Journal of Systemics, Cybernetics and Informatics (2025) 23(3), 8-15
https://doi.org/10.54808/JSCI.23.03.8

Modelling Student Performance in a Structural Steel Graduate-based Module: A Comparative Analysis Between K-Nearest Neighbor and Dummy Classifiers

Masengo ILUNGA
Department of Civil and Environmental Engineering and Building Science, University of South Africa
Pretoria, Gauteng Province, South Africa

Omphemetse ZIMBILI
Department of Civil and Environmental Engineering and Building Science, University of South Africa
Pretoria, Gauteng Province, South Africa

Mampilo PHAHLANE
School of Computing, University of South Africa
Pretoria, Gauteng Province, South Africa

Agarwal ABHISHEK
Department of Mechanical Engineering, College of Science and Technology,
Royal University of Bhutan,
Phuentsholing, Bhutan

ABSTRACT

The predictive strength of the K-Nearest Neighbor (K-NN) and Dummy machine learning classification algorithms is investigated for students' final score. The dependent variable (label) is defined by a binary class, while the different assessments define the independent variables (features). The latter are the module student assessment marks, and the former covers students' final score. The two algorithms have been applied to the Structural Analysis IV, which is an engineering technology module in the Civil Engineering Advanced Diploma, taught at the University of South Africa. Competency level or graduate attribute characterises such a module. The results showed that the accuracy values of K-Nearest Neighbor (K-NN) and Dummy algorithms were 0.95 and 0.79 respectively. However, the values of recall, precision, f1-score, support, kappa coefficient and Matthews correlation coefficient, showed that the Dummy model predicted very poorly the "fail" instances, as opposed to the "pass" instances. Thus, the K-NN classifier outperformed the Dummy classifier. The two algorithms could be simultaneously recommended as guiding tools for academics in predicting students' final score (as fail or pass). However, K-NN is the only algorithm that could be used for both fail and pass.

Keywords: Machine learning, algorithm, prediction, student score, assessments, teaching, learning.

1. INTRODUCTION

Institutions of higher learning in the world mostly use acceptable forms of assessments to determine students' performance during the academic year. Grades or marks usually are a means of deciding on a student's competence level in a specific module. Thus, the overall performance is measured by the student capacity to pass each module in each academic programme. Assessments are administered based on the learning objectives, which align with teaching methodology [1]; [2]; [3]. The common practice in institutions of higher learning is to allocate different weights to each assessment as an integral part of the module. The deciding element on the student's fate is on the summation of the weights multiplied by each assessment mark respectively.

For engineering education in South Africa, an academic programme maintains its accreditation, through the display of proof for students to fulfil the graduate attributes as proposed by the Engineering Council of South Africa (ECSA). The council focuses on the student's demonstration of the learning outcomes at exit-level for him/her to be awarded a qualification. Therefore, evidence in teaching and learning should be gathered in this respect.

For assessments embedded in new engineering technology qualifications in South Africa, testing the graduate attribute (GA) implies the demonstration of competence level in a specific module, with the consequence for the student to succeed only when the

specific GA-based continuous assessment has been passed, irrespective of achieving more than 50 % overall in the module.

The use of Artificial Intelligence (AI) techniques becomes increasingly inevitable in education. Precisely, machine learning (ML) algorithms, which are more data driven techniques, are an enabler for data mining, exploration and modelling. The literature on ML applications for prediction of student success is well documented by [4] and [5]. Nonetheless, this study emphasises on K-NN and Dummy models. In the existing literature, Dummy model or K-NN model have been mostly employed separately, but cases are very rare where the Dummy model is compared with K-NN for students' performance, except e.g. [6].

K-NN is a simple ML algorithm where an object is graded by its neighbors' majority vote. The selection of K as an odd number in binary classification problems can be essential to eliminate tied votes [6]. In this study, K-NN was selected because of its simple computation though it has a very good ability to classify data [7]. Besides, the Dummy classifier is also a simple model as it does not require learning from the input data set. However, the complexity of the relationship between features and label is not always captured in these models. It would be also reasonable to move from simplicity to complexity for classification problems of student performance. It should be noted that the prediction of student performance has been often sparsely studied from assessments marks in a given course [8]. In supervised machine learning for classification problems, binary classes or multi-classes find their applications. The conversion of categorical entities into numerical classes eases the mapping process between the features and label.

The rest of the paper is organised as follows: an overview on the K-NN and Dummy algorithms within the context of machine learning applications is given. This is followed by the methodological approach on student performance prediction. Then, the results in the form of findings are presented and discussed. Finally, the conclusion generated from the findings is given and further research is suggested.

In what follows, student performance and student success will be used interchangeably. Module can be also understood as a course. Model, technique and algorithm will have the same meaning.

2. CLASSIFICATION SUPERVISED MACHINE LEARNING: K-NN AND DUMMY ALGORITHMS

K Nearest neighbor (K-NN)

The dimensional space is normally defined for the features and enables the K-NN machine learning algorithm to predict the labels for a given classification problem. The dominance of a specific class of the K closest data points is a major characteristic of this algorithm. These points are assumed to present similarity in their vicinity. Thus, the operation of the K-NN classification algorithm is essentially getting the K nearest neighbors to a newly found data point, and the dominant class among these neighbors is achieved by the voting process [9]. The distance between the new data point and all points in the training set is calculated. The steps involved in K-NN algorithm can be summarised as follows [9]:

1. Based on these distances, K nearest neighbors are selected.
2. The majority vote of the classes of these K neighbors is considered.
3. The new data point is assigned the majority class.

The choice of K value is very important in determining the accuracy of the algorithm [10]. K can be a small and positive number. When it takes 1 as a value, the object is allocated to its closest class of the neighbor. In K-NN model, Minkowski distance, Euclidean and Manhattan distance can be used [11]. The performance of K-NN was shown to be superior as compared to the Support Vector Machine technique, in terms for the prediction of student study period [10].

Dummy classification algorithm

Some ML algorithms have been simplified to predict instances using basic rules, without performing learning from the input data. Such algorithms are referred to as dummy ML. In the case of classification problems, they can be called dummy classifiers. They act as baseline models for establishing performance comparison with other so called complex models [12], or a "dummy" classifier structure on a simple stratified strategy, that randomly predicts labelled data by aligning with the class distribution of the training set [13]. Hence, for the Dummy Classifier, predictions are generated without the characteristics provided; it is a simply a benchmarking model to make comparison with other complex models and the specific baseline's behavior is chosen, based on a strategy parameter.

In classification problems during supervised learning, the split of the dataset is usually done for training and validation phases. Nonetheless, for cases of big data, the testing is further divided into 50% of validation and its remaining 50% constitutes finally the testing phase. There is no general rule for splitting the dataset. Sometimes it is 75% and 25% for training and testing respectively, even 80 % and 20% or 70% and 30 % respectively.

Hyperparameter tunning

This is an important step for ML algorithms. [14] has summarised some of the important aspects of this step. During this phase, the optimal parameters are discovered and fine-tuned, through an entity of several

parameters, which are combined and fitted to the model during training. Thus, the optimum parameters are selected and combined to find ultimately the desired accuracy. This yields to the GridSearch cross validation as one of the popular and effective hyperparameter tunning techniques. In this case, a discrete grid is set up and all hyperparameter combinations are tested with the grid and the model performance is carried out through validation. Hence the maximum of mean value is determined from the optimum combination of hyperparameter values, during cross-validation. Automatic hyperparameter optimisation was made possible through a grid search technique with cross-validation [13]. Besides, the Bayes Search cross-validation is also be used.

Statistical indicators for algorithm performance evaluation

ML algorithms usually perform on both training and testing. Most importantly, the interest is on the testing part to measure the prediction capability of the ML algorithm, using criteria or indicators. For example, these include Precision, Recall, F1-score, Receiving Operating Characteristic curve-Area Under Curve (ROC-AUC) score, and Cohen's Kappa (k) score for classification problems [14]. The use of confusion matrix is undeniable for classification problems, as shown by [15] and [16]. Below are equations of statistical indicators for evaluating the performance of the ML classifiers.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad \text{Eq. (1)}$$

$$Confusion\ matrix = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix} \quad \text{Eq. (2)}$$

$$Recall\ (postive) = \frac{TP}{TP+FP} \quad \text{Eq. (3)}$$

$$Recall\ (negative) = \frac{TN}{TN+FN} \quad \text{Eq. (4)}$$

$$Precision\ (positive) = \frac{TP}{TP+FN} \quad \text{Eq. (5)}$$

$$Precision(negative) = \frac{TN}{TP+FN} \quad \text{Eq. (6)}$$

$$F_1 Score = 2\frac{Precision \times Recall}{Precision+Recall} \quad \text{Eq. (7)}$$

Where:

True positives (TP) and **True negatives (TN)** correspond to the correctly predicted positive classes and negative classes respectively, **False positives (FP)** and **False negatives (FN)** correspond to the incorrectly predicted positive classes and negative classes.

From the above equations, the precision (positive/negative) shows the proportion of TP/TN with respect to all the PT positive (negative) labels. The recall (positive/negative) defines the proportion of TP/TN with respect to all possible positive (negative) labels. F_1 varies between 0 and 1, and reaches its lowest values when TP is zero, thus when all instances are FP. The maximum of F_1 is reached when FN and FP are both null, yielding to an ideal classification. The AUC, ROC, Kappa Coefficient and Matthews Correlation Coefficient (MCC) were used as indicators in assessing ML algorithm performance [17]. The increasing use of ROC-AUC, MCC, and F_1-score have been noted [13].

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP)(TN+FN)}} \quad \text{Eq. (8)}$$

$$k = \frac{2(TP \times TN - FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP)(TN+FN)}} \quad \text{Eq. (9)}$$

When the prediction obtained good results in all the four confusion matrix categories (TP, FN, TN, FP), proportionally both to the size of positive elements and the size of negative elements in the dataset, MCC can be a reliable metric producing a high score [17].

It was noted that in a specific problem, the MCC generated score more informatively and truthfully for classification problem than accuracy and F_1 score [17]. However, it should be noted that the purpose of this study was not to decide which of the metrics was superior over the others. When the ROC-AUC score is considered as a standard statistical indicator for models' evaluation, the models can be classified based on their scores [14]. These authors defined the following cases: "weak discriminator" is considered between two classes when a model scores less than 0.6; "acceptable discriminator," takes places, when the score falls in the interval (0.6 and 0.7). Hence, scores between 0.7 and 0.8, and scores over 0.9 are considered to be excellent and perfect discriminators respectively. The Kappa Coefficient displays an agreement between two observers for two-class classification problems, and it depicts a lot similarities with the MCC [18]. In addition, these authors used Brier score and MCC score. Majority of these indicators draw their foundation on the components of the confusion matrix.

3. DATA AVAILABILITY AND METHODS

Data used

The module Structural Analysis IV is an advanced diploma course and is made of 120 notional hours or 12 credits. The credits are translated into the total amount of time the student spends on different learning activities such as readings, lessons, assessments, etc. This module is housed and taught in the Department of Civil and Environmental Engineering, and Building Science, at the University of South Africa. The main characteristic of this module in the engineering technology programme, is Graduate Attribute 1 (GA1). This is part of the student's ability to satisfy the engineering educational objectives. This GA revolves around the student's capability to identify, formulate, analyse and solve broadly defined engineering problems. The competency requirement for

the module is that the student should achieve at least 50% for the final grade and pass assessment 2, a project submitted in a portfolio format, where GA1 is explicitly tested. In the South African higher education environment, the development of engineering programmes revolves currently around 11 graduate attributes. Hence, GA1 is obviously one of these. Details of students were not contained in the files of assessments that were used for the purpose of data processing. The number of labels/features were 146 pertaining to year 2023 semester 2 and did not have missing values.

Methods.

It is reiterated that the algorithms used in this study helped achieve supervised learning classification. Thus, the purpose was mapping between the features and labels. The categorical entities "fail" or "pass" were converted into the binary class. For the Structural Analysis IV module under consideration, the binary digit class were represented by the entities respectively. Even through there was no missing data values in the files, the appropriate statement code in python was used for that. Training and testing phases constituted 80% and 20% of data points respectively.

There was no need for variable scaling/standardisation since all assessments' marks ranged from 0 to 100%. The correlation among features was carried out mainly to determine the level of strength among these. However, it was not used to eliminate any of the 2 variables if they were found corrected, since in practice all marks were deemed to contribute to the calculation of the final mark.

PyCaret, as an automated machine learning library was used to achieve training and testing. However, for the purpose of the study, the built-in function namely "create" was used just to select ML algorithms that were relevant to this study. The code was adapted from https://nbviewer.org/github/pycaret/examples/blob/main/PyCaret%202%20Classification.ipynb.

The dummy model was chosen with the default strategy (randomly prior) as default baseline in the Scikit-Learn library. For the "prior" strategy, the most frequent class label in the observed y argument passed to fit (like "most_frequent") is always returned by the predict method.

```
DummyClassifier
DummyClassifier(constant=None, random_state=786, strategy='prior')
```

The K-NN 10 folds was chosen as default model in the library.

```
KNeighborsClassifier
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=-1, n_neighbors=5, p=2,
                     weights='uniform')
```

Most importantly, the default cross-validation of 10 folds was used in both algorithms. The hyperparameter tuning was achieved through the random grid search optimisation technique algorithm as default in the PyCaret library. The randomised state was set to achieve reproducibility ultimately. The performance of the modelling process was carried out using the indicators as briefly given earlier.

4. RESULTS AND DISCUSSION

It is recalled that K-NN and Dummy classification models were implemented on features and label, which represented assessments marks and the final mark respectively. The label was defined by the binary class. The characteristics of input variables were derived using descriptive statistics, as depicted in Table 1.

Table 1. Statistics of features (assessments); Ass1, 2, 3, 4 are assessments 1, 2, 3 and 4; as input variables.

	Ass1	Ass2	Ass3	Ass4
count	146.000000	146.000000	146.000000	146.000000
mean	59.506849	54.698630	57.815068	67.623288
std	34.483029	20.084432	32.008836	27.524319
min	0.000000	0.000000	0.000000	0.000000
25%	36.000000	51.000000	30.500000	50.000000
50%	60.000000	56.000000	55.500000	80.000000
75%	95.000000	60.000000	88.000000	87.000000
max	100.000000	100.000000	100.000000	100.000000

From Table 1, the mean value mark of assessments 1, 2, and 3 were relatively close with 10% variation from Assessment 2. Assessments 1 and 3 had a higher variability. Assessment 2 displayed the lowest 75% percentile.

The Pearson correlation was carried out to discover the associations between the input and output variables. Figure 1 summarises the correlation values, using the appropriate Seaborn library, in Python. The following conclusions can be drawn from the correlation values depicted in Figure 1: only assessments 2 and 4 displayed a relatively good correlation among the features whereas, assessments 2, and 3 were strongly associated with the score (namely result). These assessments each carried 30% in weight of the year mark.

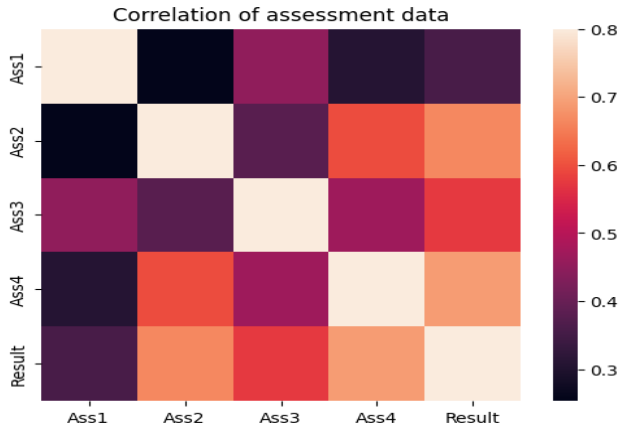

Figure 1. Correlation among assessments and between assessments and score. Ass1, 2, 3, 4 are assessments 1, 2, 3 and 4; as input variables.

The comparison of performance between the 2 algorithms was done based on the magnitudes of the different statistical indicators performed on the test dataset phase, as displayed in Tables 2 and 3. The experiments for each algorithm w repeated 10 times (folds) and averaged. For these results, the initial model was simply used to differentiate from the hyperparameter tunned model.

Table 2. Statistical indicator for initial Dummy model evaluation

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8182	0.5	1.0	0.8182	0.9000	0.0	0.0
1	0.7273	0.5	1.0	0.7273	0.8421	0.0	0.0
2	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
3	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
4	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
5	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
6	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
7	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
8	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
9	0.8000	0.5	1.0	0.8000	0.8889	0.0	0.0
Mean	0.7945	0.5	1.0	0.7945	0.8853	0.0	0.0
Std	0.0231	0.0	0.0	0.0231	0.0148	0.0	0.0

Table 3. Statistical indicator for initial K-NN model evaluation

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	1.0000	1.0000	1.0	1.0000	1.0000	1.0000	1.0000
1	0.9091	0.8125	1.0	0.8889	0.9412	0.7442	0.7698
2	1.0000	1.0000	1.0	1.0000	1.0000	1.0000	1.0000
3	0.9000	0.7188	1.0	0.8889	0.9412	0.6154	0.6667
4	1.0000	1.0000	1.0	1.0000	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0	1.0000	1.0000	1.0000	1.0000
6	0.9000	0.7188	1.0	0.8889	0.9412	0.6154	0.6667
7	0.9000	1.0000	1.0	0.8889	0.9412	0.6154	0.6667
8	1.0000	1.0000	1.0	1.0000	1.0000	1.0000	1.0000
9	0.9000	1.0000	1.0	0.8889	0.9412	0.6154	0.6667
Mean	0.9509	0.9250	1.0	0.9444	0.9706	0.8206	0.8436
Std	0.0492	0.1171	0.0	0.0556	0.0294	0.1831	0.1591

From Tables 2 and 3, the results showed that both models performed relatively well. The values of AUC suggested also the models to be characterised by excellent and perfect discrimination respectively. However, the higher values of all statistical indicators K-NN algorithm revealed that K-NN outperformed the Dummy algorithm, during test phase. The results of other parameters supported the high values of accuracy were good, except the Dummy model decisively yielded to null k and MCC.

K-NN algorithm could be used as a predictive tool. The values of recall and precision were relatively high, implying that the lower number of false positives and false negatives respectively. Hence the recall and precision values could support the high performance of K-NN model. The hyperparameter tuning exercise for the 2 models yielded the following for Dummy and K-NN models respectively.

Dummy model
Processing: 0%| | 0/7 [00:00<?, ?it/s] Fitting 10 folds for each of 4 candidates, totalling 40 fits
Original model was better than the tuned model, hence it will be returned. NOTE: The display metrics are for the tuned model (not the original one).

K-NN model
Processing: 0%| | 0/7 [00:00<?, ?it/s]Fitting 10 folds for each of 10 candidates, totaling 100 fits
Original model was better than the tuned model, hence it will be returned. NOTE: The display metrics are for the tuned model (not the original one).

From the above, it was obvious that the original models were considered. The easy visual representation of indicators was added, in an elegant way. Figures 2 and 3 show the confusion matrices of the 2 models respectively.

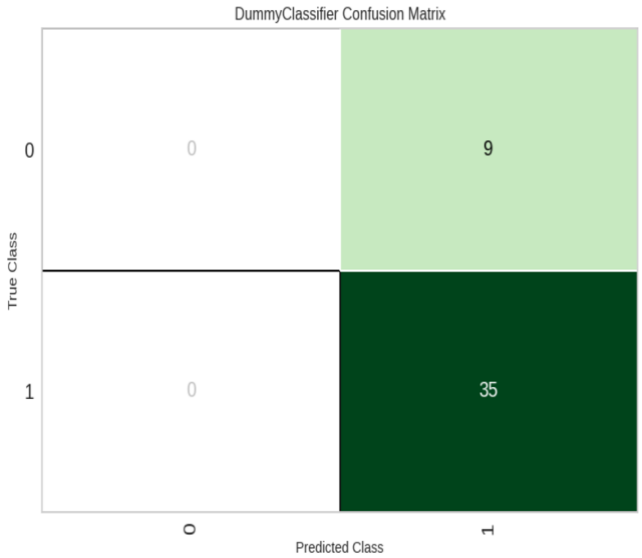

Figure 2. Confusion matrix of the Dummy algorithm

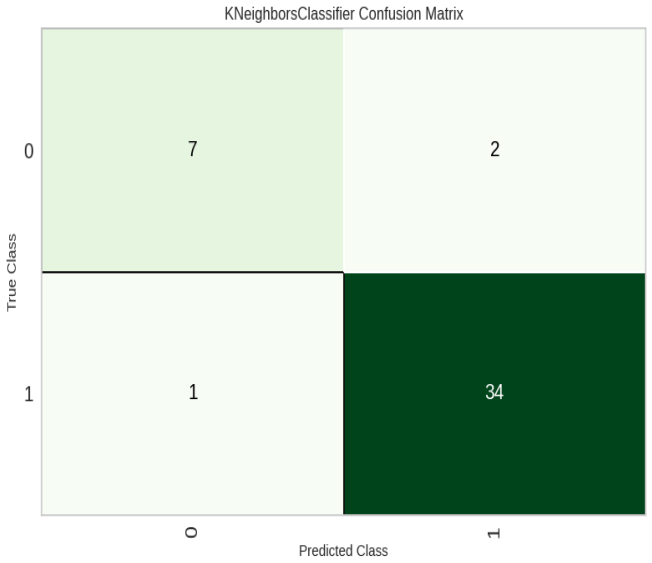

Figure 3. Confusion matrix of the K-NN algorithm.

From Figure 2, it was revealed that the number of FN and FP were 0 and 9 respectively. From these results, although the proportion of the sum of these entities predicted labelled was relatively low as compared to the sum of TN and TP instances, the performance of the Dummy model was only acceptable on positive instances. This yielded a higher model accuracy during test phase; however, all fail labels were completely mis-predicted.

Figure 3 revealed the prediction of 34 instances of true positives and 7 of TN was done accurately, However, the accurate prediction the TP overshadowed the TN.
The results of ROC (AUC) curves were added for the K-NN and Dummy algorithms as per Figures 4 and 5.

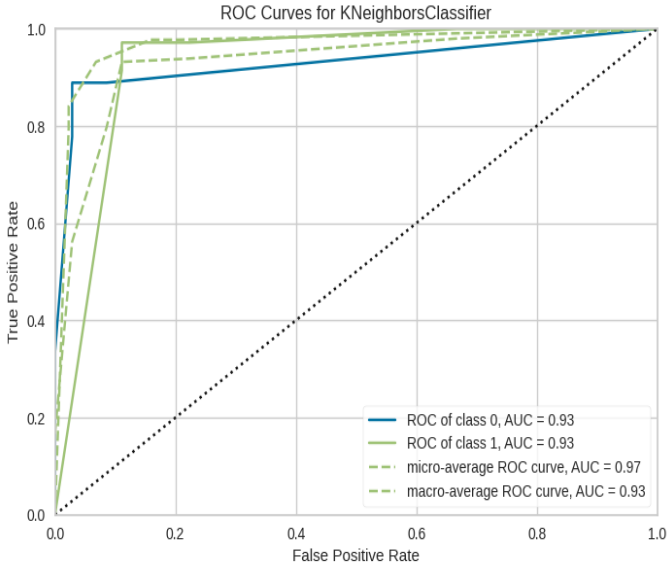

Figure 4. ROC curves for K-NN classification model

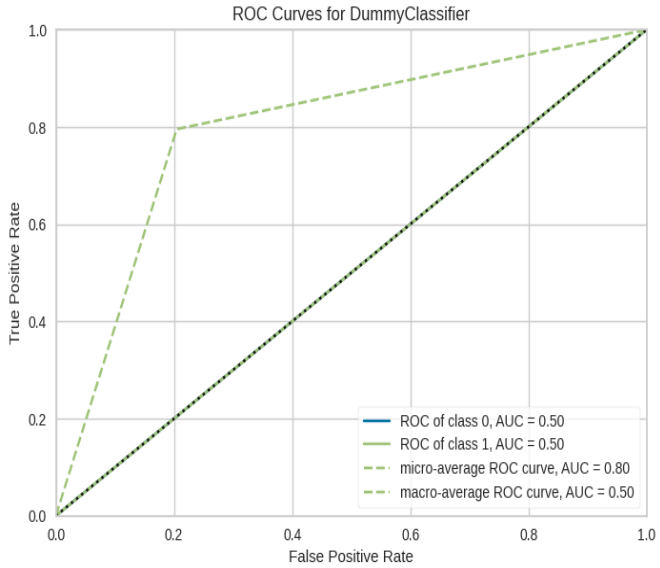

Figure 5. ROC curves for K-NN classification model

Comparing Figure 4 and 5, the previous results were enhanced, i.e. the area under the ROC curve for the K-NN algorithm was bigger than that of the Dummy classifier. This suggested that the former performed better that the latter for assessment prediction. This demonstrated that both the Dummy classifier and the K-NN were capable in predicting students' performance, only with reference to AUC values. Finally, the summary of the overall performance of the 2 classifiers were generated and presented in Figures 6 and 7.

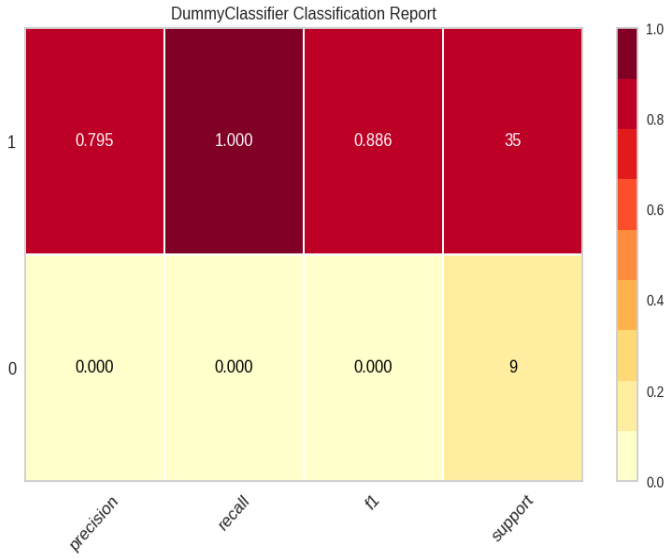

Figure 6. Classification report for Dummy algorithm

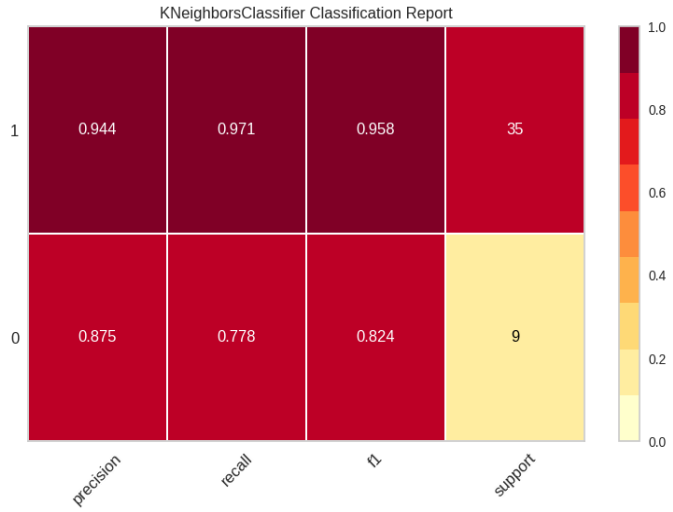

Figure 7. Classification report for K-NN algorithm

Figure 6 revealed that, in the case of Dummy algorithm, the positive labelled instances were truly predicted whereas the negative labelled instances were completely mis-estimated. Figure 7 showed that, K-NN performed very well about precision, recall, f1-score, and support were concerned. The Dummy algorithm was found to be weaker in a specific application by [19]. This could only be verified in this study on the "fail" values as far as its precision, recall, f1-score, and support were concerned. MCC was also completely weak.

Overall K-NN algorithm could fit for the purpose of simulating the non-competent students, such that means could be gathered to assist this type of students to be adequately prepared for assessments. Hence, informed strategies should be created to align with teaching and learning. In this way, these students could ultimately achieve the desired outcomes. K-NN algorithm was shown to be a non-linear mapping technique in the prediction of students' performance, as far as assessments are concerned.

This is an ongoing study that needs to be pursued as more datasets become available. Sufficient large datasets increase an ML model performance, however larger datasets may not necessarily [20]. Hence, the capability of the machine learning algorithms should be assessed as the availability of data increases.

5. CONCLUSION

The two machine learning models, i.e. K-NN and Dummy, were assessed comparatively, for predicting students' pass/failure in an academic environment. This was achieved primarily through the assessment activities performed by students during their academic year in the Structural Analysis IV module taught in engineering technology. These models showed their capability to map the association between the different assessments and the final score, by noting the superiority of the K-NN algorithm on the Dummy algorithm. Although the accuracy of the Dummy model was noticeably good, its precision, f1-score and MCC were completely compromised. The Dummy algorithm performed very well on true positive instances as opposed to true negative instances. Hence, academic practitioners could make use of K-NN primarily as a guide in assessing the students' competency as far as graduate attributes are concerned. This could give an overall sense of the pass rate in different courses before they focus on specific cases such as borderline cases. Therefore, efficiency in exploring machine learning could be promoted in enhancing best practices since handling assessment marks can be a tedious process. To investigate the generalization capability, the implementation of these algorithms should be reassessed as the number of data points increases every year in the module under investigation. It is recommended that more algorithms be implemented to assess students' performance in the same module. A sensitivity analysis for cross-validation (number of folds) should be investigated.

6. REFERENCES

[1] M. Montenegro-Rueda, A. Luque-de la Rosa, J. L. Sarasola Sánchez-Serrano, and J. Fernández-Cerero, **Assessment in Higher Education During the COVID-19 Pandemic: A Systematic Review**, Sustainability, Vol. 13, No. 19, 2021, p. 10509.

[2] J. C. González-Salamanca, O. L. Agudelo, and J. Salinas, **Key Competences, Education for Sustainable Development and Strategies for the Development of 21st Century Skills: A Systematic Literature Review**, Sustainability, Vol. 12, No. 24, 2020, p. 10366.

[3] R. Ajjawi, J. Tai, T. L. Huu Nghia, D. Boud, L. Johnson, and C. J. Patrick, **Aligning Assessment with the Needs of Work-Integrated Learning: The Challenges of Authentic Assessment in a Complex Context**, Assessment & Evaluation in Higher Education, Vol. 45, No. 2, 2020, pp. 304-316.

[4] I. El Guabassi, Z. Bousalem, R. Marah, and A. Qazdar, **Comparative Analysis of Supervised Machine Learning Algorithms to Build a Predictive Model for Evaluating Students' Performance**, International Journal of Online and Biomedical Engineering (iJOE), 17(02), 2021, pp. 90–105.

[5] P. Balaji, S. Alelyani, A. Qahmash, and M. Mohana, **Contributions of Machine Learning Models Towards Student Academic Performance Prediction: A Systematic Review**, Applied Sciences, Vol. 11, No. 21, 2021, p. 10007.

[6] A. S. Hashim, W. A. Awadh, and A. K. Hamoud, **Student Performance Prediction Model Based on Supervised Machine Learning Algorithms**, IOP Conference Series: Materials Science and Engineering, Vol. 928, No. 3, 2020, p. 032019.

[7] B. Assegaf, **Student Academic Performance Prediction on Problem-Based Learning Using Support**

Vector Machine and K-Nearest Neighbor, J. Telemat. Informatics, Vol. 5, No. 1, 2017, pp. 22-28.
[8] M. Ngonyama, M. Ilunga, and S. Mathenjwa, **Evaluating Decision Tree Model for Prediction of Student Performance in Teaching and Learning: Case of Structural Steel-Based Graduate Attribute Assessed Module**, Proceedings of the 28th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2024), pp. 236-240.
[9] S. Baladram, **K-Nearest Neighbor Classifier Explained: A Visual Guide with Code Examples for Beginners**, 2024. Available: https://towardsdatascience.com/k-nearest-neighbor-classifier-explained-a-visual-guide-with-code-examples-for-beginners-a3d85cad00e1.
[10] E. Novianto and S. Suhirman, **Comparison of K-Nearest Neighbor Classification Methods and Support Vector Machine in Predicting Students' Study Period**, Journal of Education and Science, Vol. 33, No. 1, 2024, pp. 32-45.
[11] G. Chaubey, P. R. Gavhane, D. Bisen, and S. K. Arjaria, **Customer Purchasing Behavior Prediction Using Machine Learning Classification Techniques**, Journal of Ambient Intelligence and Humanized Computing, Vol. 14, No. 12, 2023, pp. 16133-16157.
[12] S. Baladram, **Dummy Classifier Explained: A Visual Guide with Code Examples for Beginners**, 2024. Available: https://towardsdatascience.com/dummy-classifier-explained-a-visual-guide-with-code-examples-for-beginners-009ff95fc86e.
[13] G. Figueroa, Y. S. Chen, N. Avila, and C. C. Chu, **Improved Practices in Machine Learning Algorithms for NTL Detection with Imbalanced Data**, Proceedings of the IEEE Power & Energy Society General Meeting, July 2017, pp. 1-5.
[14] F. Ehsani and M. Hosseini, **Customer Churn Prediction Using a Novel Meta-Classifier: An Investigation on Transaction, Telecommunication, and Customer Churn Datasets**, Journal of Combinatorial Optimization, Vol. 48, No. 1, 2024, p. 7.
[15] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, **Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking**, IEEE Access, Vol. 8, 2020, pp. 90847-90861.
[16] I. Markoulidakis, G. Kopsiaftis, I. Rallis, and I. Georgoulas, **Multi-Class Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem**, Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference, June 2021, pp. 412-419.
[17] D. Chicco and G. Jurman, **The Advantages of the Matthews Correlation Coefficient (MCC) Over F1 Score and Accuracy in Binary Classification Evaluation**, BMC Genomics, Vol. 21, 2020, pp. 1-13.
[18] D. Chicco, M. J. Warrens, and G. Jurman, **The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment**, IEEE Access, Vol. 9, 2021, pp. 78368-78381.
[19] S. M. Malakouti, **Heart Disease Classification Based on ECG Using Machine Learning Models**, Biomedical Signal Processing and Control, Vol. 84, 2023, p. 104796