Iyer, S. et al. (Eds.) (2022). Proceedings of the 30th International Conference on Computers in Education.
Asia-Pacific Society for Computers in Education

Performance Prediction of Learning Programming - Machine Learning Approach

Thien-Wan AU[a*], Rahim SALIHIN [b] & Omar SAIFUL[c]
[abc]*Universiti Teknologi Brunei, Brunei Darussalam*
[a]*twan.au@utb.edu.bn,
[b]ak.salihin20@gmail.com
[c]Saiful.omar@utb.edu.bn

Abstract: Teaching and learning programming is a challenge faced by many educational institutions. In this paper we described using machine learning with data mining techniques to predict the performance of students using SMAC-based (social, mobile, analytics and cloud) programming learning tool (SPLT) that we developed for students to learn computer programming. Being able to predict and know students' performance has many advantages from educators', students' and administrators' perspectives for better learning, teaching, pedagogy design and institutional management. With the designed SPLT, experiments were conducted with 71 students from higher institutions who are learning computing programming. Various data were collected during the course of the experiment such as participants' demographics, programming background, logged data in SPLT, chat logs, pre- and post-tests scores for data mining attributes. Four classification algorithms were used to develop the classification model for the two datasets we prepared. WEKA was used to clean raw data, train and measure the performance of the models. Our experiment indicated that we were able to predict students' performance consistently with high accuracy of F-score using Random Forest classifier.

Keywords: analytics, machine learning, data mining, WEKA, social,

1. Introduction

Most modern learning institutions are highly competitive and operate in a complex environment. Therefore, there is a need to provide high-quality education and formulating strategies for evaluating the students' performance in order to identify future needs and the challenges faced by most learning institutions today. To implement student intervention plans is one of the best practices to overcome students' problems during their studies, at entry-level, and subsequent periods. Especially the management and educators will also benefit most in the planning, managing and designing course curriculum as needed to improve students' learning outcomes.

Learning computer programming has long been regarded as a challenge, particularly among computer science students, including those enrolled in the introductory programming course (Derus & Ali, 2014). Although several studies have documented the evidence of students' struggle in learning computer programming (Yang, Yang, & Hwang, 2014), there is no consensus on the main problem that students confront when taking this course (McCall & Kölling, 2014). However, one common issue that introductory programming students confront is their inability to think algorithmically due to a lack of problem-solving skills (Chung, Chou, Hsu, & Li, 2016) (Intisar & Watanobe, 2018). On the other hand, there is a rising trend in the adoption of machine learning techniques in programming education

due to its tremendous potential, especially in predicting students' performance and understanding students' behaviour in programming education.

In this study SMAC (Social, Mobile, Analytics, and Cloud)-based programming learning tool (SPLT) (Rahim, Omar, Au, & Mashud, 2022) was used as an online learning system with an architecture that adopted the SMAC concept in the programming learning environment for helping to mitigate the issues faced by learning computer programming. To further enhance the SPLT, machine learning using data mining techniques were employed using students' background data and also data collected through the SPLT to predict the performance of the students. Several models of the machine learning were tested with 2 slightly varied datasets to examine the best model.

In the next section of the paper, background of machines leaning in education would be briefly discussed. Section 3 discussed the main components of SPLT. Section 4 described the experiment followed by section 5 which presented the results of the experiment. Section 6 concluded the paper.

2. Machines learning in Educations

Learning has evolved into many sizes and shapes including e- and blended-learning platforms such as LMS, MOOC and other types of digital environment. While this has provided a more flexible environment it also introduces challenges as traditional direct face-to-face learning interaction has reduced tremendously and some even to zero. Challenges such as loss of motivation, high dropout rates and ineffective learning outcomes are very common nowadays. This is especially so in learning programming in a computer course. In such case predicting students. performance at risk using machine learning would be useful.

In (He, Bailey, Rubinstein, & Zhang, 2015), Sequentially Smoothed Logistic Regression (LR-SEQ) and Simultaneously Smoothed Logistic Regression (LR-SIM) were proposed. DisOpt 1 and DisOpt2 datasets were used to evaluate the two algorithms. Comparing the results with the baseline Logistic Regression (LR) algorithm, LR-SIM outperformed the LR- SEQ in termsof AUC when compared to the results with Logistic Regression (LR), where the LR-SIM had a high ACU value in the first week. This could be very useful for early intervention during the entry stage of higher institution.

In (Kasem, Shahrin, & Wan, 2018), the authors predicted students' performance undergraduates' performance at an early stage of their study program and identified modules that could serve as strong indicators of performance at the end of the degree program in a university running computing course. Data was collected on students' academic performance throughout the four years of their study as well as related demographic and background information. Several classification techniques and sampling methods, were experimented with to over data imbalance. The studies achieved reasonable accuracy in predicting three graduation classifications adopted in the university by using Naïve Bayes method with Feature Selection technique based on Gain Ratio attribute evaluator. The study also indicated that modules in semesters 2 to 4 are more prominent than modules of first semester in serving as strong predictors.

In (Kotsiantis, Patriarcheas, & Xenos, 2010) the authors proposed combinational incremental ensemble of classifiers to predict students' performance. In the studies, three classifiers were combined to calculates the prediction output. In the end a voting methodology is used to select the final prediction. The technique was found to be useful for continuously generated data. When a new sample arrived each classifier predicted the outcome. A voting system was used to select the final prediction. In the studies the dataset consisted of assignments marks with 1347 instances each having four attributes with four features. Three algorithms used for building the system were I Bayes (NB), Neural Network (NN), and WINDOW. When a new instance of observation arrives, all three classifiers predict the value, and the ones with high accuracy are automatically selected.

In (Osmanbegovic & Suljic, 2012) they used I Bayes (NB), Decision Tree (DT), and Multi layer perception (MLP) algorithms to predict students' success. The first part of data is collected from the survey conducted at the University of Tuzlain 2010–2011 which consisted of the first year economics students. Meanwhile the second part of the data were collected from students' enrollment database. Overall the dataset has 257 instances with 12 attributes. In the studies, it was found that NB scored a high accuracy score of 76.65% with a training time of less than1 second.

In (Lakkaraju et al., 2015), a machine learning framework was proposed to predict and identify at risk students who were likely to fail and not graduating on time. This studies collected students data from two schools in two districts. The machine learning algorithms purposed consisted of Support Vector Machine (SVM), Random Forest, Logistic Regression, Adaboost, and Decision Tree. These algorithms are evaluated using precision, recall, accuracy, and AUC for binary classification. Every student was ranked according to the risk score estimated from the classification based on the proposed models. This studies found that Random Forest performed the best.

4. SPLT

In this study we used SPLT (Rahim et al., 2022) as the learning platform for students' learning and to collect data for our purpose. There are two main requirements for SPLT: first, the system must be a cloud-based, mobile, and synchronous collaborative learning platform for programming education. Second, the system must generate and record relevant data necessary for educational data mining, mainly predicting students' performance and identifying at-risk students. Based on these two requirements, this study first analysed all widely experimented and proven effective programming learning tools identified from the SLR conducted by (Rahim, Omar, Au, & Mashud, 2021). The features of these identified learning tools were analysed to examine their effectiveness in enhancing students' problem-solving and collaborative skills. Analysing these learning tools helped this study identified the features that could adopt the SMAC elements into the system architecture. Figure 1 shows the identified features and the SMAC element they represent. The features were proposed to support a collaborative and synchronous programming environment that allowed distant students to collaborate seamlessly over a cloud-based system.

Based on the features in Figure 1, the architecture holds four primary functional units that worked in tandem to provide these features. In general, the synergy of these four functional units aided students in developing their collaborative skills and problem-solving ability through continuous collaboration in problem-solving oriented programming exercises. As illustrated in Figure 1, the four functional units are the *Synchronous Collaborative Unit* (SCU), *Exercise Analysis Unit* (EAU), *Portfolio Unit* (PU), and *Visualisation Unit* (VU). These units will be discussed throughout the chapter.

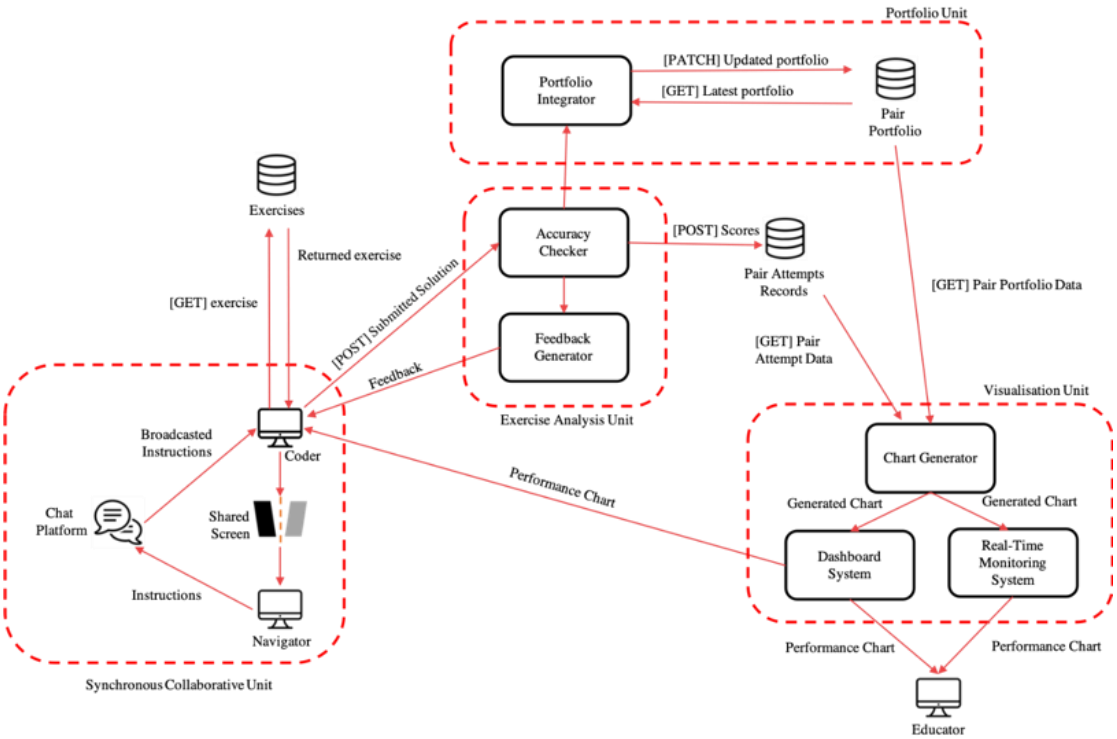

Figure 1. The system architecture for a SMAC-based Programming Learning Tool (SPLT) (Rahim et al., 2022)

5. *Synchronous Collaborative Unit (SCU)*

Referring to Figure 1, Synchronous Collaborative Unit (SCU) is a functional unit that specifies how combining three SMAC elements (social, mobile, and cloud elements) can establish a cloud-based (online) and mobile programming learning environment with synchronous collaboration capabilities. This unit resides at the architecture frontend to hold all the user interfaces for students to interact with the system. Overall, it included all student-oriented features that enabled students to exchange information (Khan, 2020), developed their problem-solving skills for programming and collaborate with other students in programming activities through an online environment.

This functional unit first examined the social element's incorporation into the architecture. Existing studies indicated frequent use of these two collaborative formats in programming education: 1) pair programming and 2) group programming. The goal of this unit was to establish an online and cloud-based learning environment.

3.2 Synchronous Collaborative Unit (SCU)

The SCU functional unit incorporated the cloud and analytic elements into the learning tool by holding two main components: *Accuracy Checker (AC)* and *Feedback Generator (FG)*. Residing at the business logic layer of the architecture, this unit and its two components aimed to perform the analytical tasks in the system by analysing students' performance and behaviour based on their activities in SPLT. All analytical processes in this architecture were done on the cloud to establish a proper cloud-based system. In general, this unit analysed the data generated by students, such as their solution accuracy, solving time, number of movements, and number of attempts. This functional unit helped to strengthen the grounding of all analytical processes by two outcome measurements suggested by (Ko, LaToza, & Burnett, 2015), namely *"success on task"* and *"time on task"*.

6. *Visualisation Unit (VU)*

This functional unit incorporates cloud and analytics elements with the goal to 4ategoriz the output generated by all analytical processes carried out in the Portfolio Unit (PU) and Exercise Analysis Unit (EAU). As illustrated in Figure 1, this unit has three main components: (1) *Chart Generator*, (2) *Dashboard System* and (3) *Real-Time Monitoring System*. These components resided both at the architecture's front- and back-end and 4ategori the data they received from the other functional units to generate learning progression charts.

7. *Portfolio Unit (PU)*

This functional unit also incorporated the cloud and analytic elements into the architecture, but it was responsible for establishing a portfolio system (PS) that classified students based on their performance over time. For this purpose, all data generated by the students in the system could be used to create a PS. The PS would provide insights into their learning progression, allowing educators to identify at-risk students quickly and effectively for responsive assistance. Studies have shown the importance of providing necessary assistance as quickly as possible (Azcona & Smeaton, 2017; Nakayama, Ishiwada, Morimoto, Nakamura, & Miyadera, 2018)

This functional unit had two core components: the *Portfolio Integrator* and the *Pair Portfolio Database*. The Portfolio Integrator must include all the functionalities required to establish the PS. First, it must extract the Exercise Analysis Unit data, specifically new students' activity data (such as scores and solving time). Then the Portfolio Integrator would get the latest portfolio of the student from the Pair Portfolio Database and update the students' portfolio accordingly based on this new data. To this end, the Port Portfolio Database is only responsible for storing, updating, creating, and deleting students' portfolio data.

8. Experiment

The population of this study involved students enrolling in the introductory programming modules. In the context of Brunei Darussalam, different institutions named the introductory programming module differently, such as *Fundamental of Programming*, *Introduction to Programming*, and *Programming 1*. This module is usually offered as a core module in the first or second semester of the first academic year, and all computing students must enroll in this module. This study applied the convenience sampling method to recruit participants from various public and private higher institutions offering introductory programming modules in Brunei Darussalam. Participation in this experiment was voluntary. An online registration form through Google Form was created and opened for one week to allow students to register at their own will. Additionally, the participants were allowed to withdraw without providing any valid reason. Table 1 showed the number of participants participating, participants completing the experiment, the levels and the institutions.

Data collection and analysis methods were essential components of this study because they provided the data necessary for understanding the effectiveness of SPLT in enhancing the problem-solving and collaborative skills of the programming students. Accordingly, the data collected in this study originated from three primary sources: (1) logs from the SPLT proof-of-concept system, (2) questionnaires, and (3) pre-and post-study test scores.

The pre-and post-study tests were administered before and after the experiment. The pre-test aimed to measure the participants' prior problem-solving ability, meanwhile the post-test measured any changes in their problem-solving ability after using the SPLT. Both tests had similar structures, which involved five questions that all participants must solve within allocated time. The five questions covered programming topics on code tracing, code debugging, iterations, conditions, and arrays. In ensuring the validity and reliability of the test questions and their marking rubrics, a senior programming lecturer from the Universiti was appointed to validate them.

Table 1. *Total number of recruited participants and who completed the experiment*

Institution	Educational level	No. of participants participating in the experiment	No. of participants completing the experiment
IBTE	National Diploma	4	4
Politeknik Brunei	Diploma Level 4 and Level 5	23	10
Universiti Teknologi Brunei	Bachelor's Degree	37	32
Universiti Brunei Darussalam	Bachelor's Degree	3	3
Micronet International College	Diploma Level 4 and Level 5	17	8
Kolej IGS	Diploma Level 5 and Bachelor's Degree	14	14

Table 2. *Attributes for dataset 1and dataset 2*

Dataset	**Attributes**
1	• Demographics: age and gender • Programming background: years, modules enrolled, and the project involved • SPLT log: up move, down move, total move, input score, process score, output score, total scores, and solving time • Class: Normalised learning gain
2	• Demographics: age and gender. • Programming background: years, modules enrolled, and the project involved • SPLT log: total move, total scores, and solving time • Class: Normalised learning gain

Data collected included students' demographics (age and gender), programming background (programming years, programming modules enrolled, and programming project involved), chat logs, and pre and post-study tests scores. These data can be used to develop the datasets required to train the predictive model that predicts or classifies students' performance based on their behaviour while using SPLT and identified weak-performing students as early as possible after using SPLT.

This study proposed classifying the students into at least three primary classes: low performing, average performing, and high performing. This grading strategy could also be used as the class attribute. The pre-and post-study test scores after using SPLT were used to calculate each student's 6ategoriza learning gain (NLG). Then, the grading strategy was used to label the class for each instance in the dataset based on its NLG value. Hence, the class attribute in the datasets of this study depended on the NLG of each student.

Two datasets were developed based on the collected data above mentioned. Table 2 showed the attributes of each dataset. In brief, the first dataset included students' demographics (age and gender), programming background (programming years, programming modules enrolled, and programming project involved), SPLT logs (up move, down move, total move, input score, process score, output score, total scores, and solving time), and the NLG class. This dataset examined whether finer data details would better classify students' performance after using the system.

Meanwhile, the second dataset had fewer attributes because it aggregated some related SPLT logs. In short, the SPLT logs included in Dataset 2 comprised the total move, total scores, and solving times. The rationale of this dataset was to examine whether aggregated details (lesser number of attributes) are better than segregated details (more attributes, as seen in Dataset 1) in classifying the students' performance.

All data mining processes, from data cleaning to model performance evaluation in this study, were done entirely by using WEKA Explorer (Bouckaert et al., 2008). The data cleaning conducted included identifying and handling outliers and missing values.

There were various supervised data mining algorithms to make the classifications. Classification was a supervised technique that 6ategorizat an instance based on the class attribute. This study adopted four classification algorithms to develop the classification model. These algorithms were adopted due to their wide application in the literature relating to educational data mining (Hämäläinen & Vinni, 2010; Poonguzhali, Sujatha, Sripriya, Deepa, & Mahalakshmi, 2021): *Naïve Bayes: C4.5 Decision Tree: K-Nearest Neighbour:Random Forest:*

In evaluating the model performance we adopted the three most commonly used tools to measure the performance of a classification model included confusion matrix, learning curves, and receiver operating curves (ROC) (Oprea & Ti, 2014).

This study experimented with filter (InfoGain and GainRatio) and wrapper approaches to explore and compare their performance with the original and discretised dataset. The performance for each model was evaluated and compared to identify the best performing model based on its F-Score value. Also, all models were tested by using the ten-fold cross-validation technique.

9. Results and Discussion

Results were tabulated in Table 3. The results indicated that Dataset 1 performed the best with the Random Forest classifier (F-Score = 0.957), especially when its continuous attributes were discretised and its irrelevant features were removed by using the GainRatio method. Meanwhile, Dataset 1 performed the poorest when using the KNN classifier (F-Score = 0.515) without applying 6ategorization and feature selections. In identifying the attributes that significantly contributed to the classifications, we further observed that the four leading indicators to classify students' performance are *age*, *prog_modules*, *prog_project*, and *scores*. And students' age, programming background and overall scores in SPLT could help classify their performance. Table 4 showed the confusion matrix of the best performing model for Dataset 1. The confusion matrix informs that the model can accurately classify 246 of the 254 instances, which translates to 95.6 per cent accuracy

Meanwhile, Dataset 2 contained fewer attributes but still hold the characteristics that could examine whether the data collected from SPLT, demographic, and programming background could be used to classify students' performance. The results showed that the KNN classifier did not perform well with Dataset 2. After applying the features reduction and 6ategorization, its performance only

peaked at F-Score = 0.658. However, the results showed that Dataset 2 performed the best when using the Random Forest classifier and its attributes were discretised (F-Score = 0.949). Table 4 showed the confusion matrix for the best performing model on Dataset 2. The confusion matrix informed that the model can accurately classify 246 of the 254 instances, which translates to 93.7 per cent accuracy.

Comparing all classifiers' accuracy (F-Score) on Dataset 1 and 2, the best performing model was Dataset 1 with Random Forest classifier, which involved applying 7 ategorization and the GainRatio method on its attributes (F-Score = 0.957). Meanwhile, the poorest performing model used the KNN classifier on Dataset 1 without 7 ategorizati (F-Score = 0.515). Most importantly, the outperforming result of Dataset 1 using Random Forest classifier suggests that having finer attributes and reducing them by using feature selections might results in better prediction model performance. This finding aligned with (Beniwal & Arora, 2012; Cahyani & Muslim, 2020; Doshi, 2014) on the usefulness of feature reduction that removed irrelevant and redundant attributes from the model. We postulated that having more relevant data about students' behaviour would help in predicting their performance better. Future educators who wish to improve the model may try adding more relevant attributes into the existing model. However, we recommended performing feature selections especially when some algorithms may not perform well with many attributes (Beniwal & Arora, 2012).

Furthermore, the results also demonstrated that the C4.5 Decision Tree classifier's performance peaked at F-Score = 0.924. In choosing the best model with the C4.5 Decision Tree classifier, this study examined the decision trees produced by each model and explored the classifier with the smallest decision tree (Table 4). The rationale is that fewer leaves will make it easier to interpret the decision tree which is in line with (Fayyad, Irani, & Arbor, 1990) that fewer leaves would reduce the classifier's error rate and increase its performance. Table 4 compares the sizes of the decision trees produced by the classifiers. Model no. 1 had the smallest tree produced (tree size = 57). However, the model had the lowest F-Score (0.886) compared to all other models. Hence, this study considers both F-Score and tree size to determine the best classification model with the C4.5 Decision Tree. The fourth model (model no. 4) satisfied these two factors from the results. the model had a tree size of 93 (105 leaves) and an F-Score of 0.924. Therefore, by considering the assertion of Fayyad et al. (1990), this study concluded that the decision tree produced by the fourth model could be used to identify the attributes that contributed to the classification of the students' performance

Table 3. *Comparisons of classifiers' accuracies (F-Score) on Dataset 1 and 2*

Experiment	**Classifier**	**Dataset 1**	**Dataset 2**
Original	NB	0.596	0.624
	C4.5	0.886	0.886
	RF	0.863	0.898
	KNN	0.515	0.652
Discretisation	NB	0.698	0.707
	C4.5	0.905	0.924
	RF	0.905	0.949
	KNN	0.646	0.614
Discretisation + Wrapper	NB	0.765	0.757
	C4.5	0.924	0.917
	RF	0.941	0.937
	KNN	0.666	0.658
Discretisation + GainRatio	NB	0.729	0.718
	C4.5	0.913	0.924
	RF	0.957	0.933
	KNN	0.669	0.567

Discretisation + InfoGain	NB	0.687	0.711
	C4.5	0.920	0.924
	RF	0.921	0.925
	KNN	0.649	0.607

Key:
NB – Naïve Bayes, C4.5 – C4.5 Decision Tree, RF – Random Forest, KNN – K-Nearest Neighbour

Table 4. *Sizes of Decision Trees Produced by Different Classifiers*

Experiment	Model No.	Data set	F score	No of leaves	Tree size
Original	1	1	0.866	29	57
	2	2	0.866	30	59
Discretisation	3	1	0.906	120 93	135
	4	2	0.924		105
Discretisation + Wrapper	5	1	0.924	112	125
	6	2	0.917	102	115
Discretisation + GainRatio	7	1	0.913	110	123
	8	2	0.924	100	111
Discretisation + InfoGain	9	1	0.920	100	111
	10	2	0.924	100	111

Table 5. *Confusion Matrix of best performing classifier on Dataset 1 and Dataset 2*

Dataset 1		*Predicted Cla s*		
		HL	HM	HH
Actual Class	HL	68	0	2
	HM	0	87	5
	HH	3	1	88
Dataset2				
Actual Class	HL	65	2	3
	HM	0	88	4
	HH	4	3	85

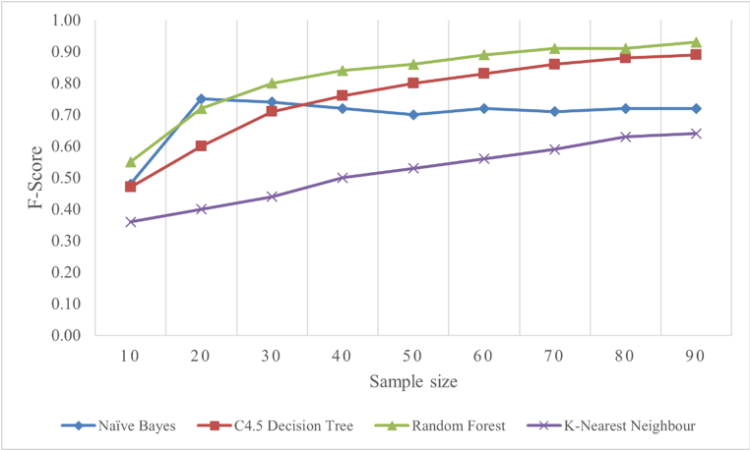

Figure 2. Learning curve of best-performing classifiers for Dataset 1

Figure 2 and 3 showed the learning curves and the performance of all classifiers improved with the increasing percentage of the sample size used in the training dataset. This finding indicated the accuracy of the current size used in training the classifiers, suggesting that adding more data to the training dataset will improve its accuracy. However, as indicated in Figure 2 and 3, further adding more data into the training dataset will not provide any significant benefit. Therefore the learning curve could be used to determine the point in which adding more data will not help improving the performance further and also to 9ategori the training time ensuring that the model will not suffer from overfitting (Mohr & van Rijn, 2022).

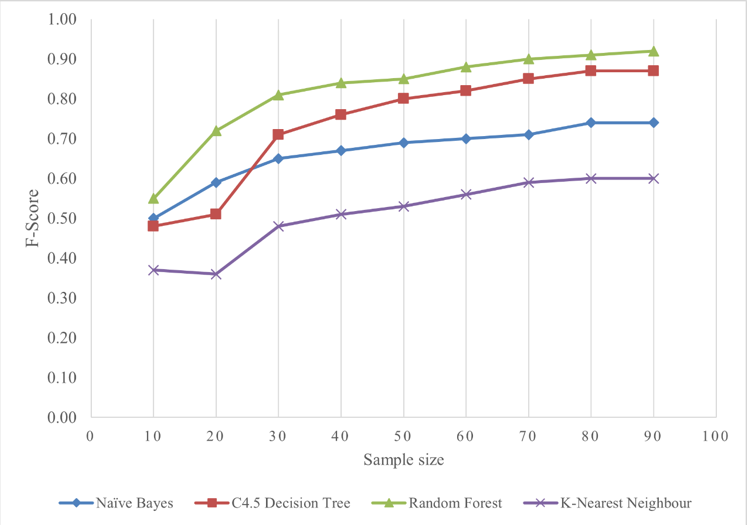

Figure 3. Learning curve of best-performing classifiers for Dataset 2

10. Conclusion

In this study we were able to prepare and develop two datasets from students' pre- and post-tests scores, demographics, programming background, and SPLT logs in order to develop the predicting machine learning models. The models were evaluated by different performance metrics, including the accuracy, true positive rate (sensitivity), F-Score. In particular the study 9ategor more on evaluating their F-Score due to its wide application in the literature. Four classifiers (algorithms) were used to train the model: Naïve Bayes, C4.5 Decision Tree, Random Forest, and K-Nearest Neighbour. In further enhancing the performance of the models, this study adjusted the parameters, including applying 9ategorization, 9ategorizatio, and feature selections (Wrapper, InfoGain, and GainRatio). Also, this study tested all models by using the ten-fold cross-validation technique. The evaluation results showed that Random Forest Classifier model was the best performing model in this study. In Dataset 1 the model performed the best when using the Random Forest classifier with 9ategorization and the GainRatio method. On the other hand, Dataset 2 performed the best when using Random Forest classifier with 9ategorization and Wrapper method. The learning curves for both datasets also demonstrated that adding more data into the training dataset will not increase the performance significantly as the models reached their plateau. The results from this study demonstrated the ability of using demographic data and behaviour data to predict at-risk students in early phase of the course and automate the process of grading students' performance. In the future, we recommend researchers to extend this study by examining the ability of using the two data to predict students' performance at different intervals during the duration of the course. This study believes that it would help educators to clearly identify any possible problems with the course structure and perform any necessary interventions to mitigate the identified problems.

References

Azcona, D., & Smeaton, A. F. (2017). Targeting at-risk students using engagement and effort predictors in an introductory computer programming course. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10474 LNCS*, 361–366.

Beniwal, S., & Arora, J. (2012). Classification and feature selection techniques in data mining. *International Journal of Engineering Research & Technology (Ijert)*, *1*(6), 1–6.

Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., & Scuse, D. (2008). Weka manual for version 3-6-0. *University of Waikato, Hamilton, New Zealand*, *2*.

Cahyani, N., & Muslim, M. A. (2020). Increasing Accuracy of C4. 5 Algorithm by applying discretization and correlation-based feature selection for chronic kidney disease diagnosis. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *12*(1), 25–32.

Chung, I., Chou, C., Hsu, C., & Li, D. (2016). A programming learning diagnostic system using case-based reasoning method. In *2016 International Conference on System Science and Engineering (ICSSE)* (pp. 1–4).

Derus, S. R. M., & Ali, A. Z. M. (2014). Integration of visualization techniques and active learning strategy in learning computer programming: a proposed framework. *International Journal on New Trends in Education and Their Implications*, *5*(1), 93–103.

Doshi, M. (2014). Correlation based feature selection (CFS) technique to predict student 10 ategorizat. *International Journal of Computer Networks & Communications*, *6*(3), 197.

Fayyad, U. M., Irani, K. B., & Arbor, A. (1990). What Should Be Minimized in a Decision Tree? *National Conference on 10ategoriza Intelligence*, 749–754.

Hämäläinen, W., & Vinni, M. (2010). Classifiers for educational data mining. *Handbook of Educational Data Mining*, 57–74.

He, J., Bailey, J., Rubinstein, B., & Zhang, R. (2015). Identifying at-risk students in massive open online courses. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 29).

Intisar, C. M., & Watanobe, Y. (2018). SMAC (social, mobile, analytics, cloud)-based learning intervention for introductory programming–the trend in the past 15 years. In *Proceedings of the 3rd International Conference on Applications in Information Technology* (pp. 23–28).

Kasem, A., Shahrin, S. N. A. M., & Wan, A. T. (2018). Learning analytics in Universiti Teknologi Brunei: predicting graduates performance. In *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)* (pp. 1–5). IEEE.

Khan, H. U. (2020). The role of SMAC (social media, mobility, analytics, cloud) for students and educators in online education. *Journal of Theoretical and Applied Information Technology*, *98*(6), 915–934.

Ko, A. J., LaToza, T. D., & Burnett, M. M. (2015). A practical guide to controlled experiments of software engineering tools with human participants. *Empirical Software Engineering*, *20*(1), 110–141.

Kotsiantis, S., Patriarcheas, K., & Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, *23*(6), 529–535.

Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K. L. (2015). A machine learning framework to identify students at risk of adverse academic outcomes. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1909–1918).

McCall, D., & Kölling, M. (2014). Meaningful 10ategorization of novice programmer errors. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1–8).

Mohr, F., & van Rijn, J. N. (2022). Learning Curves for Decision Making in Supervised Machine Learning – A Survey, 1–20.

Nakayama, H., Ishiwada, K., Morimoto, Y., Nakamura, S., & Miyadera, Y. (2018). Methods for analyzing the editing-processes of source codes in programming exercise for estimating learning situations. *2017 IEEE Conference on E-Learning, e-Management and e-Services, IC3e 2017*, 79–84.

Oprea, C., & Ti, P. Ș. (2014). Performance Evaluation of the Data Mining Classification Methods. *Analele Universităţii Constantin Brâncuşi Din Târgu Jiu : Seria Economie*, *1*(Special number-Information society and sustainable development), 249–253.

Osmanbegovic, E., & Suljic, M. (2012). Data mining approach for predicting student performance. *Economic Review: Journal of Economics and Business*, *10*(1), 3–12.

Poonguzhali, S., Sujatha, P., Sripriya, P., Deepa, V., & Mahalakshmi, K. (2021). Performance Evaluation of Classification Methods for Predicting Heart Disease.

Rahim, S., Omar, S., Au, T. W., & Mashud, I. M. (2021). SMAC (social, mobile, analytics, cloud)-based learning intervention for introductory programming–the trend in the past 15 years. In *International Conference on Computational Intelligence in Information System* (pp. 63–74). Springer.

Rahim, S., Omar, S., Au, T. W., & Mashud, I. M. (2022). SMAC-Based Programming Tool: Validating a Novel System Architecture. *International Journal of Emerging Technologies in Learning*, *17*(13).

Yang, T. C., Yang, S. J. H., & Hwang, G. J. (2014). Development of an interactive test system for students' improving learning outcomes in a computer programming course. *Proceedings – IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014*, 637–639.