23rd SEAAIR Conference "Best Paper" Citation

Development of a Forecasting Model of Teaching Effectiveness

Mateo Borbon, Jr.[1], Jeffrie Atendido[2], Adlin Mae Dimasuay [3]

([1]mateo.borbonjr@benilde.edu.ph, [2]jeffrie.atendido@benilde.edu.ph, [3]adlinmae.dimasuay@benilde.edu.ph)
De La Salle-College of Saint Benilde, Manila, Philippines

Abstract

This research project aims to utilize Python programming and machine learning algorithms to design a predictive model for assessing faculty effectiveness. The model considers various factors such as teaching effectiveness, course management, course materials, class openness, and course management. By analyzing these factors and testing the various model's performance against standard metrics, the collected data is processed and analyzed using regression analysis and decision trees, enabling the development of a predictive model. This model may provide estimates of future performance, allowing for the identification of high-performing faculty members, areas for improvement, and optimal resource allocation.

The study results demonstrate that Naive Bayes, Random Forest, and Decision Tree algorithms are particularly effective in predicting faculty performance based on the provided data. These findings promise to inform the development of strategies and policies that enhance faculty effectiveness and contribute to institutional excellence. By employing a data-driven approach, this study offers valuable insights into the utility of different machine learning algorithms and their predictive capabilities in assessing faculty performance within the context of higher education.

Keywords: faculty effectiveness, predictive model, data-driven approach, machine learning

Introduction

Faculty evaluation systematically assesses a faculty member's performance, teaching effectiveness, research contributions, and overall professional competence. Faculty evaluation aims to ensure accountability, enhance teaching quality, promote faculty development, and maintain high academic standards within educational institutions. One of the key advantages of faculty evaluation is that it provides valuable feedback to faculty members, helping them identify areas for improvement and refine their teaching methodologies (Ching, 2019). It also allows institutions to recognize and reward exceptional faculty members, promoting a culture of excellence. Additionally, faculty evaluation can contribute to the overall enhancement of student learning experiences by fostering a supportive and engaging academic environment.

Research Background and Review of the Literature

Faculty evaluation plays a crucial role in identifying and addressing poor performance among faculty members. By systematically assessing their teaching effectiveness, research contributions, and professional competence, institutions can pinpoint areas of weakness and take appropriate corrective actions. Through the evaluation process, faculty members who consistently demonstrate subpar performance can be identified, allowing institutions to implement targeted interventions, such as mentoring programs, professional development opportunities, or performance improvement plans. This proactive approach ensures faculty members receive the necessary support and guidance to enhance their skills and meet the institution's expectations, ultimately promoting overall academic quality and student success.

However, this exercise is not without limitations. One of the challenges is ensuring the use of fair and unbiased evaluation criteria and processes. Subjective judgments, potential biases, and inconsistency among evaluators can undermine the reliability and validity of the evaluation results. Another concern is the potential emphasis on quantitative metrics, which may overlook qualitative aspects of teaching and fail to capture the full scope of faculty contributions (Theall & Franklin, 2010). Another limitation is the time in the evaluation process, which usually takes months or even years to complete. The evaluation process and interpretation become more complicated for a large institution with a large faculty roster. Automating these procedures is a welcome innovation, and the adoption of an automated prediction and/or forecasting is a big help (Munford, 2021; Liu et al., 2020; Martin et al.,2019)

Predictive analysis is the process that involves data analysis, machine learning, artificial intelligence, and statistical models to find patterns that might predict future behavior and outcomes (Google, n.d.). Teacher evaluation is a study of great interest where numerous efforts converge to establish models from the association of heterogeneous data from academic actors (Ordoñez-Avila et al., 2023). Machine Learning is one sector generating exciting undertakings regarding teacher evaluations (e.g., Lin, 2021; Xia & Yan, 2021).

Data Mining and Machine Learning are similar since both gather an extensively enormous amount of data (also known as Knowledge Discovery Databases or KDD) from one or more sources for analysis to discover hidden knowledge, new trends and patterns to make predictions (Vijayalakshmi et al., 2020; Yağcı, 2022), however, for machine learning, it learns from its

previously executed tasks by analyzing and predicting and improving based on the gathered data (Ray, 2019).

Machine learning algorithms used in predictive analysis utilize regression, classification, clustering, control, time series, neural networks, and decision tree techniques in choosing the appropriate predictive model. Machine learning algorithm is divided into four (4) types: (1) supervised learning, such as classification and regression; (2) unsupervised learning, such as clustering and association; (3) semi-supervised learning, such as clustering and classification. And lastly, (4) reinforced learning such as control and classification. There are numerous examples of machine learning algorithms, and among the popular ones are: (1) Naive-Bayes, (2) K-Nearest Neighbors, (3) Random Forest, (4) Support Vector Machine, and (5) Decision Tree (Kathiroli & Vijayalakshmi., 2020; Ray, 2019). This research will only focus on these five (5) algorithms.

The Naive Bayes algorithm is a simple yet powerful classification algorithm based on the 20 principles of Bayes' theorem and conditional probability. It assumes that the features in a dataset are independent, hence the term "naive." Naive Bayes calculates the probability of a given instance belonging to a specific class by considering the probabilities of its features. It uses training data to estimate these probabilities and builds a probabilistic model. When classifying new instances, the algorithm calculates the likelihood of each class based on the observed features and selects the class with the highest probability. Despite its feature independence assumption, Naive Bayes often performs remarkably well in various real-world applications, particularly in text classification, spam filtering, and sentiment analysis, where it has demonstrated efficiency and scalability.

This classification algorithm can be applied to faculty evaluation. It uses conditional probability and assumes feature independence to predict the likelihood of a faculty member's performance based on various indicators. By analyzing training data, Naive Bayes estimates the probabilities of different performance levels and builds a predictive model. This algorithm is often used in tasks such as classifying faculty performance levels or identifying factors contributing to teaching effectiveness. Naive Bayes provides a straightforward and efficient approach to evaluating faculty members' performance and can be a valuable tool in the assessment process (Kumar et al., 2018; Lalata et al., 2019; Pacol & Palaoag, 2021).

The K-Nearest Neighbors (KNN) algorithm is a classification algorithm that stores and generates new data points based on a defined similarity measure. Unlike the Naive Bayes algorithm, KNN is considered non-parametric as it makes no assumptions about the data or its distribution. This characteristic allows KNN to be flexible and adaptable to different datasets. In a recent study by Yağcı (2022), the utility of KNN in the research was evident.

The Decision Tree is a non-parametric approach employed in supervised learning for classification and regression tasks. It is capable of handling output variables that are either continuous or categorical. The classification process of a decision tree consists of two steps: learning and prediction. In the learning stage, the model is trained using the provided training data, while in the prediction step, the trained model is used to predict responses for new, unseen

data. This allows decision trees to make accurate predictions based on the learned patterns (Navlani, 2018).

In the Random Forest technique, the final prediction in a model is derived by aggregating the results from multiple decision trees. Random Forest is a supervised learning approach that utilizes an ensemble of decision trees to generate more accurate predictions than other algorithms. It effectively addresses the problem of overfitting and can be applied to both linear and non-linear models. Each decision tree within the Random Forest contributes to the overall classification process. By training different models and employing multiple decision trees, a variety of outputs are produced. Through careful analysis of these results, a final output is generated. The decision-making process for each sample is guided by constructing a decision tree (Reinstein, 2017).

Support Vector Machine (SVM) is a machine learning algorithm initially introduced by Cortes and Vapnik in 1995. SVM finds applications in both regression and classification tasks. The SVM classifier aims to separate different example classes while maximizing the distance between the nearest cleanly separated examples. This is achieved by constructing a maximum-margin hyperplane in a transformed input space. The support vectors are the data points located on the boundaries, and the optimal hyperplane is determined as the center of the margin. The parameters of the solution hyperplane are obtained through a quadratic programming optimization problem. Further insights into SVM and its optimization methods can be found in the work of Shmilovici (2009).

Machine learning transforms faculty evaluation and teaching assignments by leveraging large datasets and advanced algorithms. These models analyze performance indicators and patterns, enabling data-driven assessments and identification of areas for improvement. Machine learning also optimizes teaching assignments by considering faculty expertise, course requirements, and student preferences. However, challenges include training data quality, decision-making fairness, and ethical considerations (Lalata et al., 2019).

The Faculty Assessment Scale is a systematic evaluation instrument employed within educational organizations to assess the performance and efficiency of faculty members. Typically, it encompasses a range of criteria or facets against which faculty members' teaching, research, service, and broader contributions to the institution are gauged. These criteria are frequently assessed using a rating scale with descriptors ranging from "Poor" to "Outstanding," which quantifies and communicates faculty performance, assisting in decisions about tenure, advancement, and career growth captured in the organization's different manuals and policies.

In the case of XYZ College, the faculty assessment scale's existence can be traced back to its establishment in 1998, wherein academic advising, Student Instructional Report (SIR), and Peer Evaluation Form (PEF) served as the basis for faculty assessment. To address whether evaluating teaching performance in the institution meets the standards and requirements of a sound evaluation, a study was conducted in 2006 following the meta-evaluation techniques by Stufflebeam (2000). The study found that the measures lacked the accuracy standard of the meta-evaluation checklist, resulting in its overhaul (Magno, 2009). Based on this evaluation and the tendency of the respondents to select the midpoints and avoid extreme responses on Likert scales

(Grandy, 1996; Wang et al., 2008; Pornel & Saldaña, 2013), a new instrument called Students' Teacher Assessment Report (STAR) was crafted and implemented in 2007 which measures the framework of *preparation and planning*, *classroom environment*, *instruction*, and *professional responsibilities* (Danielson, 1996).

It should be noted that the reference scale and its interpretation of the old faculty assessment (see Table 1) are already enshrined in the various operational manuals. The STAR utilizes a 4-point scale which throws off the assessment's computation and interpretation. Suffice it to say that various stakeholder consultations are thus made to account for this peculiarity, and institutional adjustments and solutions are implemented. A new component (*learner-centered practices*) is included in AY 2008-2009 as a quantitative measure for "effective learner-centered teaching" to conform to its reference in the faculty manual. It is fully utilized beginning in AY 2009-2010. Following technological advancements and societal upheaval in 2019 necessitates its repurposing to Learners' Assessment of Teachers and Courseware in a Hybrid Environment (LATCH), giving rise to Effectiveness Of Teachers (EOT), Online Course Management (OCM), Effectiveness Of Courseware (EOC), Promotion Of Openness (POO), and Promotion Of Deep Learning (PODL) measures.

In XYZ College, faculty teaching loads are assigned every trimester, factoring in their most recent evaluation. In other words, there is at least a month for corrective actions by the college to assign competent facilitators to a subject offered. Predicting the faculty's performance as a factor in course and subject offerings justifies the investigation of the different machine learning models. This research aims to supplement decision-making processes, resource allocation, and teaching quality by addressing the question: "Which machine learning algorithm can deliver the most accurate predictive model based on the evaluation dataset?".

Methods

Data. The data comprises 3203 teacher evaluations by students spanning four (4) years beginning in 2019. EOT is measured using thirteen (13) items, OCM with four (4) items, EOC with eight (8) items, POO with two (2) items, and PODL with three (3) items are used using XYZ College's faculty evaluation for every academic year. These evaluation values are averaged and categorized according to the equal-width discretization model, which preprocesses continuous numerical data into discrete intervals of identical width. By identifying the data range, choosing the most suitable number of bins, calculating the width of the bin, creating new bins based on the calculated bin width, assigning data points to the bin, and finally, depicting the data within each bin with a single value, the proponents came to create the performance category for interpreting the outcome.

Conforming with Table 1 necessitates the creation of a faculty efficiency index (FEI) column, which is utilized in interpreting the performance category. The decision to establish wider ranges for categories 1 and 2 in contrast to the other categories can be attributed to a multifaceted rationale to capture a nuanced spectrum of performance within these tiers. This deliberate choice accounts for various factors contributing to the varying proficiency levels or inadequacy present in the lower performance bands. The broader span allotted to categories 2 (Needs Improvement) and 1 (Poor) aligns with understanding the developmental potential, acknowledging that individuals within these ranges may exhibit diverse degrees of room for improvement. This

approach recognizes that faculty performance within these categories may encompass varying levels of subpar accomplishment, accommodating scenarios where individuals' skills might range from moderately below standard to significantly underperforming.

The broader range approach also accommodates the diversity inherent in these lower-performance tiers. In contexts involving individuals with distinct backgrounds, experiences, or learning challenges, the wider range acknowledges that disparities in performance can arise due to many factors. This inclusive perspective allows evaluators to consider a more diverse circumstance while categorizing performance. Moreover, the broader range facilitates the emphasis on improvement within these categories. This reflects an educational ethos that categorizes individuals and provides constructive feedback for growth and development (Isoré, 2009; Tufts University, n.d.).

The Panda library scales the data and removes outliers, unnecessary data points, and missing values, producing the most relevant inputs to the models. Feature selection involves reducing the number of variables used to predict the outcome to boost model interpretability, lower complexity, improve the algorithms' computing efficiency, and avoid overfitting. This process results in the inclusion of school code and program code variables in the active dataset.

Table 1: Categorization of Criteria

Category	Criteria	Interpretation
1	Between 1.0 and 1.99	Poor
2	Between 2.0 and 2.99	Needs improvement
3	Between 3.00 and 3.33	Satisfactory
4	Between 3.34 and 3.66	Very satisfactory
5	Between 3.67 and 4.00	Outstanding

Models. Predictive modeling involves developing a model by utilizing data that has known outcomes. Subsequently, this model forecasts result values for datasets without known outcomes. Various forecasting models such as Naive-Bayes, K-Nearest Neighbor, Random Forest, Support Vector Machine, and Decision Tree are employed to enhance efficiency and accuracy. These machine-learning techniques autonomously generate models correlating input data with the desired target values in supervised optimization scenarios. The model's performance is assessed using metrics derived from the confusion matrix alongside other evaluation metrics. Given the existing literature (e.g., Asif et al., 2017), it is recognized that there is no universally superior classifier for result prediction. Hence, examining and identifying the most researched classifiers suitable for the analyzed data is crucial. The Python code used in the examinations is found in Appendix A.

The research employs the Python Programming Language, specifically utilizing JupyterLab as an interactive development environment for coding and data analysis. Python is an open-source language freely available for personal and commercial use. It is versatile, running on various operating systems, and finds applications in web development, scientific computing, software development, and more. To leverage Python's capabilities in scientific computing, the researchers utilized the Pandas library, which provides powerful data processing, analysis, and

manipulation functionalities (Welcome to Python.Org, n.d.). For machine learning tasks, the researchers relied on the scikit-learn (*sklearn*) library, which offers a wide range of classification, regression, and clustering algorithms (Pedregosa et al., 2011). In addition, *numpy*, *seaborn*, and *matplotlib* libraries for data visualization are utilized. These tools were instrumental in data cleansing, splitting the data into train and test datasets (80% and 20%, respectively), loading the data into the selected machine learning algorithms, and evaluating the performance of each algorithm and its results.

Evaluation Metrics. The evaluation metrics common to these algorithms are *accuracy* (expressed as a percentage with a higher value indicating better performance), *precision* (ranges from 0 to 1 with the latter indicating perfect precision), *recall* (ranges from 0 to 1 with the latter indicating a perfect recall), and *F1 score* (the harmonic mean of precision and recall with 1 indicating the best possible score). The Receiver Operating Characteristic (ROC) curve and the Area Under the ROC Curve (AUC-ROC) are not evaluation metrics since several classifiers do not have the boundary values to generate the curve.

The Faculty Effectiveness Index (FEI) data frame is divided into train and test data. A machine learning technique was used to train the machine based on the knowledge learned from the train set. The needed attribute will be predicted for the test set using an algorithm and the information learned from the training set. The train set must be larger than the test set to ensure superior data learning (Brownlee, 2020; Galarnyk, 2022). Normally, eighty percent (80%) of the dataset comprises the train set, and twenty percent (20%) of the observations are for testing. According to Tokuç (2021), since there is no single rule of thumb in splitting the dataset into train and test sets, a 70:30 train and test ratio is used if the dataset is relatively small (n<10,000), while a 99:1 train and test ratio is used if the dataset is very large (n>=1,000,000). In this instance, the Train set consists of 1,249 records, and the Test set, which comprises the remaining 641 records, is 20% of the total.

Results

The model's performance was evaluated with a confusion matrix, accuracy, precision, recall, and f-score (F1) metrics. The confusion matrix shows the current situation in the dataset and the number of correct/incorrect predictions of the model. The number of correctly and incorrectly classified instances calculates the model's performance. In the succeeding tables, the rows show the real numbers of the samples in the test set, and the columns represent the estimation of the model. The table is a 3x3 matrix that shows the number of instances correctly predicted on the diagonal. The other numbers in the table represent the number of errors made in the predictions.

Table 2: Confusion matrix of the KNN algorithm

kNN		**Predicted**			
		Needs Improvement	Outstanding	Poor	*Sum*
Actual	Needs Improvement	97	15	0	112
	Outstanding	0	504	0	504
	Poor	0	5	20	25
	Sum	97	524	20	**641**

Table 2 displays the confusion matrix of the KNN algorithm. It shows that 20 (86.6%) of those with actual "Poor" ratings, 504 (100%) of those with "Outstanding" ratings, and 97 (80%) of those with "Needs Improvement" ratings were predicted correctly.

Table 3: Confusion matrix of the Naive Bayes algorithm

Naive Bayes		**Predicted**			
		Needs Improvement	Outstanding	Poor	*Sum*
Actual	Needs Improvement	112	0	0	112
	Outstanding	0	504	0	504
	Poor	0	0	25	25
	Sum	112	504	25	**641**

The confusion matrix for the Naive Bayes algorithm is shown in Table 3. It demonstrates that all predictions for 112 (100%) actual "Poor" ratings, 504 (100%) actual "Outstanding" ratings, and 25 (100%) actual "Needs Improvement" ratings were accurate.

Table 4. Confusion matrix of the SVM algorithm

SVM		**Predicted**			
		Needs Improvement	Outstanding	Poor	*Sum*
Actual	Needs Improvement	102	10	0	112
	Outstanding	0	504	0	504
	Poor	0	25	0	25
	Sum	102	539	0	**641**

Presented in Table 4 is the confusion matrix for the Support Vector Machine algorithm. It demonstrates that 102 (91.1%) of the actual "Poor" ratings, 504 (100%) of the actual "Outstanding" ratings, and 0 (0% of the actual "Needs Improvement" ratings) were properly forecasted.

Table 5: Confusion matrix of the Random Forest algorithm

RF		**Predicted**			
		Needs Improvement	Outstanding	Poor	*Sum*
Actual	Needs Improvement	112	0	0	112
	Outstanding	0	504	0	504
	Poor	0	0	25	25
	Sum	112	504	25	**641**

The confusion matrix shown in Table 5 is about the Random Forest algorithm. It demonstrates that 112 (100%) of those who received "Poor" evaluations, 504 (100%) of those who received "Outstanding" ratings, and 25 (100%) of those who received "Needs Improvement" ratings had their ratings accurately identified.

Table 6: Confusion matrix of the Decision Tree algorithm

DT		**Predicted**			
		Needs Improvement	Outstanding	Poor	*Sum*
Actual	Needs Improvement	112	0	0	112
	Outstanding	0	504	0	504
	Poor	0	0	25	25
	Sum	112	504	25	**641**

The confusion matrix for the Decision Tree method can be seen in Table 6. It reveals that 112 (100%) of the actual "Needs Improvement" ratings, 504 (100%) of the "Outstanding" ratings, and 25 (100%) of the "Poor" ratings were accurately determined.

Table 7: Predictive data mining models' performance evaluation

Model	Accuracy	Precision	Recall	F1 Score
KNN	0.969	0.970	0.969	0.968
Naive Bayes	1.000	1.000	1.000	1.000
SVM	0.945	0.910	0.945	0.926
Random Forest	1.000	1.000	1.000	1.000
Decision Tree	1.000	1.000	1.000	1.000

Based on the findings presented in Table 7, it was observed that the Naive Bayes, Random Forest, and Decision Tree algorithms achieved the highest accuracy value of 100%. This indicates a strong correlation between the predicted and actual data, demonstrating that all samples were correctly classified. The results highlight the effectiveness of these algorithms in accurately predicting and classifying the data under consideration.

Discussion

In this study, the primary focus was evaluating the efficiency score of faculty members at XYZ College using various machine learning algorithms. The algorithms tested included Naive Bayes, KNN, Random Forest, SVM, and Decision Tree on the faculty evaluation data collected spanning four years from 2019 onwards.

The results obtained from the confusion matrices indicated that Naive Bayes, Random Forest, and Decision Tree achieved the highest accuracy, followed by KNN and SVM. Specifically,

Naive Bayes, Random Forest, and Decision Tree algorithms exhibited 100% accuracy, precision, recall, and F1 scores, indicating a strong correlation between the predicted and actual data. KNN exhibits an accuracy of 96.9%, precision of 97%, recall of 96.9%, and an F1 score of 96.8%. At the same time, SVM had the lowest prediction performance, with an accuracy of 94.5%, precision of 91%, recall of 94.5%, and an F1 score of 92.6%. These findings align with the results of Meyer et al. (2003) and Sun et al. (2002). However, they contradict the study conducted by Yağcı (2022), which found KNN to have the lowest classification accuracy in predicting final student grades while SVM achieved high accuracy in classification tasks.

Despite the compelling findings presented in this study, several limitations warrant consideration when interpreting the results. Firstly, the high accuracy, precision, recall, and F1 scores achieved by certain algorithms, particularly Naive Bayes, Random Forest, and Decision Tree, might raise concerns regarding potential overfitting to the specific dataset. Further investigation into the generalization capacity of these models is recommended, possibly through cross-validation and testing on external datasets. Additionally, the achieved prediction performance of KNN and SVM might be contingent on the selected hyperparameters, distance metrics, and kernel functions. The study acknowledges that KNN exhibited favorable metrics but outperformed other algorithms. Thus, systematically exploring hyperparameters for KNN and SVM and assessing alternative distance metrics and kernel functions could provide insights into further optimizing their performance.

Furthermore, the faculty evaluation dataset from XYZ College, though spanning multiple years, might introduce temporal biases or institutional peculiarities that influence the algorithms' performance. Machine learning algorithms are only as good as the data they are trained on. If the data used for evaluation is biased or incomplete, it can lead to biased or unfair evaluations. The potential impact of such factors on the generalizability of the results necessitates caution when extending the findings to other academic institutions or contexts.

Several recommendations can guide future research in this domain based on the results and limitations identified in this research. Firstly, the study encourages a more robust evaluation framework that includes cross-validation techniques to assess the stability and generalization capacity of the models. This will mitigate concerns regarding overfitting and ensure that the reported performance metrics indicate the models' true predictive abilities. Additionally, for algorithms like KNN and SVM that exhibited comparatively lower prediction performance, the study suggests conducting an extensive hyperparameter search to identify the optimal configurations that may enhance their predictive accuracy. Exploring various distance metrics and kernel functions could address the observed disparities and elevate their performance to align with the other algorithms. Moreover, considering the varying results reported in related studies, the field would benefit from larger-scale comparative analyses across different institutions and datasets. This would contribute to a more comprehensive understanding of the algorithms' generalizability and effectiveness in diverse educational settings.

For educational institutions, faculty members, and the overall standard of education, developing a predictive model for evaluating faculty effectiveness can have substantial implications. A methodology like this might shed light on how well a faculty member is performing and can be a guide to professional development that contributes to better student outcomes. Predictive models

objectively evaluate faculty performance, reducing bias and subjectivity in the assessment process. These can also be used to make more informed decisions when hiring new faculty members and considering tenure and promotions. Insights from the model can inform curriculum development and ensure alignment with effective teaching methods. Also, by identifying effective faculty members, institutions can learn from their practices and potentially implement strategies that lead to higher student satisfaction and better learning outcomes. Moreover, schools can establish a continuous improvement and transparency culture, where faculty members are encouraged to reflect on their teaching practices and adjust based on the model's feedback. Lastly, educational institutions can use the model to maintain and improve the overall quality of education they provide, ensuring that faculty meet or exceed defined effectiveness standards.

Conclusion

Predicting faculty performance is a crucial aspect in academia as it allows for the identification of high-performing individuals who can be recognized and rewarded, as well as the identification of mid and low-performing individuals who can be provided with opportunities and training for improvement. By repeatedly utilizing prediction models, educational institutions can continuously enhance education quality and improve student outcomes and performances. Effective performance prediction will enable educational managers and faculty to allocate resources and instruction more accurately.

This research proves that Naive Bayes, Random Forest, and Decision Tree effectively predict faculty performance on the given data. Implementing these algorithms in the faculty evaluation process is still being evaluated, pending publication, dissemination, and scrutiny of the results to the major stakeholders. As such, this report does not present the specific model deployment on new and unseen data using the existing or the development of new application systems and the monitoring and maintenance process.

In conclusion, this study has significant room for improvement and expansion. One avenue for enhancement lies in including additional variables within the machine learning algorithms, such as results of peer evaluations, attendance reports, and faculty research outputs. These supplementary data sources promise to provide a deeper and more comprehensive understanding of faculty performance. Furthermore, exploring a wider array of machine learning algorithms, including supervised, unsupervised, semi-supervised, reinforcement, and deep learning methods, can offer valuable insights and enable comparative analysis. Adopting ensemble methods like bagging, boosting, and stacking can be considered to refine the study's methodology. These techniques leverage the collective predictive power of multiple models to enhance accuracy. Additionally, conducting an in-depth Error Analysis is crucial in advancing machine learning models. By thoroughly investigating the sources of model errors and discerning their underlying causes, we can refine and fine-tune the models, thus improving their overall performance.

Finally, while this study provides valuable insights into applying machine learning algorithms for faculty evaluation, it is imperative to acknowledge its limitations and adhere to the recommendations provided. By doing so, we can pave the way for a more robust, dependable, and widely applicable understanding of the efficacy of these algorithms in real-world academic assessment scenarios. The potential for further refinement and development in this field is substantial, and continued research will undoubtedly yield even more accurate and insightful results.

References

Asif, R., Merceron, A., Ali, S. A., & Haider, N. G. (2017). Analyzing undergraduate students' performance using educational data mining. *Computers & Education*, *113*, 177–194. https://doi.org/10.1016/j.compedu.2017.05.007

Brownlee, J. (2020, August 26). Train-Test Split for Evaluating Machine Learning Algorithms. Machine Learning Mastery. https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

Danielson, C. (1996). Enhancing professional practice: A framework for teaching. Alexandria, VA: Association for Supervision and Curriculum Development.

Galarnyk, M. (2022, June 6). Understanding Train Test Split (Scikit-Learn + Python). Medium. https://towardsdatascience.com/understanding-train-test-split-scikit-learn-python-ea676d5e3d1

Grandy, J. (1996), Differences In The Survey Responses Of Asian American And White Science And Engineering Students. ETS Research Report Series, 1996: i23. https://doi.org/10.1002/j.2333-8504.1996.tb01703.x

Isoré, M. (2009), "Teacher Evaluation: Current Practices in OECD Countries and a Literature Review", OECD Education Working Papers, No. 23, OECD Publishing. http://dx.doi.org/10.1787/223283631428

Kathiroli, P., & Vijayalakshmi., V. (2020, December 26). *Predicting the performance of instructors using Machine learning algorithms*. Unknown. https://www.researchgate.net/publication/347928189

Kumar, S., Jain, A., & Mahalakshmi, P. (2018). Enhancement of healthcare using naïve Bayes algorithm and intelligent data mining of social media. Int J Appl Eng Res, 13, 4109-4112.

Lalata, J. A. P., Gerardo, B., & Medina, R. (2019, June). A sentiment analysis model for faculty comment evaluation using ensemble machine learning algorithms. In Proceedings of the 2019 International Conference on Big Data Engineering (pp. 68-73).

Lin, L. (2021). Smart teaching evaluation model using weighted naive Bayes algorithm. *Journal of Intelligent & Fuzzy Systems*, *40*(2), 2791–2801. https://doi.org/10.3233/jifs-189320

Liu, H. Y., Tsai, H. M., Wang, I. T., & Chen, N. H. (2020). Predictors of self-perceived levels of creative teaching behaviors among nursing school faculty in Taiwan: A preliminary study. Journal of Professional Nursing, 36(3), 171-176.

Magno, C. (2009). A Meta-evaluation study on the assessment of teacher performance in an assessment center in the Philippines. The International Journal of Educational and Psychological Assessment, 3, 75-93.

Martin, F., Ritzhaupt, A., Kumar, S., & Budhrani, K. (2019). Award-winning faculty online teaching practices: Course design, assessment and evaluation, and facilitation. The Internet and Higher Education, 42, 34-43.

Meyer, D., Leisch, F., & Hornik, K. (2003). The support vector machine under test. Neurocomputing, 55(1-2), 169-186.

Munford, V. (2021). Designing a centralized faculty performance dashboard: Optimizing feedback and scholarly data reporting (Doctoral dissertation).

Navlani, A. (2018, December 28). Decision tree classification in Python tutorial. *DataCamp*. https://www.datacamp.com/community/tutorials/decision-tree-classification-python

Ordoñez-Avila, R., Salgado Reyes, N., Meza, J., & Ventura, S. (2023). Data mining techniques for predicting teacher evaluation in higher education: A systematic literature review. *Heliyon*, *9*(3), e13939. https://doi.org/10.1016/j.heliyon.2023.e13939

Pacol, C. A., & Palaoag, T. D. (2021). Enhancing sentiment analysis of textual feedback in the student-faculty evaluation using machine learning techniques. European Journal of Engineering Science and Technology, 4(1), 27-34.

Pedregosa, F., Varoquaux, G. Gramfort, A., Michel, V.; Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R.; Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Perrot, M., & Duchesnay, E. (2011). "scikit-learn: Machine Learning in Python". Journal of Machine Learning Research. 12: 2825–2830.

Pornel, J. B., & Saldaña, G. A. (2013). Four common misuses of the Likert scale. Philippine Journal of Social Sciences and Humanities, 18(2), 12-19.

Reinstein, I. (2017). Random Forest (r) Explained. kdnuggets. com.

Ray, S. (2019, February). A quick review of machine learning algorithms. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. http://dx.doi.org/10.1109/comitcon.2019.8862451

Shmilovici, A. (2010). Support vector machines. Data mining and knowledge discovery handbook, 231-247.

Stufflebeam, D.L. (2000). The methodology of meta-evaluation as reflected in by the Western Michigan University Evaluation Center. Journal of Personnel Evaluation in Education, 14(1), 95.

Sun, A., Lim, E. P., & Ng, W. K. (2002, November). Web classification using support vector machine. In Proceedings of the 4th International Workshop on Web Information and Data Management (pp. 96-99).

Theall, M., & Franklin, J. L. (2010). Assessing teaching practices and effectiveness for formative purposes. A guide to faculty development, 2, 151-168.

Tokuç, A. A. (2021, January 14). Splitting a Dataset into Train and Test Sets. Baeldung on Computer Science. https://www.baeldung.com/cs/train-test-datasets-ratio

Tufts University. (n.d.). Annual Review Process: Performance Rating Definitions | Access Tufts. Retrieved August 20, 2023, from https://access.tufts.edu/annual-review-process-performance-category-definitions

Vijayalakshmi, V., Panimalar, K., & Janarthanan, S. (2020). Predicting the performance of instructors using Machine learning algorithms. High Technology Letters, 26(12), 694-705.

Wang, Rui, Brian Hempton, John P Dugan, and Susan R Komives. (2008). "Cultural Differences: Why Do Asians Avoid Extreme Responses?" Survey Practice 1 (3). https://doi.org/10.29115/SP-2008-0011.

Welcome to python.org. (n.d.). Python.Org. Retrieved June 21, 2023, from https://www.python.org/

What is predictive analytics, and how does it work? (n.d.). Google Cloud. Retrieved June 16, 2023, from https://cloud.google.com/learn/what-is-predictive-analytics

Xia, X., & Yan, J. (2021). Construction of music teaching evaluation model based on weighted naïve Bayes. *Scientific Programming*, *2021*, 1–9. https://doi.org/10.1155/2021/7196197

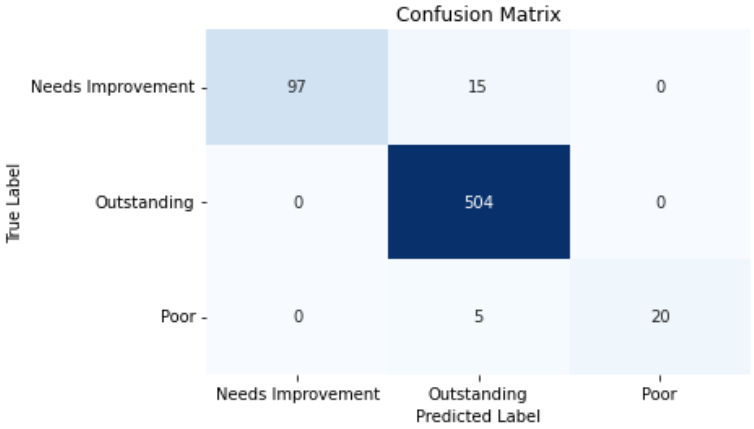
Yağcı, M. (2022). Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, *9*(1). https://doi.org/10.1186/s40561-022-00192-z

Appendix A

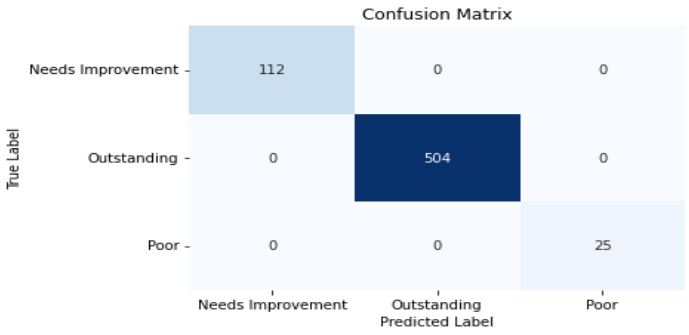
```
# Models with Confusion matrix and metrics
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix

# Load the dataset
df = pd.read_csv("june1dataset.csv")

# Define the categories or bins for teacher effectiveness
# Define bin boundaries and labels
bins = [-float('inf'), 2.0, 3.0, 3.34, 3.67, float('inf')]
labels = ['Poor', 'Needs Improvement', 'Satisfactory', 'Very Satisfactory',
'Outstanding']

# Categorize 'FEI' column
df['FEI'] = pd.cut(df['FEI'], bins=bins, labels=labels)

# Encode categorical labels to numerical values
label_encoder = LabelEncoder()
df['FEI'] = label_encoder.fit_transform(df['FEI'])

# Split the data into features (X) and target variable (y)
X = df.drop(['FID', 'SchoolYear', 'avgEoT', 'avgOCM', 'avgEoC', 'avgPoO', 'avgPoDL',
'avgAll'], axis=1)
y = df['FEI']  # Faculty Effectiveness Index

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

#print the values of the training and testing data
print('Shape of the Training and Test Dataset')
print(' X_train: ', X_train.shape)
print(' X_test: ', X_test.shape)
print('\n')

# Initialize and train different classification models
models = {
    'K-Nearest Neighbors': KNeighborsClassifier(),
    'Naive Bayes': GaussianNB(),
    'Support Vector Machine': SVC(),
    'Random Forest': RandomForestClassifier(),
    'Decision Tree': DecisionTreeClassifier()
}

for name, model in models.items():
    # Train the model
    model.fit(X_train, y_train)
```

```
    # Make predictions on the test set
    y_pred = model.predict(X_test)

    # Evaluate the model
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    cm = confusion_matrix(y_test, y_pred)

    # Print the evaluation results
    print(f"{name}:")
    print(f"  Accuracy: {accuracy:.4f}")
    print(f"  Precision: {precision:.4f}")
    print(f"  Recall: {recall:.4f}")
    print(f"  F1 Score: {f1:.4f}")
    print("  Confusion Matrix:")

    # Create a DataFrame for the confusion matrix
    cm_df = pd.DataFrame(cm, index=label_encoder.classes_,
columns=label_encoder.classes_)

    # Create a heatmap for the confusion matrix
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', cbar=False)
    plt.title('Confusion Matrix')
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    plt.show()

    print("\n")
```

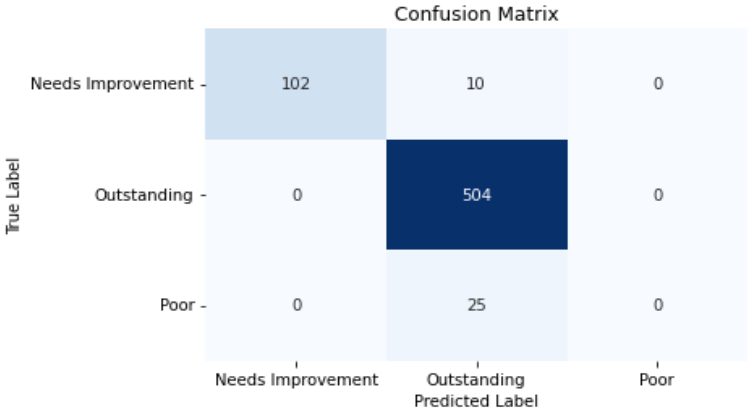
```
Shape of the Training and Test Dataset
 X_train:  (2562, 3)
 X_test:  (641, 3)


K-Nearest Neighbors:
  Accuracy: 0.9688
  Precision: 0.9700
  Recall: 0.9688
  F1 Score: 0.9678
  Confusion Matrix:
```



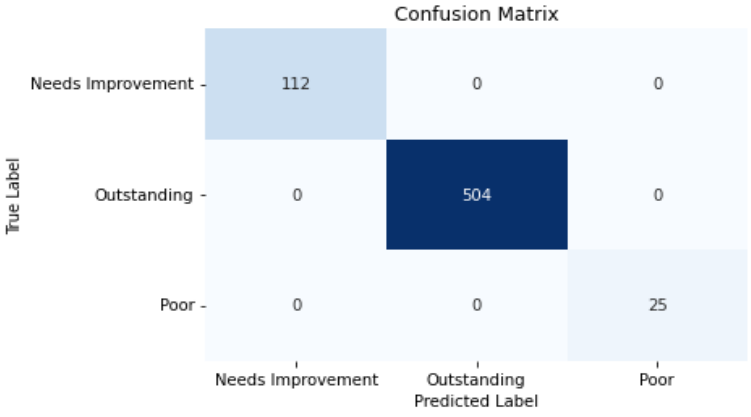
```
Naive Bayes:
  Accuracy: 1.0000
  Precision: 1.0000
  Recall: 1.0000
  F1 Score: 1.0000
  Confusion Matrix:
```




```
Support Vector Machine:
  Accuracy: 0.9454
  Precision: 0.9099
  Recall: 0.9454
  F1 Score: 0.9264
  Confusion Matrix:
```




```
Random Forest:
  Accuracy: 1.0000
  Precision: 1.0000
  Recall: 1.0000
  F1 Score: 1.0000
  Confusion Matrix:
```



```
Decision Tree:
  Accuracy: 1.0000
  Precision: 1.0000
  Recall: 1.0000
  F1 Score: 1.0000
  Confusion Matrix:
```