lu, Y., Xu, X., Jia, W., Zheng, Y., & Song, S. ,'Social anxiety prediction based on ERP features: A deep learning approach', Journal of Affective Disorders, 367, 545–553, 2024. doi.org/10.1016/j.jad.2024.09.006