Data Science and Engineering (DSE) Record, Volume 5, issue 1.

Learning Outcomes Classification Model Based on Curriculum Guidelines for Undergraduate Programs in Computer Science

Pantaree Pitivaranun[1] Dussadee Praserttitipong[2] and Wijak Srisujjalertwaja[3]

[1] Master's Degree Program in Data Science, Chiang Mai University, Chiang Mai, Thailand
[2,3] Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai, Thailand
pantaree_p@cmu.ac.th
dussadee.p@cmu.ac.th
wijak.s@cmu.ac.th

Abstract. This study emphasizes the critical role of course learning outcomes, particularly in assessing student capability, mainly in the cognitive learning framework provided by Bloom's Taxonomy. In computer science education, aligning these outcomes with curriculum guidelines is important for program quality and relevance. The study introduces machine learning models, including Multinomial Naive Bayes, Logistic Regression, Random Forest, and Extreme Gradient Boosting (XG Boost), to predict and visualize course learning outcomes classification using radar charts. The primary aim is to establish a classification model aligning with ACM/IEEE undergraduate computer science program curriculum guidelines. Additionally, the study addresses the ambiguity inherent in Bloom's Taxonomy, where the same action verb may span multiple cognitive levels, potentially confusing in defining learning objectives across Familiarity, Usage, and Assessment domains. Through a semi-automated prototype, the study showcases a scalable and adaptable framework for visualizing learning outcomes classification results by radar charts. This framework is intended to benefit educators, curriculum developers, and accreditation bodies, enhancing the coherence and effectiveness of computer science undergraduate programs.

Keywords: Course Objective Classification, Learning Outcomes Classification, Curriculum Development

1 Introduction

In recent times, there is more emphasis on the importance of students carefully deliberating on their educational pathway. It acknowledges the significance of finding inspiration and maintaining focus throughout their academic pursuits. Moreover, educational institutions are urged to intricately tailor their curriculum to align with the skills and competencies projected to be in high demand in the evolving job market. Learning outcomes serve as crucial benchmarks in academic settings by explicitly defining the skills and potential expected upon completing a course, learning outcomes provide transparency, guiding focused learning and creating a basis for effective assessment.

Additionally, education has witnessed a significant paradigm shift towards outcome-based education (OBE) or performance-based education, which defines what students are expected to know and do upon completing their education and emphasizes measuring educational effectiveness by assessing results rather than inputs like class time [1] and outcome-based education (OBE) can define clear learning outcomes aligned with curriculum objectives. This shift has been particularly pronounced in undergraduate programs in computer science, where technology has emerged as one of the most prominent and captivating subjects of interest, particularly among students. Students are strongly encouraged to carefully delineate their educational trajectories, considering their aspirations and preferred areas of concentration in computer science programs. Furthermore, educational institutions should align the course curriculum to provide students with their demands, knowledge, skills and competencies that demand in the forthcoming.

Learning outcomes classification plays a crucial role in helping students comprehend their competencies for prospective job searches. Extracting skills is a pivotal endeavor in the development of job recommendation systems. Furthermore, it holds significance in the construction of skills profiles and knowledge bases dedicated to skills within organizations [2]. The concept of skill, although challenging to pinpoint precisely, typically denotes the attributes possessed by employees and the aptitudes necessary for the successful execution of specific responsibilities. Skills can be categorized into cognitive proficiencies, including literacy and numeracy, as well as non-cognitive aptitudes like teamwork and various behavioral attributes that are essential for the diverse array of tasks integral to a given occupation [3].

Foundational to competency, knowledge comprises the "know-what" dimension, encapsulating subject matter acknowledged by educators, academic programs, accrediting bodies, and employers. Skills, integral to the overall competency framework, constitute the "know-how" dimension, empowering individuals to actively apply knowledge in task accomplishment, with their proficiency observable in work processes. The critical aspect of skills, evident in work outcomes, underscores that the true value of knowledge hinges on its application at a specific level of skillfulness. Dispositions, shaping the "know-why" dimension, guide the quality of character in task performance, bridging knowledge application and skillful execution with contextual factors. Finally, tasks delineate the concrete setting for competency, defining purposeful engagement, providing a pragmatic context for program development, and facilitating graduates in demonstrating competency aligned with the program's envisioned outcomes [4].

Moreover, Technology has emerged as one of the most prominent and captivating subjects of interest, particularly among students. This heightened enthusiasm is a direct result of the rapid advancements in technology, the proliferation of automation, and the ever-expanding utilization of big data and the Internet of Things (IoT).

The study aims to establish a classification model for learning outcomes that correspond with the ACM/IEEE undergraduate computer science program curriculum guidelines. This study also develops and visualizes a semi-automates prototype of the classification process by integrating Bloom's Taxonomy with existing datasets for clearness classification addressing the ambiguity of the action verbs in Bloom's Taxonomy.

2 Literature Review

2.1 Natural Language Processing (NLP)

Natural Language Processing (NLP) constitutes a subfield within the realm of Artificial Intelligence, primarily dedicated to the computational interpretation of linguistic phenomena. This interdisciplinary domain encompasses diverse facets of textual and auditory data analysis, extensively leveraging statistical machine learning techniques. Furthermore, it encapsulates an expansive scope of research endeavors in computational linguistics, steadily advancing in breadth and potency through the application of diverse methodological approaches and techniques [5].

2.2 Naïve Bayes (NB)

The Naïve Bayes (NB) text classifier produces its classification model as a result of learning (estimation) process based on the Naïve Bayes learning algorithm which belongs to a family of probabilistic classifiers based on the Bayes theorem. For classifying a given document, Naïve Bayes learning system estimates the posterior probability of each class via Bayes rule; that is, $\Pr(c|d) = \frac{\Pr(c)\cdot\Pr(d|c)}{\Pr(d)}$, where Pr(c|d) is the probability that a document d belongs to the class c in a set of classes C, Pr(c) is the class prior probability that any random document from the document corpus belongs to the class c, Pr(d|c) is the probability that a randomly chosen document from documents in the class c is the document d, and Pr(d) is the probability that a randomly chosen document from the whole corpus is the document d. The document d is then assigned to a $class\ argmax_{c\in C}\Pr(c|d)(=\Phi_{\hat{\theta}_{NB}}(d))$ with the highest posterior probability. Here, in the context of the Naïve Bayes, the document d is represented by a bag of words (t1,t2,...,t|d|), where multiple occurrences of words are preserved. Moreover, the Naïve Bayes assumes that the terms in a document are mutually independent and the probability of term occurrence is independent of position within the document given a class [6].

2.3 Logistic Regression (LR)

Logistic regression, a fundamental analytical tool in the realms of both social and natural sciences, holds a central position in the domain of natural language processing. Serving as the foundational supervised machine learning algorithm for classification tasks, it also maintains a close affinity with neural networks. In essence, a neural network can be conceptualized as a hierarchical arrangement of logistic regression classifiers superimposed upon one another. Consequently, the classification and machine learning methodologies introduced herein assume a pivotal role, permeating the discourse of this scholarly work. Logistic regression is adaptable to the task of binary classification (such as distinguishing 'positive sentiment' from 'negative sentiment') as well as multi-class classification, although the former is initially expounded due to its computational simplicity, with subsequent discussion extending to the utilization of multinomial logistic regression [7].

2.4 Random Forest Classifier

Random Forest algorithm is composed of a predetermined number of binary decision trees, each constructed using a bootstrap sample drawn from the training dataset. In the context of a feature vector with M features, the growth of an individual tree entails the random selection of a subset of f (where $f < M$) features at each node. Subsequently, one feature from this subset is chosen for node splitting. An enhanced version of the random forest introduces an iterative augmentation of the number of trees, with each iteration termed a "construction pass." This iterative process begins with an initial number of trees and progressively expands the ensemble through consecutive construction passes. During each pass, new trees are added to the existing ensemble, contributing to the diversity and robustness of the overall model. This iterative approach allows the random forest to adapt and improve over multiple passes, capturing complex relationships within the data and enhancing the predictive performance of the algorithm. [8].

2.5 Extreme Gradient Boosting (XG Boost)

Extreme Gradient Boosting is applied in the domain of supervised learning, focusing on scenarios where training data, denoted as xi and consisting of multiple features, is utilized to predict a corresponding target variable yi. In the domain of supervised learning, the term "model" typically denotes the mathematical framework governing the derivation of predictions yi from input variables xi. A commonplace instance is the linear model, where the prediction is formulated as $\mathrm{y}i = \sum j \theta j x i j$, representing a linear combination of input features weighted by coefficients θ. The interpretability of the prediction value varies depending on the nature of the task, be it regression or classification. For instance, in logistic regression, the prediction may undergo a logistic transformation to yield the probability of belonging to the positive class, or it may serve as a ranking score for output prioritization. The parameters constitute the indeterminate components necessitating extraction from the data; in the context of linear regression, these parameters are denoted as coefficients θ [9].

2.6 Bloom’s Taxonomy

Bloom's Taxonomy, originally conceived by Benjamin Bloom in 1956, serves as a foundational framework within the realm of education for categorizing educational objectives and competencies that educators seek to impart to their students. It is an essential pedagogical tool for defining and structuring desired learning outcomes. These levels include Remembering, Understanding, Applying, Analyzing, and Creating, each with its own set of cognitive processes. This taxonomy provides a systematic structure that educators can leverage to articulate their intended learning outcomes, develop pedagogical strategies, and design assessments within academic courses.

Within the academic sphere, Bloom's Taxonomy plays a pivotal role in aligning educational objectives with instructional methodologies and assessment practices. It facilitates a comprehensive and progressive approach to cognitive development by encouraging students to move through increasingly complex cognitive tasks. In essence, it equips educators with a versatile tool to enhance the quality of education by promoting higher order thinking skills, fostering critical thinking, and ultimately contributing to the holistic development of students. Then, Bloom's Taxonomy continues to be a fundamental and indispensable asset in shaping the educational landscape and promoting effective learning outcomes [10].

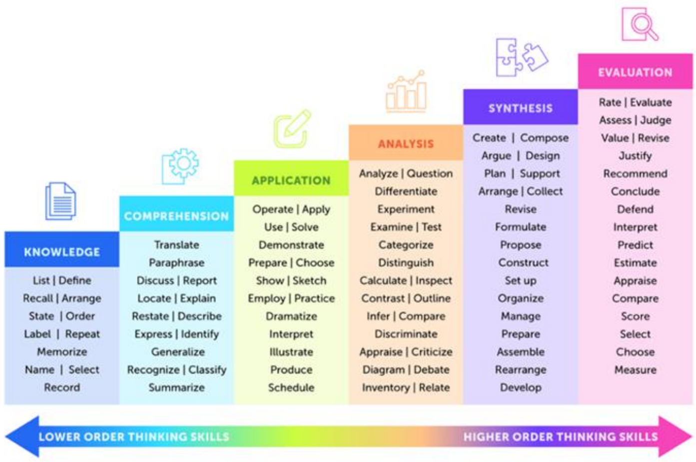


Figure 1. List of action verbs for learning outcomes [11]

2.7 Guidance on Learning Outcomes of ACM Curricula Recommendation

In accordance with the ACM Curricula Recommendations, it is discerned that the learning outcomes are not uniformly proportionate in size and do not exhibit a consistent correspondence with curriculum hours. It is evident that topics with equivalent hour allocations may exhibit considerable variation in the number of associated learning outcomes. Each of these learning outcomes is accompanied by a specified level of mastery. In the delineation of these mastery levels, we have drawn inspiration from various curriculum frameworks, with a particular emphasis on Bloom's Taxonomy, which has been extensively explored within the realm of computer science. However, it should be noted that we have not directly transposed Bloom's levels into our framework, primarily due to the contextual and pedagogical nuances associated with them. This avoidance of direct transposition is intended to mitigate the introduction of excessive heterogeneity into a document of this nature. Furthermore, we aim for these mastery levels to serve as indicative rather than imposing theoretical constraints upon the document's users.

Our framework incorporates three distinct levels of mastery, characterized as follows:

- Familiarity Level: This level involves a basic understanding of a concept without a deep knowledge of its practical application. It addresses the question: "What is your knowledge about this?"
- Usage Level: At this stage, a student can effectively apply a concept in practical situations, demonstrating skills such as utilizing it in a program or employing a specific analysis technique. It answers the question: "What can you do with this knowledge?"
- Assessment Level: The highest mastery level involves evaluating a concept from various perspectives and justifying the choice of a specific approach in problem-solving. It goes beyond mere application, aiming to answer: "Why have you chosen to employ this method?" [12].

The guidance on learning outcomes provided by the ACM/IEEE curriculum recommendation represents a thorough integration of Bloom's Taxonomy. This guidance effectively encapsulates the cognitive learning framework across all levels, addressing the domains of remembering, understanding, applying, analyzing, evaluating, and creating. accordingly, it extensively encompasses all aspects of cognitive learning frameworks.

3 Literature Reviews

There are several researchers try to use Bloom's taxonomy to match skills and the course learning outcomes for instance include Sarang Shaikh et al. [13] propose an LSTM-based deep learning model for automatic classification of course learning outcomes (CLOs) and assessment question items into different levels of Bloom's taxonomy in the cognitive domain by applying deep learning model utilizes LSTM (Long Short Term Memory) and pre-trained word embeddings for classification of course learning outcomes (CLOs) and assess items into different levels of Bloom's taxonomy in the cognitive domain.

Yuheng Li et al. [14] develop classifiers using five conventional machine learning approaches, which are Naïve Bayes, logistic regression, support vector machine, random forest and XG Boost and deep learning approach based on pre-trained language model, which is BERT to determine the cognitive levels of learning objectives automatically. The paper also highlights the importance of separating the characterization of different cognitive levels in Bloom's taxonomy, as binary classifiers focusing on a single category achieved better performance than multi-class multi-label classifiers.

Abdul Waheed et al. [15] propose the model to utilize linguistic and semantic information to classify Course Learning Outcomes (CLOs) according to the different cognitive levels of Bloom's Taxonomy by utilizing the transformer-based model named BloomNet that compared with various baselines to evaluate its performance and generalization capability along with the Part-Of-Speech (POS) and Named Entity Recognition (NER) modules to explicitly encapsulate linguistic information together with DistilRoBERTa to maintain linguistic information in the modeling of BloomNet.

Selina Banda et al. [16] investigate and demonstrate how Bloom's Taxonomy can be applied to categorize the development of cognitive processes in college education. Moreover, the study explores the implications of using Bloom's Taxonomy in educational practice, including curriculum design, teaching methodologies, and assessment strategies. The research underscores the significance of integrating Bloom's Taxonomy into course design and learning activities to enhance teaching effectiveness and facilitate student learning.

4 Research Methodology

4.1 Research framework

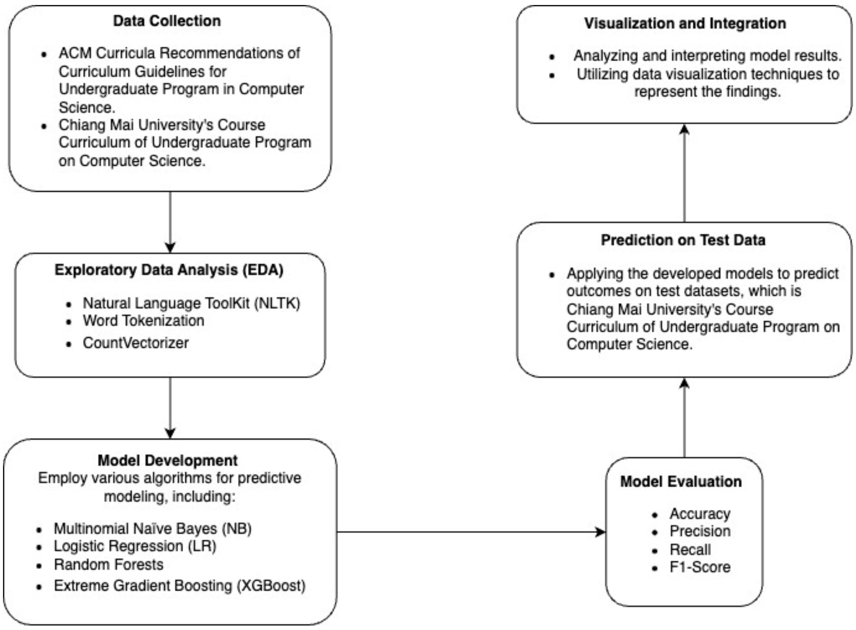


Figure 2. A Conceptual Framework.

This study aims to introduce a classification model for learning outcomes aligned with the ACM/IEEE undergraduate computer science program curriculum guidelines.

it also showcases a semi-automated prototype that visualizes the results of learning outcomes classification based on the proposed model. The study uses two datasets including ACM/IEEE Curricula Recommendations of Curriculum Guidelines for Undergraduate Program in Computer Science available via the IEEE website for training data and Chiang Mai University’s Course Curriculum of Undergraduate Program in Computer Science for testing data to predict the course learning outcomes classification in three levels, which Familiarity, Usage and Assessment, and interprets and visualizes the research findings.

4.2 Data

The study partitioned the datasets into two primary groups: the training data comprised ACM/IEEE Curricula Recommendations of Curriculum Guidelines for Undergraduate Program in Computer Science, which consists of 162 courses and 1108 CLOs. In contrast, the testing data consisted of Chiang Mai University's Course Curriculum for Undergraduate Program in Computer Science, comprising 63 courses and 267 CLOs.

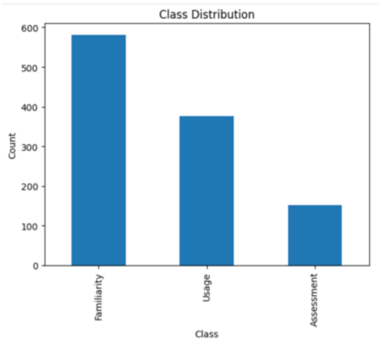


Figure 3. Class Distribution Overview of ACM/IEEE Curricula Recommendations of Curriculum Guidelines for Undergraduate Program in Computer Science.

The selected dataset in ACM/IEEE Curricula Recommendations of Curriculum Guidelines for Undergraduate Program in Computer Science comprises 162 courses, 1108 Course Learning Outcomes (CLOs) and class of each CLOs. At the same time, the selected dataset in Chiang Mai University's Course Curriculum for Undergraduate Program in Computer Science is composed of 63 courses and 267 CLOs. Within this dataset, there are 581 instances related to familiarity, 376 instances related to usage, and 151 instances related to assessment.

4.3 Exploratory Data Analysis (EDA)

This study focuses on Natural Language Processing (NLP), which is crucial for comprehending and preprocessing data before model analysis or prediction. To demonstrate the model, the dataset needs to be cleaned using the Natural Language Toolkit (NLTK), which is an open-source Python library, used for Natural Language Processing (NLP) tasks. It offers easy-to-use interfaces to over 50 corpora (collections of structured texts) and lexical resources like WordNet. Additionally, it provides a suite of text processing libraries for tasks such as classification, tokenization, stemming, tagging, parsing, and semantic reasoning [17], Word Tokenization, which is the process of splitting a text into individual words or tokens [18] and CountVectorizer, which is suitable for text processing for convert text to numerical data [19].

The initial step in text preprocessing involves utilizing NLTK's `word_tokenize` function to tokenize the Course Learning Outcomes (CLOs). Subsequently, the text data is converted into numerical format through the initialization of CountVectorizer, tailored to fit the cleaned text data. Following this, any leading digits followed by whitespace in the CLOs are removed. Additionally, periods (.) and commas (,) are replaced with an empty string. To ensure consistency in text formatting, leading whitespace is eliminated using the `str.lstrip()` method. Finally, all text is converted to lowercase using the `str.lower()` method, ensuring uniform treatment of uppercase and lowercase characters. These preprocessing steps collectively aim to cleanse and standardize the text data, making it suitable for further analysis or modeling.

4.4 Model Development

The study utilizes machine learning models for course learning outcomes (CLOs) classification like Multinomial Naïve Bayes (NB), Logistic Regression (LR), Random Forests (RF) and Extreme Gradient Boosting (XG Boost). These traditional models aim to classify the course learning outcomes (CLOs) into three classes: familiarity, usage and assessment based on Bloom's Taxonomy. By configuration parameter of the dataset to test_size = 0.2, which indicates that 20% of the data will be used for testing and random_state = 42 to set the seed for randomization, ensuring reproducibility. The dataset is divided into two parts including training data, which is assigned to feature variable X, which is course learning outcomes and testing data, which is assigned to target variable Y, which is class.

In the present study, we employ a suite of diverse classifiers to discern optimal performance in the context of a multi-class classification task. The suitable models include Multinomial Naïve Bayes with default parameters, Logistic Regression classifier with parameter max_iter = 1000 to set the maximum number of iterations for the optimization algorithm to ensure that the optimization process stops after reaching 1000 iterations, Random Forest classifier with a specific parameter that n_estimators=100 which means that the Random Forest will consist of 100 decision trees to improve the performance up to a certain point at the cost of increased computational complexity and XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable by sets the parameters as objective="multi:softmax" for indicates the perform multi-class classification to selects the class with the highest probability and num_class=len(set(y_train)) to sets the number of classes in the classification task. The adjusted parameters within the chosen models can be displayed as the following table.

Model	Parameter(s)
Multinomial Naïve Bayes (NB)	Default
Logistic Regression classifier	max_iter = 1000
Random Forest classifier	n_estimators=100
Extreme Gradient Boosting (XG Boost)	objective="multi:softmax", num_class=len(set(y_train))

Table 1. Model Parameter Adjustments

4.5 Model Evaluation

Model evaluation is a crucial component of this study, providing insights into classification models' performance and effectiveness in addressing specific problems. This assessment employs various commonly used metrics, including accuracy, precision, recall, and F1-score, to comprehensively gauge the models' performance. The study aims to robustly represent the models' effectiveness in tackling the designated tasks through these metrics. Accuracy is a metric to measure the percentage of accurately classified instances relative to the total instances, thereby offering a comprehensive evaluation of the model's efficacy across all classes. Precision indicates the model's capacity to correctly identify relevant instances while minimizing false positives which is important in scenarios where false positives have significant consequences. Recall indicates the model's performance in capturing all relevant instances within a dataset while decreasing false negatives which is important in situations where missing positive instances is costly. Lastly, the F1-score, a harmonic mean of precision and recall, offers an evaluation of a model's performance by considering both precision and recall equally which is important in the situation when the uneven class distribution.

4.6 Prediction on Test Data

This research utilizes various machine learning models, such as Multinomial Naïve Bayes, Logistic Regression classifier, Random Forest classifier, and Extreme Gradient Boosting, to predict course learning outcomes (CLOs) classification on test datasets. The models are trained on the ACM/IEEE Curricula Recommendations of Curriculum Guidelines for Undergraduate Program in Computer Science and evaluated on a separate test subset. Specifically, the study focuses on Chiang Mai University's Course Curriculum of Undergraduate Program in Computer Science.

4.7 Visualization and Data Integration

The study represents the visualization and integration aspect, aiming to analyze, interpret, and utilize data visualization techniques to present the research findings effectively. Especially, this study employs radar charts to illustrate each classification of course learning outcomes (CLOs), namely familiarity, usage, and assessment. These charts help to provide a comprehensive overview of the results and facilitate understanding better and interpretation of the findings.

5 Result

This study employs machine learning models, specifically Multinomial Naïve Bayes (NB), Random Forest (RF), Logistic Regression (LR) and Extreme Gradient Boosting (XG Boost) to classify course learning outcomes (CLOs).

Model	Accuracy	Precision	Recall	F1-Score
NB	0.76	0.75	0.76	0.73
LR	0.81	0.81	0.81	0.79
RF	0.78	0.79	0.78	0.76
XG Boost	**0.81**	**0.81**	**0.81**	**0.80**

Table 2. Performance Comparisons of Machine Learning Models.

Among the evaluated models, Extreme Gradient Boosting (XG Boost) demonstrates the highest accuracy (0.81) alongside optimal precision (0.81), recall (0.81), and F1-Score (0.80). Logistic Regression (LR) matches XGBoost in accuracy (0.81) and achieves the same values in precision (0.81) and recall (0.81), but its F1-Score is marginally lower at 0.79. Conversely, Multinomial Naïve Bayes (NB) exhibits slightly lower performance with an accuracy of 0.76, precision of 0.75, recall of 0.76, and F1-Score of 0.73.

A comprehensive analysis of classification performance regarding Course Learning Outcomes (CLOs) presents that XG Boost model yields the most promising results, correctly predicting 81 percent of instances. This performance is further validated by its optimal precision, indicating that 81 percent of the predicted CLOs are accurate, and recall, showing that the model correctly identifies 81 percent of the actual CLO instances, with an F1-Score that balances precision and recall at 0.80. In comparison, Logistic Regression, Random Forest, and Multinomial Naïve Bayes demonstrate comparatively lower performance levels.

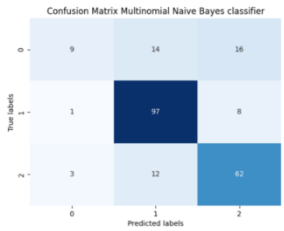


Figure 4. The Confusion Matrix of Multinomial Naïve Bayes Classifier.

Figure 5. The Confusion Matrix of Logistic Regression Classifier.

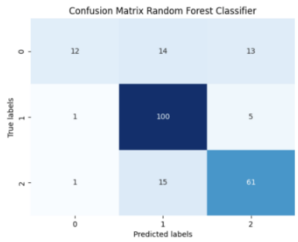


Figure 6. The Confusion Matrix of Random Forest Classifier.

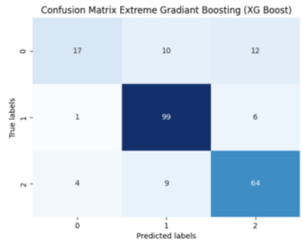


Figure 7. The Confusion Matrix of Extreme Gradient Boosting Classifier.

The visualizations of the confusion matrix in the figures above represent the performance of classification algorithms across multiple classes. Specifically, it pertains to a three-class classification task, where the classes are denoted as 0 (Assessment), 1 (Familiarity), and 2 (Usage). The figures above depict the number of data points correctly classified as well as those misclassified and elucidate the model's accuracy and areas of difficulty in classifying specific classes. For Multinomial Naïve Bayes classifier (NB), illustrates that out of the data points analyzed, 9 were classified as class 0 (Assessment), 97 as class 1 (Familiarity), and 62 as class 3 (Usage). In the case of Logistic Regression classifier (LR), it is observed that 16 data points were classified as class 0 (Assessment), 98 as class 1 (Familiarity), and 65 as class 3 (Usage). Moving on to Random Forest classifier (LR), it is evident that 12 data points were categorized as class 0 (Assessment), 100 as class 1 (Familiarity), and 61 as class 3 (Usage). Lastly, for Extreme Gradient Boosting Classifier (XG Boost), revealed that among the analyzed data points, 17 were assigned to class 0 (Assessment), 99 to class 1 (Familiarity), and 64 to class 3 (Usage).

To achieve these goals, the study classifies the course learning outcomes (CLOs) based on two features: course and course learning outcomes. The summarized findings of this examination are presented in the following approaches.

subject	outcomes	NB	LR	RF	XG Boost
Computer Science Technology	students are able to explain principle and limitation of data allocation in computer and principle big data manipulation	Familiarity	Familiarity	Familiarity	Familiarity
	students are able to describe principle of technology in computer controller and communication	Familiarity	Familiarity	Familiarity	Familiarity
	students are able to analyze cybersecurity	Usage	Usage	Usage	Usage
	students are able to analyze big data and design basic analysis model	Usage	Usage	Usage	Assessment
	students are able to work as a team for carrying out a small project by applying computer science technologies	Usage	Usage	Usage	Usage
	students are able to present their project both in oral form and in written form	Usage	Usage	Usage	Usage
	students demonstrate professional ethics have discipline punctuality as well as self and social responsibility	Familiarity	Familiarity	Usage	Usage

Table 3. The Example of Course Learning Outcomes Classification Results Based on Proposed Models.

The table illustrates the example of course learning outcomes classification using the proposed models, indicating the class of each CLO as Familiarity, Usage, or Assessment. In terms of model evaluation, the findings indicate that both the Extreme Gradient Boosting (XG Boost) model and the Logistic Regression Classifier (LR) exhibited the highest performance. However, the Random Forest Classifier and Multinomial Naïve Bayes models showed moderate performance.

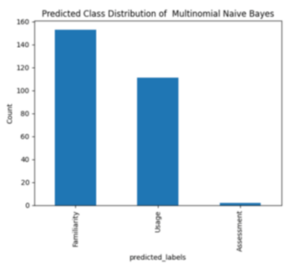


Figure 8. Predicted Class Distribution of Multinomial Naïve Bayes (NB)

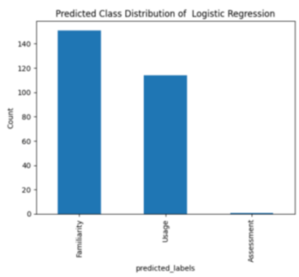


Figure 9. Predicted Class Distribution of Logistic Regression classifier (LR)

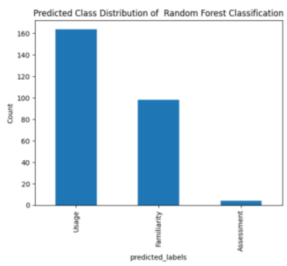


Figure 10. Predicted Class Distribution of Random Forest Classifier (RF)

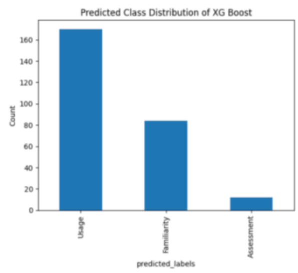
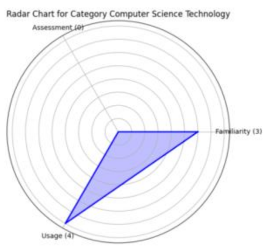
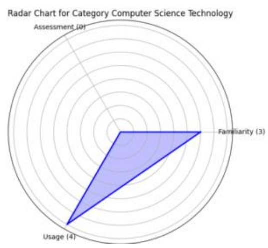


Figure 11. Predicted Class Distribution of Extreme Gradient Boosting (XG Boost)

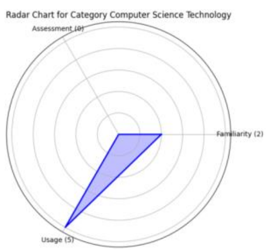
The bar chart analyses illustrate the distribution of predicted classes among the proposed models. The results indicate that Multinomial Naïve Bayes (NB) and Logistic Regression classifier (LR) models predominantly classify course learning outcomes into Familiarity and Usage classes, respectively. Conversely, the Random Forest Classifier (RF) and Extreme Gradient Boosting (EX Boost) models exhibit a higher prevalence of course learning outcomes in the Usage and Familiarity classes, respectively. Furthermore, it is noted that Assessment constitutes the least represented category among course learning outcomes.



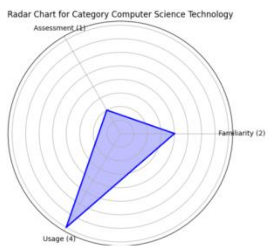
Multinomial Naïve Bayes



Logistic Regression



Random Forest



Extreme Gradient Boosting

Figure 12. Radar Chart of Computer Science Technology Course.

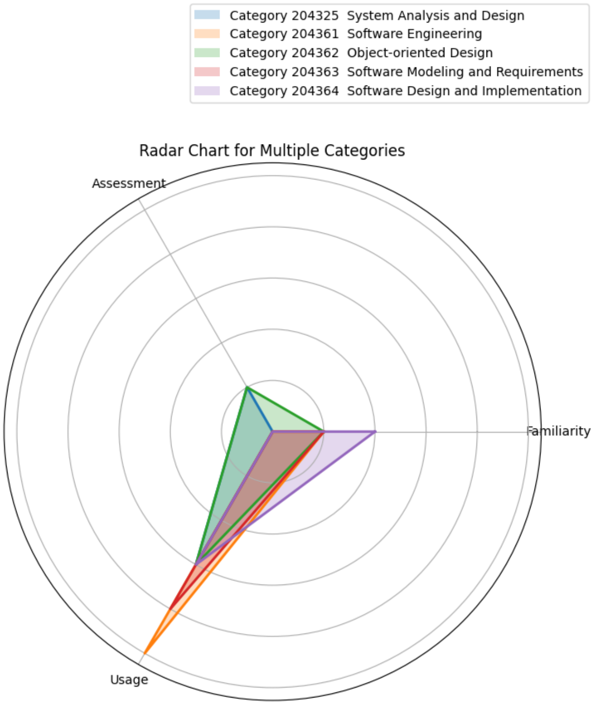


Figure 13. Radar Chart of Key Skills in Course of Computer Science

The radar chart above represents design courses in the computer science program, including 204325 (System Analysis and Design), 204361 (Software Engineering), 204362 (Object-oriented Design), 204363 (Software Modeling and Requirements), and 204364 (Software Design and Implementation). These courses extremely focus on the usage dimension, emphasizing the familiarity and assessment dimensions less. These insights suggest that the design courses are primarily geared towards practical application and hands-on experience, equipping students with the skills to utilize their knowledge effectively in real-world situations.

This study visualizes the courses in computer science technology, on radar charts which calculate the weight of Familiarity class, Usage class and Assessment class. This analysis offers several benefits for instance educators can assess each course effectively and promote the desired learning outcomes, ensuring alignment with the overarching objectives of educational institutions or organizations. These visualizations also encourage benchmarking by comparing each course within the proposed program, allowing educators to assemble insights from one another and adopt effective strategies for curriculum design and implementation. Moreover, radar charts efficiently discover gaps and overlaps in the course curriculum, enabling more effective course planning and ensuring thorough coverage of essential topics. Furthermore, educators can evaluate the alignment of their curriculum with industry-recognized benchmarks and allow the adoption of best practices outlined by ACM/IEEE to enhance the quality and relevance of the proposed course curriculum.

6 Conclusion

The study develops classification models for learning outcomes aligned with ACM/IEEE undergraduate computer science program guidelines. Using a semi-automated prototype, it visualizes classification results, focusing on course outcomes at Chiang Mai University. Trained on ACM/IEEE dataset, models classify outcomes into Familiarity, Usage, and Assessment. Visualizations highlight model differences and course alignment by considering the weight of each class. Overall, the study contributes structured evaluation methods for program and course outcomes, aiding educators. It also addresses ambiguity in Bloom's Taxonomy verbs, paving the way for further research in educational assessment and curriculum design.

7 Discussion

Despite the significant contributions and findings of this research, several limitations must be acknowledged. Constraints related to data availability and quality restricted access to comprehensive and high-quality datasets, potentially impacting the robustness of the research findings. Additionally, the class imbalance in the distribution of Course Learning Outcomes (CLOs) affects the fairness and accuracy of the classification prediction results, leading to unsatisfactory scores in model evaluation due to insufficient information. Moreover, the precision of the classification could not be verified by domain experts because of biases and gaps in their responses. To address the issue of class imbalance, resampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and random under sampling can be employed. Another limitation pertains to the classification of CLOs in the ACM/IEEE Curricula Recommendations, which encompass three categories: Familiarity, Usage, and Assessment, in contrast to Bloom's Taxonomy, which includes six levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. The proposed classification models can be adapted to incorporate Bloom’s Taxonomy, thereby addressing this limitation.

According to the study, the aim is to classify Course Learning Outcomes (CLOs) to enable educators and lecturers to explore the multifaceted aspects of educational outcomes, thereby enhancing their understanding. Future research could benefit from integrating predictive analytics models with CLO classification frameworks to identify early indicators of student success and devise strategies for proactive student support initiatives and retention efforts. Another promising avenue for research involves applying CLO classification models to other programs, such as undergraduate computer engineering and software engineering, by collecting data from various sources and verifying results with domain experts or alternative methods. Furthermore, to achieve a clearer understanding of classification predictions, it is advantageous to classify CLOs using the six levels of Bloom's Taxonomy rather than relying solely on the ACM/IEEE curriculum recommendations, as Bloom's Taxonomy provides a more comprehensive and lucid framework for classification prediction.

References

1. D. MazzoccaOlr, Outcome-based education outcome based, https://cga.ct.gov/PS94/rpt%5Colr%5Chtm/94-R-0290.htm (accessed Feb. 10, 2024).
2. K. F. F. Jiechieu and N. Tsopze, "Skills prediction based on multi-label resume classification using CNN with model predictions explanation," *Neural Comput Appl*, vol. 33, no. 10, pp. 5069–5087, May 2021, doi: 10.1007/s00521-020-05302-x.
3. J. J. Heckman, "Policies to foster human capital," *Research in Economics*, vol. 54, no. 1, pp. 3–56, 2000, doi: 10.1006/reec.1999.0225.
4. Association for Computing Machinery (ACM) IEEE Computer Society, *Computing Curricula 2020*. ACM, 2020. doi: 10.1145/3467967.[5] H. Dalianis, "Evaluation Metrics and Evaluation," in *Clinical Text Mining*, Springer International Publishing, 2018, pp. 45–53. doi: 10.1007/978-3-319-78503-5_6.
5. I. N. Sodhar, A. H. Jalbani, A. H. Buller, A. A. Mirani, and A. N. Sodhar, *Advances in Computer Science*. AkiNik Publications, 2020. doi: 10.22271/ed.book.784.
6. H. joon Kim, J. Kim, J. Kim, and P. Lim, "Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning," *Neurocomputing*, vol. 315, pp. 128–134, Nov. 2018, doi: 10.1016/j.neucom.2018.07.002.
7. J. Daniel and J. H. Martin, "Speech and Language Processing," 2023.
8. A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved Random Forest for Classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, Aug. 2018, doi: 10.1109/TIP.2018.2834830.
9. "xgboost Release 2.1.0-dev xgboost developers," 2024. Accessed: Jan. 06, 2024. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/xgboost/latest/xgboost.pdf
10. J. Shabatura, "Using Bloom's Taxonomy to Write Effective Learning Outcomes | Teaching Innovation and Pedagogical Support," Assignments & Measuring Student Learning. Accessed: Jan. 06, 2024. [Online]. Available: https://tips.uark.edu/using-blooms-taxonomy/
11. I. Andreev, "Bloom's Taxonomy: Revised Levels, Verbs for Objectives [2024]," valamis. Accessed: Jan. 06, 2024. [Online]. Available: https://www.valamis.com/hub/blooms-taxonomy
12. Association for Computing Machinery (ACM) IEEE Computer Society, *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc, 2013. doi: 10.1145/2534860.

13. Shaikh, S., Daudpotta, S. M., & Imran, A. S. (2021). Bloom's Learning Outcomes' Automatic Classification Using LSTM and Pretrained Word Embeddings. *IEEE Access*, *9*, 117887–117909. https://doi.org/10.1109/ACCESS.2021.3106443.
14. Li, Y., Rakovic, M., Poh, B. X., Gaševic, D., & Chen, G. (2022). Automatic Classification of Learning Objectives Based on Bloom's Taxonomy. *Proceedings of the 15th International Conference on Educational Data Mining, EDM 2022*. https://doi.org/10.5281/zenodo.6853191.
15. Waheed, A., Goyal, M., Mittal, N., Gupta, D., Khanna, A., & Sharma, M. (2021). *BloomNet: A Robust Transformer based model for Bloom's Learning Outcome Classification*. http://arxiv.org/abs/2108.07249
16. Banda, S., Phiri, F., Kaale, J., & Banda, A. (2023). Application of Bloom's Taxonomy in Categorization of Cognitive Process Development in Colleges. *Journal of Education and Practice*. https://doi.org/10.7176/JEP/14-4-02
17. Mohan, V. (2016). Text Mining: open Source Tokenization Tools – An Analysis. Advanced Computational Intelligence: An International Journal (ACII), 3(1), 37–47. https://doi.org/10.5121/acii.2016.3104
18. Tutorialspoint, "Python - Word Tokenization," www.tutorialspoint.com. https://www.tutorialspoint.com/python_data_science/python_word_tokenization.htm (accessed Jan. 15, 2024).
19. P. Jain, "Basics of CountVectorizer," Medium, Jun. 01, 2021. https://towardsdatascience.com/basics-of-countvectorizer-e26677900f9c (accessed Dec. 16, 2023).