

Predicting Programming Anxiety Among Computer Studies Students Using Automated Machine Learning

Eduardo R. Yu II., Elmerito D. Pineda., Isagani M. Tano., Ace C. Lagman., Jayson M. Victoriano., Jonilo C. Mababa.,
Jaime P. Pulumbarit

La Consolacion University Philippines

DOI: <https://doi.org/10.51583/IJLTEMAS.2025.140500034>

Received: 17 May 2025; Accepted: 22 May 2025; Published: 07 June 2025

Abstract: Predicting mental health-related challenges, such as programming anxiety, is essential for delivering timely support and improving academic outcomes among students in computer-related disciplines. This study examined the use of Automated Machine Learning (AutoML) to predict programming anxiety levels using a dataset composed of demographic, academic, and behavioral attributes collected from students at a public university in the Philippines. Through the Altair AI Studio Educational 2024.0.0, eight classification algorithms were employed to automatically generate predictive models, which were subsequently evaluated to determine their performance. Among the models produced, logistic regression emerged as the best-performing, achieving the highest accuracy (97.8%), F-measure (98.3%), and recall (99.7%), while also offering advantages in computational efficiency and interpretability. Feature importance analysis identified working student status, previous semester general weighted average, multimodal learning preferences, and access to multiple ICT devices as key predictors of programming anxiety. These results underscore the practical utility of AutoML in educational contexts and highlight its potential for enabling early identification and intervention to support students' mental well-being and academic performance in computing disciplines.

Keywords: AutoML, Programming Anxiety, Machine Learning in Education, Logistic Regression, Classification Algorithms

I. Introduction

As computing becomes increasingly integrated into education and the workforce, programming has emerged as a critical 21st-century skill for students in computer-related disciplines (Mahmud & Wong, 2022). Alongside this trend, educational institutions are seeking data-driven solutions to support student learning outcomes. One growing area of interest is the application of machine learning (ML) techniques to understand and address affective factors such as anxiety, which can significantly influence academic performance and persistence in computing programs. Recently, Automated Machine Learning (AutoML) has gained traction in educational data mining because of its ability to streamline the ML pipeline. It allows even non-expert users to develop predictive models quickly and effectively. AutoML platforms present a promising opportunity for educators and researchers to explore insights from educational datasets without requiring deep technical knowledge in machine learning (Baratchi et al., 2024).

Despite the rapid advancement of ML tools in education, programming anxiety remains an underexplored area within the context of predictive analytics. Programming anxiety refers to a learner's emotional discomfort, fear, or apprehension specifically related to learning or performing programming tasks (Jiang et al., 2024; Yildirim & Ozdener, 2022). This form of anxiety can negatively impact students' engagement, academic performance, and persistence in computer-related fields (Charles & Gwilliam, 2023). To quantify programming anxiety, Yildirim & Ozdener (2022) developed structured self-report instruments that capture distinct dimensions of students' emotional responses toward programming tasks. While such instruments offer a validated means of measuring programming anxiety, their integration with machine learning methodologies for predictive analysis remains limited.

Moreover, while AutoML offers a practical approach to model development, there is a lack of empirical research evaluating its performance in real-world academic settings, particularly within Philippine higher education. While prior studies have identified various factors associated with programming anxiety, there remains limited research on determining which attributes are most predictive when applying machine learning techniques. This gap underscores the need to examine the relative importance of these attributes in predictive modeling, thereby bridging the divide between automated machine learning tools and actionable educational insights.

This study aimed to explore the use of Altair AI Studio Educational 2024.0.0, an AutoML platform, for predicting programming anxiety among computer studies students in a public university in the Philippines. Specifically, the study seeks to (1) utilize AutoML to build and evaluate classification models based on a dataset with 19 student attributes; (2) identify the most significant features for predicting programming anxiety levels; and (3) determine the best-performing prediction model in terms of accuracy and interpretability.

This study makes three key contributions that extend and refine prior research on educational machine learning and programming anxiety. First, it is one of the few studies to apply predictive modeling techniques to programming anxiety, a construct that has only recently been quantified through validated instruments. By leveraging an educational AutoML platform, the study demonstrates the practical feasibility of developing predictive models for programming anxiety in real-world academic settings with minimal manual intervention. Second, while earlier studies often focused on isolated factors, this study systematically identifies which student attributes contribute most to the predictive accuracy of the models, offering data-driven insights to support targeted interventions

by educators and administrators. Third, the study provided a comprehensive comparative evaluation result of multiple prediction models generated by the AutoML platform using a variety of classification algorithms, enhanced methodological rigor, and provided practical guidance for future research applying automated machine learning in educational and psychological contexts.

II. Literature Review

Programming Anxiety in Computing Education:

Multiple studies have confirmed that high levels of programming anxiety correlate with poor academic performance, low self-efficacy, and a decline in motivation to persist in computing disciplines (McGrath, 2025; Olipas et al., 2021; Wen et al., 2023). While conventional interventions such as tutoring, peer mentoring, and curricular redesign have yielded moderate success, these approaches are largely reactive in nature. The literature suggests an urgent need for early identification mechanisms capable of detecting anxiety before it adversely affects learning outcomes. Notably, few studies have transitioned from descriptive analyses to predictive frameworks that support proactive intervention strategies.

Supervised Machine Learning in Educational Prediction:

Supervised Machine Learning (SML) has demonstrated considerable promise in predicting various educational outcomes, including academic performance, student engagement, dropout likelihood, and mental health indicators (Faizan et al., 2025; Orji & Vassileva, 2022; Villar & De Andrade, 2024). These models typically utilize diverse input features, such as demographic attributes (e.g., age, gender), academic metrics (e.g., GPA, course grades), and behavioral indicators (e.g., sleep patterns, attendance). While SML techniques have been effectively applied to predict anxiety-related outcomes (e.g., Mahalakshmi & Sujatha, 2023; Mutalib, 2021), they often require extensive domain expertise for tasks such as feature engineering, model selection, and hyperparameter tuning.

Furthermore, the existing body of literature lacks critical examination of the relative advantages and limitations of different algorithms in various educational settings. Although several studies report high model accuracy, discussions surrounding model interpretability, scalability, and ease of deployment are frequently underdeveloped. This limits the practical applicability of these models, particularly in academic institutions with constrained technical resources.

Automated Machine Learning (AutoML) as an Emerging Solution:

Automated Machine Learning (AutoML) has emerged as a promising paradigm to democratize access to advanced predictive modeling by automating the end-to-end machine learning pipeline, including preprocessing, feature selection, algorithm selection, and performance evaluation (Baratchi et al., 2024). Prominent AutoML platforms—such as Google AutoML, H2O AutoML, Auto-Sklearn, Microsoft Azure AutoML, and Altair AI Studio—have been increasingly utilized in educational and healthcare domains (Ferreira et al., 2021; Van Eeden et al., 2021).

Studies such as those by Zeineddine et al. (2020) and Hýbl and Alférez (2024) underscore the capacity of AutoML to yield models with predictive performance comparable to manually curated approaches, while substantially reducing the time and technical expertise required. However, a critical limitation of the current literature is its predominant focus on academic performance outcomes, with limited application of AutoML in predicting affective or psychological constructs such as anxiety. This represents a substantive methodological gap that warrants further investigation.

ML and AutoML in Predicting Mental Health and Anxiety:

A growing number of studies have begun to explore the utility of machine learning techniques in predicting mental health outcomes in educational settings. For instance, Iparraguirre-Villanueva et al. (2024) employed logistic regression to predict depression among college students, whereas Bhatnagar et al. (2023) utilized classification algorithms to model student anxiety levels. Similarly, Mahalakshmi and Sujatha (2023) evaluated multiple algorithms—such as Decision Trees (DT), Support Vector Machines (SVM), Naïve Bayes, and K-Nearest Neighbors (KNN)—and reported that KNN achieved the highest accuracy in predicting mental stress.

While these studies highlight the feasibility of using machine learning for mental health prediction, they often generalize anxiety without accounting for domain-specific stressors—such as those unique to programming instruction. Moreover, the majority rely on traditional machine learning workflows rather than automated solutions, thereby limiting accessibility and scalability, particularly in non-specialist educational contexts.

Student Academic Performance and Anxiety Prediction:

Test anxiety related to programming education has also been examined. Tomić et al. (2022) explored students' anxiety levels in introductory programming courses, revealing that while general and test-specific anxiety were prevalent, prior programming experience did not significantly alleviate programming-related test anxiety. This suggests a deeper interplay between cognitive, emotional, and academic factors unique to programming education.

Although there is a growing body of literature on ML and AutoML in education, most existing studies have concentrated on predicting general academic performance or student attrition. Khor and Darshan (2024), for example, developed supervised learning models based on social network analysis to forecast student success in online learning environments, underscoring how digital behavior metrics can be valuable predictors of academic outcomes. However, there remains a notable paucity of research

specifically targeting programming-related anxiety, particularly through automated modeling approaches that integrate both behavioral and academic predictors.

Research Gap and Study Direction:

The literature reviewed reveals two principal gaps. First, while machine learning and AutoML have been employed to predict academic and mental health outcomes, their application to domain-specific anxieties—such as programming anxiety—remains underexplored. Second, there is a paucity of studies that critically assess the operational feasibility and interpretability of AutoML-generated models in educational environments.

To address these gaps, the present study proposes the use of Altair AI Studio's AutoML platform to develop predictive models for programming anxiety among students in computer studies programs. By integrating academic, behavioral, and demographic predictors, the study aims not only to evaluate the predictive performance of AutoML but also to identify key variables that may inform the design of early, data-driven mental health interventions. This approach contributes to both the methodological advancement of AutoML in education and the practical goal of mitigating programming anxiety through timely and accessible predictive analytics.

III. Methodology

The researchers employ AutoML workflow to facilitate the end-to-end development of predictive models for student programming anxiety level. AutoML offers a systematic approach that automates various stages of the machine learning pipeline, allowing for efficient model building, evaluation, and selection without requiring extensive manual intervention. This methodology supports reproducibility, reduces human bias in model selection, and enhances accessibility for educational researchers. The AutoML process followed in this study is outlined below:

Data Collection: The dataset used in this study was obtained from a public university in the Philippines and includes responses from students enrolled in Bachelor of Science programs in Information Technology (BSIT), Information Systems (BSIS), Computer Science (BSCS), and Entertainment and Multimedia Computing (BSEMC). It includes 19 attributes related to academic performance, behavioral factors, and programming anxiety scores. Ethical considerations and data privacy protocols were strictly observed during the collection and handling of student data.

Data Preprocessing: Prior to modeling, the Student Programming Anxiety Scores from the cross-sectional dataset were reclassified to enable categorical classification of the target variable. The original scores ranged from 11 to 55, where higher scores indicated greater levels of programming anxiety. These were grouped into two categories: “Low” for scores between 11 and 33, and “High” for scores between 34 and 55. Following this, the dataset underwent cleaning and transformation to ensure quality and consistency. This process included handling missing values, encoding categorical variables into numerical formats, and normalizing continuous variables. Automated outlier detection was also performed using models available in Altair AI Studio, specifically Distance-Based Outlier Detection and Local Outlier Factor (LOF), to identify and address rows that exhibited anomalous values. These preprocessing steps helped ensure that extreme or erroneous data points did not unduly influence the model's performance and were essential for preparing the dataset for accurate and reliable algorithmic processing.

Load Data and Prepare Target: In this stage of the AutoML pipeline, the dataset was loaded, and the target variable for classification was defined. The variable of interest, programming anxiety level, was categorized into two classes: “High” and “Low.” A cost matrix was configured to guide the learning process, where correct classifications (True High predicted as High and True Low predicted as Low) were assigned a value of 1.0, while incorrect classifications (True High predicted as Low and True Low predicted as High) were penalized with a cost of -1.0. This matrix aims to reduce misclassification errors and support balanced predictive performance. A preliminary data visualization revealed a class imbalance, with 1,124 instances labeled as “High” anxiety and 591 as “Low” anxiety. Although this imbalance was observed, it was systematically addressed by the AutoML platform, which employs automated techniques to manage skewed class distributions during data preprocessing, model training, and evaluation. These techniques include the application of class weighting, stratified sampling, and metric-based adjustments that are automatically integrated within the pipeline. Consequently, this ensured that the modeling process-maintained fairness and predictive validity without requiring manual rebalancing procedures.

Automatic Feature Selection: In this stage, the AutoML system automatically evaluated the importance of each feature and selected the most relevant ones for the prediction task. The researchers chose to perform Automatic Feature Engineering by selecting an option that triggered the creation and validation of hundreds of thousands of potential models. While this significantly enhanced the feature selection process, it also increased the overall modeling time due to the extensive exploration of feature combinations and transformation.

Algorithm Selection: Altair AI Studio platform offers a comprehensive suite of machine learning algorithms for model development, specifically designed for classification tasks. The platform automatically explores a variety of supervised learning algorithms to determine the most suitable for predicting categorical outcomes. These algorithms include Naive Bayes (NB), Generalized Linear Model (GLM), Logistic Regression (LR), Fast Large Margin (FLM), Deep Learning (DL), Decision Trees (DT), Random Forest (RF), Gradient Boosting Machines (GBM), and Support Vector Machines (SVM). By evaluating these diverse algorithms, the software ensures that the best-performing model is selected based on the characteristics of the dataset and the

classification objectives.

Hyperparameter Optimization: For each selected algorithm, the system performed automated hyperparameter tuning to identify the optimal configuration that would maximize model performance while ensuring generalizability to unseen data. In Altair AI Studio, this process was facilitated through built-in options for algorithm-specific optimization. For instance, GLM included a checkbox option labeled “Use Regularization” to control model complexity and prevent overfitting. Additionally, algorithms such as FLM, DT, RF, GBT, and SVM provided an “Automatically Optimize” checkbox, which enabled the platform to automatically fine-tune critical parameters. This streamlined the optimization process and ensured that each algorithm was assessed under its most favorable configuration.

Model Evaluation and Validation: The final models were evaluated through cross-validation using a comprehensive suite of performance metrics available in the Altair AIL Studio educational software. These metrics included accuracy, precision, recall, F-measure, specificity, sensitivity, and classification error percentage, enabling a multidimensional evaluation of model effectiveness. The classification error percentage metric provided a direct estimate of the misclassification rate, offering an essential counterpoint to accuracy in assessing predictive reliability. Additionally, gains, total time, and training time are also tracked, supporting comparative analysis of computational efficiency and resource requirements. The platform’s AutoML functionality further streamlined model selection by automatically tagging candidate models with performance indicators such as Best Performance, Best Gain, Fastest Scoring Time, and Fastest Total Time. These tags facilitated rapid identification of optimal algorithms across different evaluation dimensions. The highest-ranked model was subsequently interpreted to uncover the most influential features contributing to programming anxiety, offering empirically grounded insights to inform targeted student support strategies.

IV. Results and Discussion

The findings are detailed in the following sections, covering the results of automatic outlier detections, automatic feature selection, performance evaluation of classification algorithms for predicting students’ programming anxiety levels.

Automatic Outlier Detections

The results from the Distance-Based Outlier Detection (DBOD) and Local Outlier Factor (LOF) models revealed notable differences in the identification of outliers within the dataset. DBOD tagged rows 1 to 17 as true outliers, indicating that these data points exhibited extreme or unusual distances relative to their neighbors, which could significantly influence the accuracy of model predictions. On the other hand, LOF identified a larger range of potential outliers, tagging rows 1 to 1163 with LOF scores ranging from 1.141 to 1.001. While these values suggested a range of local anomalies, the scores were not extreme enough to conclusively categorize the data points as outliers in the context of this study. The results from DBOD, which are visually summarized in Figure 1, highlight the rows (1 to 17) that were identified as outliers and subsequently removed from the dataset. Based on these DBOD results, the researchers chose to remove rows 1 to 17 as outliers to prevent potential distortion of the model’s performance by extreme or erroneous data points. This decision was made to ensure that the model was trained on reliable, representative data, enhancing its generalizability and predictive accuracy.

variable, these variables were retained from the initial attribute selection process. Age and Gender were also recognized as potentially sensitive attributes that could introduce bias into the predictive model. Nevertheless, their inclusion was considered necessary to allow the automated feature selection process to determine their actual contribution to model performance. This decision was further supported by the study of Mutalib (2021), who included Age and Gender as an explanatory variable in her research on mental health prediction. Adopting a similar approach ensured that attributes with potentially meaningful influence were not excluded solely based on initial attribute recommendation metrics as illustrated in Figure 2.

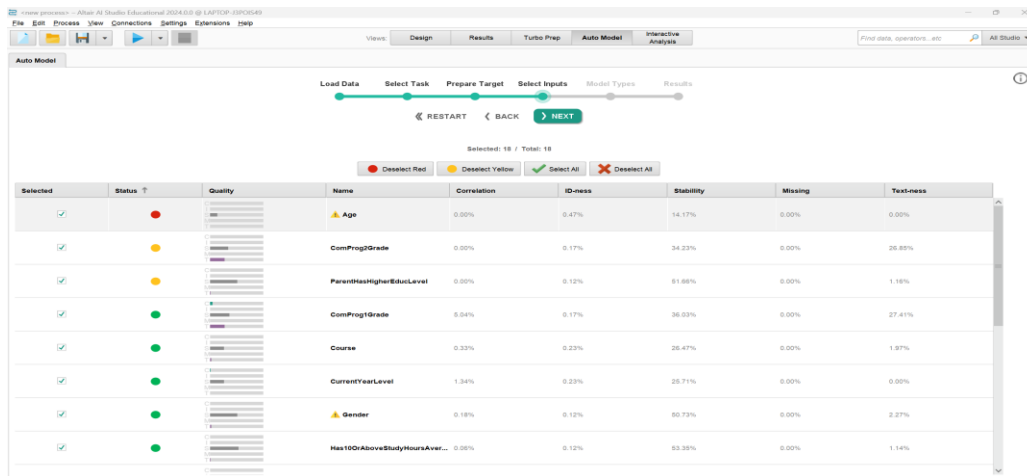


Fig. 2. Results from Automatic Feature Quality Recommendations

Significance Analysis Results of All Attributes

The bar chart presented in Figure 3 illustrates the global importance of each input attribute in relation to students' programming anxiety levels, based on their individual correlation with the target variable. Generated after the model experimentation phase using the Altair AI Studio platform, these correlation-based weights capture the strength and direction of association between each predictor and programming anxiety, independent of any specific algorithm or feature engineering process. The attribute *IsWorkingStudent* exhibited the highest weight (0.344), indicating a strong positive association with elevated programming anxiety, which may be linked to increased cognitive demands and time constraints experienced by working students. This was followed by *PrevSemGPA* (0.297) and *IsMultiModal* (0.281), suggesting that lower academic performance and engagement with multimodal learning formats may be contributing factors to heightened anxiety. Additional attributes such as *HasMultipleICTEquip* (0.273) and *Has6HrsOrAboveSleepAverage* (0.272) also demonstrated notable associations, emphasizing the role of technological access and adequate rest in students' emotional well-being. In contrast, features like *Age* (0.000), *ParentHasHigherEducLevel* (0.007), and *ComProg2Grade* (0.022) showed negligible correlation with the target variable, indicating limited independent importance. These findings highlight key correlates of programming anxiety and can inform data-driven intervention strategies to support student well-being and academic persistence.

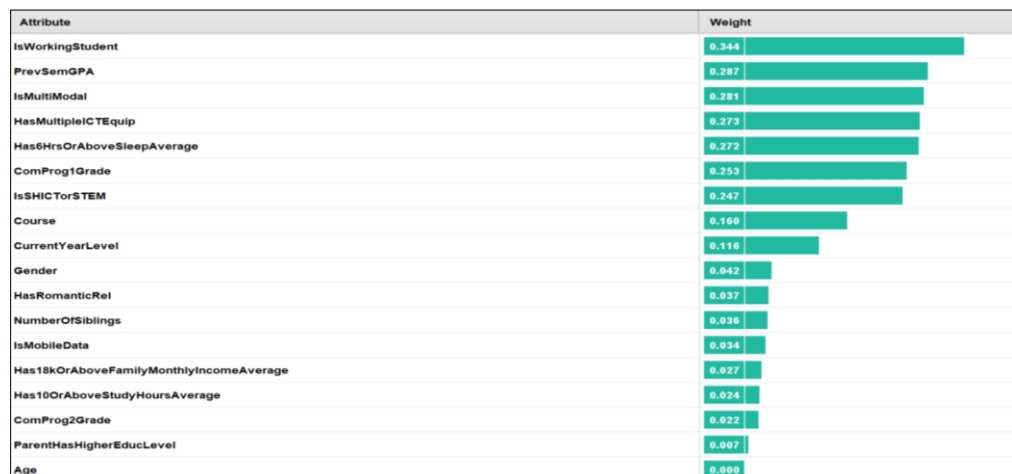


Fig. 3. Coefficient Chart of Weight by Correlation

The correlation matrix in Figure 4 presents the pairwise Pearson correlation coefficients among all attributes considered in the study, including the target variable *PAS2LEVEL*, which represents programming anxiety levels. As shown, most features exhibit weak correlations with *PAS2LEVEL*, with values generally below ± 0.30 , suggesting a limited linear relationship between

Attributes	Age	CompPro...	CompPro...	CompPro...	CompPro...	Course...	Course...	Course...	Current...	Gender...	Has100...	Has10R...	Has10L...	HasMath...	HasSci...	IsMath...	IsMath...	IsSci...	IsWork...	Number...	Parent...	PAS2LE...	Prev...	Prev...
Age	1	-0.036	0.026	-0.001	-0.015	0.046	-0.007	-0.022	0.012	-0.038	0.053	0.040	-0.007	0.043	-0.027	-0.019	0.027	-0.043	-0.037	-0.010	-0.004	-0.000	-0.018	-0.019
CompProgGrade > Above Average	-0.036	1	-0.523	-0.019	-0.001	0.004	0.004	-0.006	-0.042	-0.055	-0.004	0.005	-0.001	0.020	-0.007	-0.015	-0.041	-0.003	-0.003	0.037	0.011	0.253	-0.039	0.024
CompProgGrade < Average	0.026	-0.523	1	0.006	0.003	0.001	-0.006	-0.012	0.016	-0.020	0.008	-0	0.023	-0.001	0.016	-0.038	0.011	0.012	0.034	0.021	-0.011	-0.126	0.019	-0.020
CompProgGrade > Above Average	-0.001	-0.019	0.006	1	-0.512	-0.018	0.019	0.001	0.008	0.001	-0.027	0.003	-0.013	-0.028	-0.005	-0.012	0.026	0.029	-0.055	-0.015	-0.035	-0.015	0.010	-0.009
CompProgGrade < Average	-0.015	0.001	0.003	-0.512	1	0.013	0.030	0.001	-0.004	0.013	0.004	-0.054	-0.007	0.022	0.007	0.049	-0.029	0.002	0.050	0.003	0.048	0.021	0.000	-0.025
Course = BIOSC	0.046	0.004	0.001	-0.016	0.013	1	-0.349	-0.330	-0.002	0.008	0.017	0.018	0.034	-0.018	-0.063	0.011	-0.028	-0.023	-0.049	0.003	0.018	-0.100	0.036	0.001
Course = BIOSM	-0.007	0.004	-0.006	0.019	0.030	-0.349	1	-0.340	-0.022	0.018	-0.048	0.020	0.001	-0.017	0.039	-0.006	-0.012	-0.006	-0.060	-0.050	0.020	0.115	-0.006	0.003
Course = EBST	-0.022	-0.006	-0.012	0.011	0.001	-0.330	-0.340	1	0.032	0.012	0.012	0.007	-0.027	0.021	-0.002	0.005	0.038	0.008	-0.029	0.011	-0.020	0.055	-0.016	-0.002
CurrentExamLev	0.012	-0.042	0.018	0.008	-0.004	-0.002	-0.022	0.032	1	0.020	-0.011	-0.025	0.014	0.042	0.048	-0.017	-0.001	0.016	-0.032	-0.013	-0.045	-0.015	0.017	0.009
Gender = Female	-0.038	0.055	-0.020	0.001	0.013	-0.008	0.018	0.012	0.020	1	0.025	0.022	-0.014	-0.040	-0.009	0.019	-0.033	0.012	-0.030	-0.031	-0.041	-0.042	-0.032	0.001
Has10HoursOfSelfHoursAverage > No	0.053	-0.004	-0.008	-0.027	0.004	0.017	-0.048	0.012	-0.011	0.025	1	0.017	-0.019	-0.003	-0.018	-0.019	-0.009	-0.041	-0.005	0.022	0.012	-0.024	0.011	-0.021
Has10HoursOfSelfHoursAverage < No	-0.004	0.005	-0	0.003	-0.054	0.018	0.020	-0.007	-0.025	0.022	0.017	1	-0.016	0.003	0.000	0.033	0.003	0.046	-0.008	0.049	0.029	0.027	0.008	-0.007
Has10HoursOfSelfHoursAverage > No	-0.007	-0.001	0.023	-0.013	-0.007	0.034	0.001	-0.027	0.014	-0.019	-0.016	1	-0.032	-0.001	-0.037	-0.026	-0.024	-0.042	-0.021	-0.014	0.272	-0.009	0.020	
HasMathAPCEase > No	0.043	0.020	-0.001	-0.028	0.022	-0.019	-0.017	0.021	0.042	-0.040	-0.003	0.003	-0.032	1	0.045	-0.009	-0.003	0.022	-0.015	0.022	-0.028	0.273	-0.012	-0.047
HasMathAPCEase < No	-0.027	-0.007	0.015	-0.005	0.007	-0.063	0.030	-0.002	0.048	-0.009	-0.018	0.000	-0.001	0.045	1	0.027	-0.018	0.027	0.016	-0.028	-0.002	0.077	0.031	-0.047
IsMathCats > No	-0.019	0.015	-0.038	-0.012	0.049	0.011	-0.006	0.005	-0.017	0.019	0.019	0.033	-0.007	-0.009	0									

Performance Evaluation of Classification Algorithms for Predicting Students' Programming Anxiety Levels

The evaluation results, presented in Tables 1 and 2, offered a comparative overview of classification performance and computational efficiency across various models generated through the AutoML platform. Logistic Regression (LR) emerged as the top-performing model, achieving the highest overall accuracy (97.8%) with excellent sensitivity (99.7%) and specificity (94.1%). It also demonstrated computational efficiency, requiring only 80.5 milliseconds per 1,000 rows and a total processing time of approximately 7 minutes and 1 second. LR's strong performance may be attributed to its ability to model linear relationships effectively while maintaining low variance, making it well-suited for datasets with well-separated classes and limited noise.

The Fast Linear Model (FLM), which achieved 100% sensitivity and the fastest scoring time (71.4 milliseconds), had a lower specificity (90.6%) and a longer total processing time (15 minutes 34 seconds). FLM's fast inference is ideal for real-time applications, but its lower specificity suggests potential overfitting to the positive class, possibly due to class imbalance or feature noise.

Support Vector Machine (SVM) and Generalized Linear Model (GLM) yielded moderate accuracy rates (94.1% and 89.4%, respectively). SVM's relatively high scoring time (352.8 milliseconds) reflects the algorithm's sensitivity to feature scaling and kernel complexity, which may not have been optimal in this automated setup. GLM, a generalization of LR, underperformed likely due to its simpler structure and potential inability to capture more nuanced relationships in the data.

Naive Bayes (NB), while offering the shortest total runtime (2 minutes 48 seconds) and perfect sensitivity (100%), had the lowest specificity (54.7%). This reflects the algorithm's strong bias and independence assumptions, which are often violated in real-world educational datasets, leading to misclassification of negative cases.

Deep Learning (DL) and Random Forest (RF) both showed strong sensitivity (99.1% and 97.5%, respectively) but were slower

overall (7 minutes 51 seconds and 8 minutes 1 second total time, respectively). DL's lower precision (81.0%) may be due to overfitting or sensitivity to hyperparameter initialization, particularly in the absence of extensive tuning within AutoML. RF, an ensemble of decision trees, typically performs well on tabular data but may have suffered from higher model complexity and redundancy among features.

Overall, the AutoML platform selected Logistic Regression as the most balanced model in terms of classification performance, generalization ability, and runtime efficiency. Its strong performance underscores the advantage of simpler, interpretable models when the feature space is well-structured, and the classes are linearly separable. Nonetheless, the final choice of model should be guided by application-specific constraints, such as the acceptable trade-off between false positives and false negatives, interpretability requirements, and available computational resources.

Table 1 Performance Results of Classification Algorithms

Model	Accuracy	Precision	Recall	F Measure	Specificity	Sensitivity	Gains
LR	97.8%	97.0%	99.7%	98.3%	94.1%	99.7%	316.0
GBT	97.1%	97.0%	98.8%	97.9%	94.1%	98.8%	308.0
FLM	96.7%	95.3%	100.0%	97.6%	90.6%	100.0%	306.0
SVM	94.1%	93.3%	98.1%	95.6%	86.4%	98.1%	280.0
GLM	89.4%	87.7%	97.5%	92.3%	74.1%	97.5%	236.0
RF	87.8%	87.7%	94.7%	91.0%	74.6%	94.7%	218.0
NB	84.5%	80.9%	100.0%	89.5%	54.7%	100.0%	184.0
DL	84.1%	81.0%	99.1%	89.1%	55.6%	99.1%	182.0

Table 2 Performance Results of Classification Algorithms using Time Metrics

Model	Total Time	Training Time for 1,000 Rows (milliseconds)	Scoring Time for 1,000 Rows (milliseconds)
LR	7 min 1 sec	80.5	97.7
GBT	8 min 32 sec	647.2	172.0
FLM	15 min 34 sec	61.2	71.4
SVM	7 min 50 sec	302.0	352.8
GLM	7 min 9 sec	86.9	123.9
RF	8 min 1 sec	64.7	476.7
NB	2 min 48 sec	58.3	437.3
DL	7 min 51 sec	787.2	138.5

The classification error rates of the evaluated machine learning models demonstrated substantial variation, reflecting important differences in their predictive effectiveness for programming anxiety level classification. Logistic Regression (LR) produced the lowest classification error at 2.2%, indicating it was the most accurate in minimizing misclassifications. Gradient Boosted Trees (GBT) and Fast Large Margin (FLM) followed closely, with error rates of 2.9% and 3.3%, respectively. The Support Vector Machine (SVM) model exhibited a moderate error rate of 5.9%, while Generalized Linear Model (GLM) and Random Forest (RF) yielded higher rates of 10.6% and 12.2%, respectively. In contrast, Naïve Bayes (NB) and Deep Learning (DL) models showed relatively poor performance, with classification errors of 15.5% and 15.9%. The Decision Tree (DT) model performed the worst, with the highest classification error at 21.2%. These results, presented in Figure 4, highlighted the clear advantage of simpler linear models in this context, suggesting that model complexity does not necessarily guarantee improved performance. The findings further emphasized the importance of selecting algorithms that are well-suited to the structure and characteristics of educational data, especially when addressing nuanced psychological outcomes such as programming anxiety.

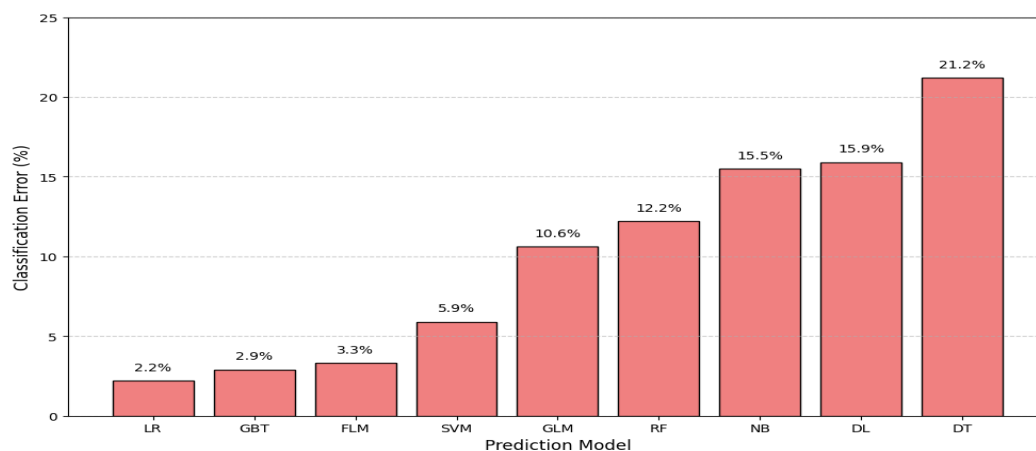


Fig. 5. Classification Error Metric Comparison Chart for Programming Anxiety Level Prediction

Figure 6 presents the Receiver Operating Characteristic (ROC) curve comparison chart, which visually demonstrates the performance of nine machine learning models based on their true positive and false positive rates. The models are represented by different curves, with the area under the curve (AUC) indicating the overall effectiveness of each classifier. From the plot, the Fast Large Margin (AUC = 0.998) and Gradient Boosted Trees (AUC = 0.995) perform exceptionally well, demonstrating near-perfect discrimination between the classes. These models have a nearly perfect trade-off between sensitivity and specificity, meaning they can accurately identify both positive and negative instances. Logistic Regression (AUC = 0.987) and Support Vector Machine (AUC = 0.985) also exhibit strong performance, with AUC values suggesting they are highly effective in distinguishing between the classes, though slightly less so than the top performers. On the other hand, Naive Bayes (AUC = 0.977) and Deep Learning (AUC = 0.983) also show strong performance, but with slightly lower AUCs compared to the top models. Decision Tree (AUC = 0.841) has the lowest AUC, indicating it struggles more than other models in accurately classifying instances, with a greater tendency to misclassify. In summary, the ROC curve and AUC values provided a comprehensive view of each model's ability to correctly identify instances across varying thresholds, with Fast Large Margin and Gradient Boosted Trees emerged as the best performers in terms of AUC, while Decision Tree demonstrates limited classification accuracy.

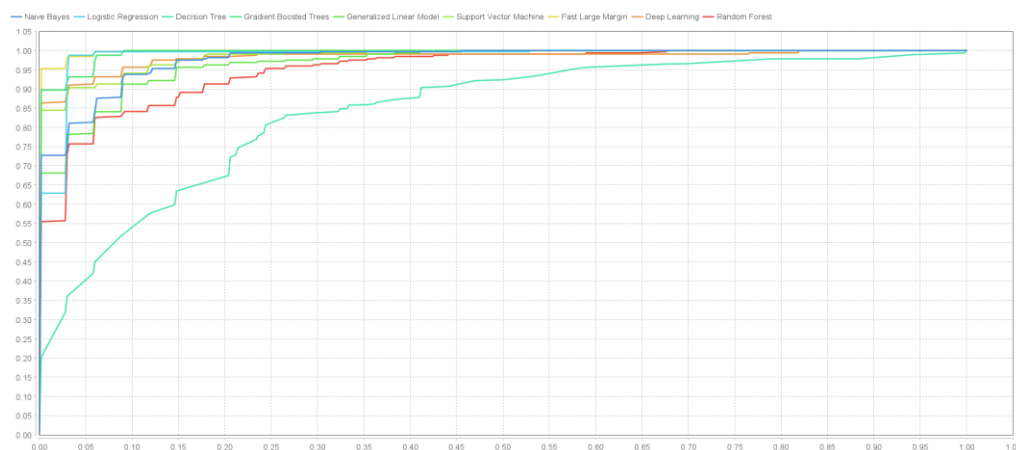


Fig. 6. ROC Curve Comparison Chart of Programming Anxiety Level Prediction

Best Model for Predicting Student's Programming Anxiety Based on AutoML Results

The AutoML platform Altair AI Studio identified Logistic Regression as the best-performing model for predicting students' programming anxiety levels. As part of its automated pipeline, the platform produced a feature importance chart (Figure 7) that quantitatively ranked input attributes by their relative contribution to the model's predictive performance. The most influential features were IsWorkingStudent (0.254) and Has6HrsOrAboveSleepAverage (0.253), indicating that students who balance employment alongside their studies and those who maintain sufficient sleep are key differentiators in predicting anxiety levels. This is followed closely by IsSHICTorSTEM (0.232), suggesting that students previously enrolled in ICT or STEM senior high school programs are particularly salient in the classification process. Academic performance indicators such as PrevSemGPA (0.147), ComProg1Grade (0.145), and ComProg2Grade (0.036) also emerged as relevant, reinforcing the role of academic preparedness in anxiety prediction. Technological access, as reflected in HasMultipleICTEquip (0.140), and learning modality preferences, indicated by IsMultiModal (0.124), further contributed to the model's predictive accuracy. Meanwhile, variables such as Gender

(0.032), ParentHasHigherEducLevel (0.022), and NumberOfSiblings (0.011) were found to have minimal influence. These results were derived under a cost-sensitive classification framework that prioritized accurate identification of students with high programming anxiety and reflected the model's capacity to surface both behavioral and academic factors as key predictors. Such insights support data-driven approaches for identifying and supporting students most at risk of anxiety in programming courses.

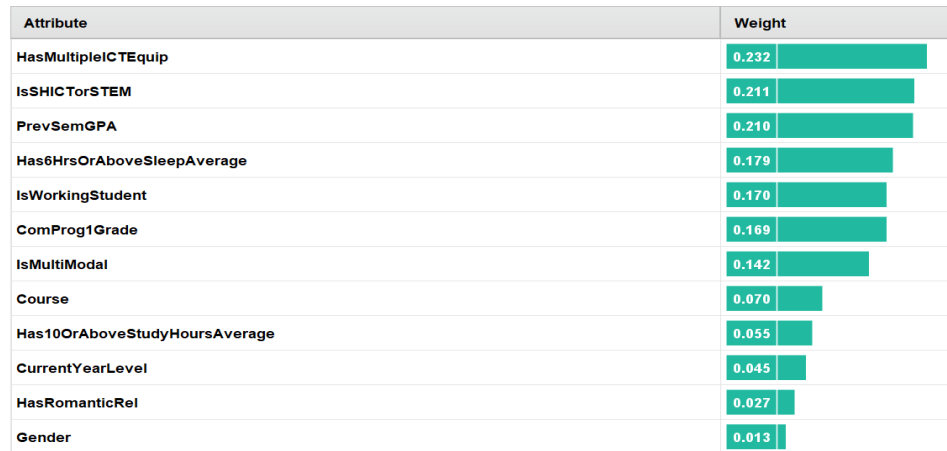


Fig. 7. Results of Automatic Feature Selection for Logistic Regression

The logistic regression model was automatically generated by the platform and revealed a statistically grounded perspective on the predictive influence of various academic and demographic factors associated with high programming anxiety. Among all predictors, the academic program of the student emerged as a significant factor. Students enrolled in the BSIT ($\beta = 10.047$, $p < 0.001$) and BSEMC ($\beta = 9.811$, $p < 0.001$) programs were significantly more likely to exhibit high levels of programming anxiety compared to the reference group, suggesting that program-specific curricula or instructional practices may influence anxiety levels. In contrast, students in the BSIS program did not show a statistically significant association ($p = 0.390$).

The student's current year level also showed a statistically significant negative association with anxiety ($\beta = -4.068$, $p < 0.001$), indicating that students further along in their academic program were less likely to report high anxiety—possibly due to accumulated programming experience or increased confidence. Other features such as prior programming grades, GPA, study habits, and behavioral indicators showed relatively large coefficient values but did not achieve statistical significance, likely due to high standard errors or limited variability within the sample. Furthermore, the variable “Is Working Student” had a high positive coefficient ($\beta = 32.598$) yet was not statistically significant ($p = 0.152$), suggesting a possible relationship that merits further exploration. Attributes such as Age and Gender were retained in the model due to their theoretical relevance in anxiety-related literature, despite their low statistical impact in this specific dataset. Overall, the results in Table 2 highlight the relevance of academic program and year level as the most reliable predictors in this logistic regression model, providing valuable insights for targeted educational interventions.

Table 2 Automatically Generated Logistic Regression Model Coefficients for Predicting Programming Anxiety Levels

Attribute	Coefficient	Std. Coefficient	Std. Error	z-Value	p-Value
Course.BSEMC	9.811	9.811	2.004	4.895	0.000
Course.BSIS	-1.126	-1.126	1.311	-0.859	0.390
Course.BSIT	10.047	10.047	2.025	4.961	0.000
Course.MISSING	0.000	0.000			
ComProg1Grade.Average	-30.402	-30.402	22.764	-1.335	0.182
ComProg1Grade.Below Average	-29.620	-29.620	22.768	-1.301	0.193
ComProg1Grade.MISSING	0.000	0.000			
PrevSemGPA.Average	-34.309	-34.309	22.799	-1.505	0.132
PrevSemGPA.Below Average	-34.664	-34.664	22.797	-1.521	0.128
PrevSemGPA.MISSING	0.000	0.000			
Has10OrAboveStudyHoursAverage.Yes	-0.275	-0.275	0.742	-0.370	0.711
Has10OrAboveStudyHoursAverage.MISSING	0.000	0.000			

Has6HrsOrAboveSleepAverage.Yes	-28.811	-28.811	22.755	-1.266	0.205
Has6HrsOrAboveSleepAverage.MISSING	0.000	0.000			
HasMultipleICTEquip.Yes	-28.414	-28.414	22.753	-1.249	0.212
HasMultipleICTEquip.MISSING	0.000	0.000			
HasRomanticRel.Yes	-0.796	-0.796	0.822	-0.969	0.333
HasRomanticRel.MISSING	0.000	0.000			
IsMultiModal.Yes	-28.432	-28.432	22.753	-1.250	0.211
IsMultiModal.MISSING	0.000	0.000			
IsSHICTorSTEM.Yes	-28.821	-28.821	22.755	-1.267	0.205
IsSHICTorSTEM.MISSING	0.000	0.000			
IsWorkingStudent.Yes	32.598	32.598	22.776	1.431	0.152
IsWorkingStudent.MISSING	0.000	0.000			
Gender.Male	-0.190	-0.190	0.765	-0.249	0.803
Gender.MISSING	0.000	0.000			
CurrentYearLevel	-4.068	-4.558	0.823	-4.941	0.000
Intercept	91.883	81.868	68.273	1.346	0.178

V. Conclusion

This study concluded that the logistic regression model was the most effective among the classification algorithms generated by the Altair AI Studio Educational 2024.0.0 platform in predicting programming anxiety. It achieved the highest overall performance in terms of accuracy (97.8%), F-measure (98.3%), and recall (99.7%), while also maintaining strong precision (97.0%) and the highest specificity (94.1%) among the top-performing models. These results indicate that logistic regression is not only reliably identified students experiencing programming anxiety but also minimized false positives, an important consideration for practical interventions in educational settings. Additionally, analysis of feature importance revealed that the 9 student attributes were among the most significant predictors of programming anxiety level, suggesting that academic performance, learning modality, access to technology, and lifestyle factors play a central role in students' emotional responses to programming. These features offer actionable insights for educators and administrators aiming to design proactive support mechanisms. The model also demonstrated computational efficiency, recording the highest gains score (316.0) and one of the lowest training times (80.5 ms per 1,000 rows). Collectively, these findings affirm the practical utility of AutoML in educational data mining and highlight logistic regression as both a robust and interpretable model for supporting early identification of at-risk students in higher education institutions in the Philippines.

References

1. Baratchi, M., Wang, C., Limmer, S., & others. (2024). Automated machine learning: Past, present and future. *Artificial Intelligence Review*, 57, 122. <https://doi.org/10.1007/s10462-024-10726-1>
2. Charles, T., & Gwilliam, C. (2023). The effect of automated error message feedback on undergraduate physics students learning Python: Reducing anxiety and building confidence. *Journal for STEM Education Research*, 6(2), 326–357. <https://doi.org/10.1007/s41979-022-00084-4>
3. Faizan, R., Dhyani, S., Singh, P., & Rawat, D. (2025). Machine learning approaches for predicting and improving student mental health. In *Lecture Notes in Networks and Systems* (pp. 123–131). https://doi.org/10.1007/978-981-96-1267-3_12
4. Jiang, Y., Wu, H., Yu, X., & Ji, T. (2024). A study of factors influencing programming anxiety among non-computer students. In *Proceedings of the 2024 9th International Conference on Information and Education Innovations (ICIEI '24)* (pp. 63–69). Association for Computing Machinery. <https://doi.org/10.1145/3664934.3664956>
5. Khor, E. T., & Darshan, D. (2024). Prediction of students' performance in online learning using supervised machine learning. *International Journal of Information and Learning Technology*, 41(2), 166–179. <https://doi.org/10.1108/ijilt-08>
6. Mahalakshmi, G., & Sujatha, G. (2023). Predictions of college students' mental stress using machine learning algorithms. In *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 288–295). IEEE. <https://doi.org/10.1109/ICICCS56967.2023.10142672>
7. Mahmud, M. M., & Wong, K. W. (2022). Digital age: The importance of 21st century skills among undergraduates. *Frontiers in Education*, 7, 950553. <https://doi.org/10.3389/feduc.2022.950553>
8. McGrath, K., Negoita, S., Sorrentino, Z., Laurent, D., Pierre, K., Solar, S., ... Koch, M. (2025). Case study: Traumatic

- dural arteriovenous fistula – An opportunity for AI to accelerate diagnosis. *Academia Medicine*, 2(1). <https://doi.org/10.20935/AcadMed7622>
9. Mutalib, S. (2021). Mental health prediction models using machine learning in higher education institutions. *Turkish Journal of Computer and Mathematics Education*, 12(5), 1782–1792. <https://doi.org/10.17762/turcomat.v12i5.2181>
10. Olipas, C. N. P., Leona, R. F., Villegas, A. C. A., Cunanan, A. I., Jr., & Javate, C. L. P. (2021). The academic performance and the computer programming anxiety of BSIT students: A basis for instructional strategy improvement. *International Journal of Advanced Engineering Management and Science*, 7(6), 125–129. <https://doi.org/10.22161/ijaems.76.15>
11. Orji, F. A., & Vassileva, J. (2022). Machine learning approach for predicting students academic performance and study strategies based on their motivation. *arXiv*. <https://doi.org/10.48550/arxiv.2210.08186>
12. Tomić, B., Stojanović, T., & Stojanović, T. (2022). Examining students' test anxiety and pre-university programming education in an undergraduate introductory programming course. *EDULEARN Proceedings*. <https://doi.org/10.21125/edulearn.2022.0938>
13. Van Eeden, W. A., Luo, C., Van Hemert, A. M., Carlier, I. V., Penninx, B. W., Wardenaar, K. J., Hoos, H., & Giltay, E. J. (2021). Predicting the 9-year course of mood and anxiety disorders with automated machine learning: A comparison between auto-sklearn, naïve Bayes classifier, and traditional logistic regression. *Psychiatry Research*, 299, 113823. <https://doi.org/10.1016/j.psychres.2021.113823>
14. Villar, A., & De Andrade, C. R. V. (2024). Supervised machine learning algorithms for predicting student dropout and academic success: A comparative study. *Discover Artificial Intelligence*, 4(1). <https://doi.org/10.1007/s44163-023-00079-z>
15. Wen, F., Wu, T., & Hsu, W. (2023). Toward improving student motivation and performance in introductory programming learning by Scratch: The role of achievement emotions. *Science Progress*, 106(4). <https://doi.org/10.1177/00368504231205985>
16. Yildirim, O. G., & Ozdener, N. (2022). Development and validation of the Programming Anxiety Scale. *International Journal of Computer Science Education in Schools*, 5(3), 17–34. <https://doi.org/10.21585/ijcses.v5i3.140>
17. Zeineddine, H., Braendle, U., & Farah, A. (2020). Enhancing prediction of student success: Automated machine learning approach. *Computers & Electrical Engineering*, 89, 106903. <https://doi.org/10.1016/j.compeleceng.2020.106903>