ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **23**, Number 3, 2020, 238–249

Predicting Students Final GPA using 15 Classification Algorithms

Njoud ALANGARI and Raad ALTURKI

Department of Computer Science, Al-Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia

E-mail: naalanqari@sm.imamu.edu.sa, ralturki@imamu.edu.sa

Abstract. Predicting students' performance is one of the important tasks in Educational Data Mining (EDM). Early prediction can help the university to take actions that help students to graduate on time and have the best learning outcomes. It also helps the universities to save billions of dollars spent on students who fail, change major or dropout. In this paper, we report our work in student performance prediction using 15 classification algorithms. The experimental results show that we were able to predict students' final GPA with 91% accuracy for two models Naive Bayes and Hoeffding tree which significantly outperform competitive models across different datasets. The average accuracy for all the 15 classifiers was around 71%. We also analyzed the rules generated by tree-based and rule-based classifiers and found that some courses at early levels can have a major effect on the final GPA. In comparison to other works in the field, we were able to have more comprehensive analysis that produced better accuracy with higher variation of class values. These results show the potential for data mining to improve students' success rate by informing who are at risk in order for Instructors and administrators to provide the necessary support at the right time.

Key-words: Data Mining, Machine Learning, Classification, Prediction, Educational Data Mining.

1. Introduction

DM is the process of finding patterns, correlations and anomalies within data to extract knowledge from raw data. *Educational Data Mining* (EDM) is one of the areas that elicited some interest in researchers. Several DM methods have been used within the area including classification, association rule mining and clustering. Our review of the area has shown that classification was the most popular DM method that has been used by researchers. That could be because of the high potential of using prediction in EDM, the availability of rich data stored in the systems and the diversity of classifications methods.

One of the classical prediction problems in the field is to predict student performance and discover students who are most probable to fail in order to provide help and support. Also, data mining can help instructors and administrators to make better-informed decisions when designing courses and programs [1]. For example, knowing which course has the most effect on students' performance can encourage educators to give more attention to such course and provide more resources like better instructors, more supportive materials, etc. Achieving such results can be done by using DM methods and since DM is one of *Knowledge Discovery in Database* (KDD) process' stages, research can apply KDD Process to find and extract knowledge from data.

KDD include many stages that help to have an accurate prediction and transform the raw data to information. These stages are preprocessing data, mining data and post processing data. Preprocessing data include selecting the target data, transforming data, etc. Mining data is to extract patterns. Postprocessing data include interpreting and evaluating these patterns. Every stage had its methods, techniques, algorithms and suitable tools. Selecting the right choice for each stage from these depends on the nature of the dataset. For instance, selecting the target data can be done by extracting the data from database or combining the data from several sources like XML files and databases.

In the preprocessing stage, the dataset could suffer from a missing data problem, and this problem could be handled by using several methods like if the dataset is huge then deleting the records that have missing data is an option. However, If the dataset is not huge then there are other options like depending on the column type that have the missing data. For instance, if it is a categorical then replacing the missing with most frequent value but if it is a numerical then replacing the missing values with the average of that column. In addition, the dataset could need feature engineering by creating new features or deleting some of the features. Also, depending on the available attributes, sometimes we need to select the useful attributes manually based on our domain knowledge, and at other times we could use selecting feature techniques.

The way we preprocess the dataset depends on the objective of analyzing the dataset, the problem that we are trying to solve or the question that we are trying to answer. Any of these guide us through selecting data mining methods and algorithms. For instance, Regression and Classification are useful for prediction, outlier detection is used for fraud discovery, and Clustering and Association rule mining are good for finding the interesting patterns.

Post processing stage is about interpreting and evaluating the patterns that been extracted by data mining stage. Not all resulting models form DM stage are powerful. Evaluating these models are performed by using the suitable measures for the DM methods that been used. For instance, classification have many measures like Accuracy, Precision, Recall, etc. Others DM methods like Regression, and Association rule mining also have their own measures. Interpreting the resulting patterns depend on the DM algorithms that been used. Some of these algorithms like Decision tree-base or Rule-based are interpretable. And for other algorithms, using their prediction models for instance in application or program, help to benefit from theses pattern. In this paper, we applied KDD stages by conducting several experiments to predict the cumulative grade point average (CGPA) for computer science students at our college. In these experiments, we compared the performance of 15 classifiers in terms of accuracy. We used 10-fold cross-validation to test and validate the resulting models. The rest of the paper is structured as follows: we present the background and review some of the related work in section 2. In section 3, we provide a detailed description of our experiments using 15 classifiers. Then we present the results of the experiments in section 4. Finally, in section 5, we give our conclusion.

2. Literature Review

2.1. Classification Algorithm

Classification is an important DM method that has received attentions by many researchers in recent years. Classification can be categorized into different categories such as Decision Tree based algorithms, Rule based algorithms, and others like NaiveBayes, Support Vector Machine (SVM), etc. In this paper, we have used 15 classification algorithms and an overview of each algorithm is summaries in the following (Table 1).

Table 1. Overview on the 15 classification algorithms

	Classifier	**Description**
Decision Tree (DT) based algorithms	Decision stump	It is a simple decision tree that has only one level [2].
	HoeffdingTree or Very Fast Decision Tree (VFDT)	It is an anytime system that builds a decision tree for stream data [3].
	J48	It is the improved version of the decision tree algorithm C4.5 [2].
	logisticmodel tree algorithm (LMT)	It is a classification tree with logistic regression function at the leaves [2].
	Random Forest	It is constructed by building random trees from each subset of the dataset, then it uses the Bagging ensembles technique to predict the class of a new instance [2].
	Random Tree	It builds a decision tree based on a random subset features or columns [2].
	REP Tree	It builds a decision or regression tree using information gain and prunes it [2].
Rules Based Algorithms	Jrip	It implements RIPPER algorithms for fast and effective rule induction. It includes heuristic global optimization of the rule set [2].
	PART	It uses J4.8 to builds a tree. Then it extracts the rules from partial decisiontrees [2].
	ZeroR	It predicts the majority class, if the class is nominal and the average value, if the class is numeric. And it could be used as a baseline for the classifiers'performance [2].
Other Algorithms	Logistic Regression	It is a way to use regression for classification [4]. It models the probability of occurring a class attribute based on other attributes [2].
	Simple Logistic	It is a linear logistic regression with built-in-attribute selection [2].
	Naïve Bayes	It is a very simple Bayesian network composed of a directed acyclic graph (DAG) with only one parent and several children [5].
	SupportVector Machine (SVM)	When data is represented as data points in a graph, it searches for the linear optimal hyperplane that separates two classes [6]. For training SVM, the sequential minimal optimization algorithm (SMO) is used. And it uses polynomial or Gaussian kernels [2].

Other Algorithms	Classifier	Description
	K-nearest Neighbours (KNN)	It compares the new instances with the training set instances which have been stored in memory; it is an example of the instance-based learning algorithms and that why it is also named as instance-based learning with parameter k (IBK) [2]

2.2. Classification in Education

There is a clear growth in the research conducted in the DM field which was clearly indicated by the increase in the number of research publications and the specialized tools developed and used in the area [7]. Such an increase could be because of the effect of such studies in helping and improving Learning Outcomes using EDM [1]. Classification tasks are considered to be the most popular DM tasks used in the Educational Data Mining field [8].

In the literature, there are many classification algorithms that have been used like Decision Tree-based [9], [10], [11], Rule-based algorithms [9], [10], [12], Bayesian network [9], [10], [13], [11], KNN [9], [11], SVM, SMO, Logistic Regression and Simple Logistic [11]. We noticed that classification by using decision tree-based algorithms is the most popular. This could be because DTs are easy to interpret and understand and can handle numeric and categorical variables. To compare between the models that resulted from applying classification algorithms, the studies used different evaluating measures. For instance, the Accuracy metric was used in most of the studies; other metrics were used too like Precision, Recall, F-measures [13] and Area Under the receiver operating characteristic Curve (AUC) [14], [15], [16].

In classification, in addition to algorithms, the attributes also, play a major part in the results, different studies used different attributes in their analysis. For instance, Shahiri *et al.*[8] have reviewed the important attributes used in predicting students' performance like CGPA, internal assessment, students' demographic data, external assessments and psychometric factors [8]. Ibrahim *et al.* [17] and Ogor [18] agree with Shahiri *et al.* and they used CGPA. In addition to that Ogor [18] and Al-Barrak *et al.* [10] predicted internal assessment. Verma *et al.* [9] and Harwati *et al.* [19] used students' demographic data.

Hämäläinen *et al.* emphasized the importance of having a large dataset, preprocessing it and selecting a powerful model to get high Accuracy [20]. In our work, we were able to obtain relatively large dataset that contain more than 5566 students and we conducted extensive experiments for preprocessing and model selection. Badr *et al.* [21] and Al-Barrak *et al.* [10] aimed to predict students' performance in one course. While Al luhaybi *et al.* aimed to predict the student's performance at level two [22]. In our work, we aim to predict students' final GPA.

Badr *et al.* predicted the grade in "Programming" course based on the students' performance in "English" and "Mathematics" courses. The class attributes of the "Programming" course discretized to a binary class labels Good and Bad. Good label includes (A+, A, B+ and B) grades and Bad label include (C+, C, D, and D+) grades. They used a Classification Based on Association rules (CBA) algorithm. When they used students' grades in two English courses and two Mathematics courses, four rules were generated with accuracy of 62.75%. Then, when they used students' grades only in two English courses, four rules were generating with accuracy of 67.33% [21].

Al-Barrak *et al.* predict the grade in "Data Structure" course. They used the data from the "Data Structure" course, which includes student grades in quizzes, midterms, project, tutorial, final exam and total points obtained. They aimed to predict students' final exam performances, the total points obtained and student failure. They transformed the dataset into different forms like numerical data and categorical data with categorical class attribute for both of them. These datasets were used to predict final exam and to predict total points. The class attribute (total grade) discretized to five categories: A, B, C, D, and F. All attributes of the semester discretized into four categories: excellent, good, average, and poor. To predict student failure, they firstly re-categorized the total marks of the students into "F" and "P" where "F" indicated that the student failed the course (total points <60) and "P" represented the student passing the course (total points >=60). Secondly, since they wanted to predict student failure as early as possible before the final exam, they removed the final grades from the data set. They applied three classification algorithms: Naïve Bayes, the JRip rule-based algorithm, and the C4.5 decision tree. In predicting both final exam performance and final course grades, Naïve Bayes was the most accurate classifier with Accuracy of 81.01%. However, to predict failing students earlier before the final exam, they were able to build a 91% accuracy model using JRip algorithm [10].

Al luhaybi *et al.* aimed to predict student performance to identify the high risk of failure [22]. They also wanted to identify the key attributes affecting the predictive model. They used C4.5 and Naïve Bayes classification methods to predict the student's performance at level 2 based on their admission, course-related data and level 1 final grades. The class attribute is the overall grade obtained by the student in the targeted module. It has five values A, B, C, D, F and to improve the classification results, they merged the five values into three values: low, medium and high risk of failure. Their results show that Naïve Bayes gave accuracy of 88.48%, which was better than C4.5 algorithms that gave 84.29%. At the end of this section, we present the related work that we have reviewed and summarized them in Table 2.

Table 2. DM tasks and algorithms used in previous work.

Papers	DM Task	Algorithms
[10], [9], [11], [12], [23]	Classification (Decision tree based)	C4.5 (or J48), ID3, CART (or Simple Cart), REP Tree and Random Forest
[10], [9], [12]	Classification (Rule-based)	OneR and JRip
[10], [9], [11], [12], [13]	Classification (Bayesian network)	NB and BN
[11]	Classification	KNN
[11]	Classification	SMO
[11], [12]	Classification	Logistic
[24]	Association rule mining	Apriori
[24]	Clustering	K-means

3. Methodology

3.1. Data Preprocessing

The raw dataset that we obtained was in the Arabic language, and it contained students' transactional data of computer and information college over 7 years (approximately from 2010 to 2017). The number of transactions was 301,078 rows and 16 attributes for 5566 students. Each transaction records student personal information like ID, gender, DOB and city as well as student academic information like academic status, major, semester GPA, cumulative GPA (CGPA), course name, code, term, grade, number or register students, and more.

In this raw dataset, each student was represented in multiple rows. For instance, if a student studied 50 courses, then he/she will have 50 rows or more, one row at least for each course; a student can have multiple rows for each course when he/she withdrew or failed. For example, when a student withdrew (W) then got grade F then got grade C, he/she will have 3 rows (or transactions) for this course. Because of that, we had to examine this dataset to find the most suitable ways to prepare it.

After examining the dataset, we programmed a code to engineer the raw dataset to be more suitable for our purpose of predicting the CGPA. At this point preparing the dataset included translation and feature engineering. Further pre-processing steps were needed such as feature selection, feature engineering and missing data substitution. That included deleting some of the attributes such as city and semester GPA, because city had the same value for all students and the CGPA attribute include the information of semester GPAs since CGPA is the average of semester GPAs. Also, it included handling missing data and more.

As an example, of how we handled the missing data, when the missing values was in DOB attribute, we added the average of DOB values. However, when the absent information was course grade, we replaced it with "NT" which means "not taken". In addition, since CGPA is a numerical attribute, we discretized it to an ordinal attribute based on our college grading system as in Table 3. Furthermore, we could not mention all the preprocessing steps that we applied to the dataset because of the small space. At the end, we started from 301,078 rows for 5566 students and 1550 attributes; these 1550 attributes resulted from data engineering step. And we end up with 530 rows for 530 CS students and 64 attributes including the class attribute CGPA.

Table 3. Numeric CGPA grading system at our college and the equivalent classes for each CGPA range.

CGPA Range	4 Classes
$4.50 \leq$ CGPA ≤ 5	Excellent
$3.75 \leq$ CGPA <4.50	Very good
$2.75 \leq$ CGPA <3.75	Good
$2 \leq$ CGPA <2.75	Pass

3.2. Experiments Settings

In this paper, we applied 15 classifiers to predict students' CGPA. The 15 classifiers were Decision Tree-based classifiers (like Decision stump, Hoeffding Tree, J48, LMT, Random Forest, Random Tree, and REP Tree), Rule-based classifiers like (Jrip, PART and ZeoR), Naïve Bayes, KNN, SMO, logistics and Simple Logistics. We used all classifiers with the default setting except for KNN. In KNN, we activated the automatic search to find the best value of k (that is, the best number of neighbours). The search started from (k = 1) to (k = 10). In all experiments that we conducted, we used cross-validation technique with 10 folds. Furthermore, we selected ZeroR classifier as a baseline for the classifiers' performances.

4. Results

In this section, the results of the experiments that we conducted to predict CGPA are presented. Figure 1 show a comparison between the performance of these classifiers in terms of accuracy. The average Accuracy of all 15 classifiers is 71% as presented in the red line. As we can see, the performance of all classifiers were better than the performance of the baseline ZeroR classifier. Also, there are seven classifiers that gave accuracy above the average and seven below the average (baseline excluded). The classifiers that were above the average are Naïve Bayes, Hoeffding Tree, SMO, Random Forest, LMT, Simple Logistic and IBK. The classifiers that were below the average are Random Tree, J48, REP Tree, PART, Jrip, Decision Stump, and Logistic. The best result of 91% accuracy was achieved by 2 classifiers, Naïve Bayes and Hoeffding Tree. This was followed by SMO with 87% accuracy, Random Forest with 86% accuracy, LMT with 84% accuracy and Simple Logistic with 84% accuracy. As a result, we could say that the best performance of tree-based classifiers was achieved by Hoeffding Tree with 91% then by Random Forest and LMT. IBK (that is, KNN) achieved 76% accuracy, and tree-based classifiers Random Tree, J48, and REP Tree achieved 96%, 65% and 63% accuracies, respectively. In addition, rule-based classifiers PART and Jrip achieved 68% and 67% accuracies, respectively. The worst results, 48% and 49%, were achieved by Decision Stump and Logistic.

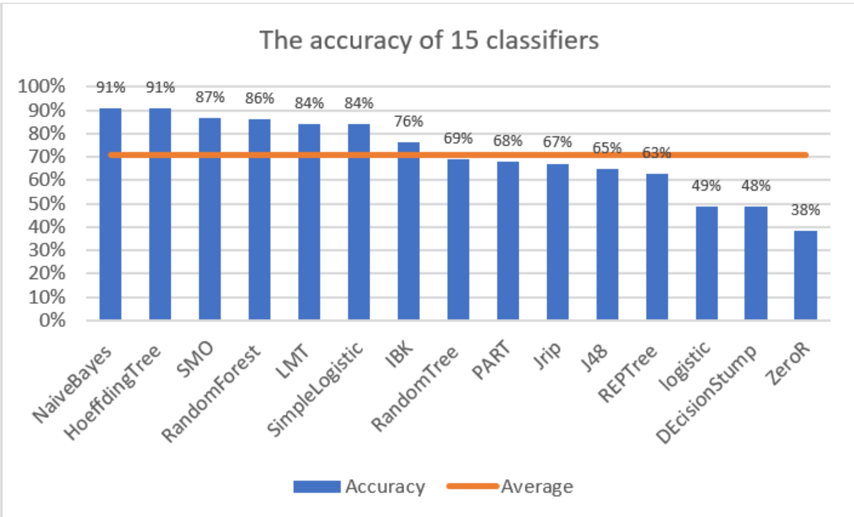

Fig. 1. Comparing between the performance of the 15 classifiers and their average.

In general, Tree-based performance varies from 86% to 48%, where 3 of these classifiers were above the average 71% and 4 of these classifiers were below the average 71%. Rule-based performances were around 68%, which is below the average (Figure 1). The resulting rules from rule-base algorithms or from tree-based algorithms followed by two numbers, where the first number indicates how many instances this rule cover and the second number indicates how many instances were misclassified by this rule. We are interested more in the rules that cover the highest number of instances and misclassified the least number of instances. We presented the results of JRip as an example of rule-based algorithms and J48 as an example of tree-based algorithms. The results of JRip were 22 rules presented in Table 4.

Table 4. The rules that result from predicting CGPA using Jrip classifier

Rule No.	Rule	Covered	Misclassified
1	(CS322 = F) =>CGPA=Pass	46	12
2	(STA111 = F) and (PH103 = F) =>CGPA=Pass	19	3
3	(CS242 = D) and (CS141 = F) and (ENG208 = C) =>CGPA=Pass	8	1
4	(CS370 = D) and (CS310 = F) =>CGPA=Pass	7	2
5	(CS322 = D) and (CS471 = F) =>CGPA=Pass	7	3
6	(CS340 = F) and (CS220 = F) =>CGPA=Pass	4	0
7	(CS370 = F) =>CGPA=Pass	3	1
8	(IDE133 = D) and (CS242 = F) =>CGPA=Pass	2	0
9	(CS141 = A) and (CS215 = A) and (STA111 = A) =>CGPA=Excellent	44	0
10	(CS220 = A) and (CS344 = A) =>CGPA=Excellent	26	4
11	(CS310 = A) and (PH104 = A) and (ARB104 = A) =>CGPA=Excellent	8	1
12	(CS445 = A) and (CS344 = A) and (DOB >= 1991) =>CGPA=Excellent	7	0
13	(CS104 = A) and (CS221 = B) and (CS391 = A) =>CGPA=Excellent	12	4
14	(ENG140 = A) and (CS340 = B) and (ACC100 = A)=>CGPA=Very Good	51	1
15	(CS242 = A) =>CGPA=Very Good	42	10
16	(ECO100 = A) and (CS220 = B) and (MATH227 = A) =>CGPA=Very Good	17	0
17	(CS492 = A) and (CS471 = NT) and (QUR301 = A) =>CGPA=Very Good	31	8
18	(CS340 = A) =>CGPA=Very Good	7	1
19	(CS370 = A) =>CGPA=Very Good	12	4
20	(CS330 = A) =>CGPA=Very Good	6	2
21	(CS445 = B) and (CS391 = A) and (BUS100 = A) =>CGPA=Very Good	4	0
22	=>CGPA=Good	167	8

For instances based on rule (14) when the 51 students got A in course EGN140 (that is, English Language 1), B in course CS340 (that is Artificial Intelligence) and A in course ACC100 (that is, Principles of Accounting), their CGPA was Very Good. This rule misclassified one student. Rule (9) covers 44 students and it did not misclassify anyone. When the 44 students got A in the three courses CS141 (that is Computer Programming 2), CS215 (that is Design and Analysis of Algorithms) and STA111 (that is Introduction to Probability and Statistics), they got Excellent CGPA. And the same applied to the rest of the rules. Therefore, as we saw in Table 4, we could say that some courses can have an indication of the final GPA especially if students got F at some courses, they will have Pass grade in the final GPA. For example, in rule (1) 46 students got F in CS322 (Operating Systems), and in rule (2) 19 students got F in both STA111 (Introduction to Probability and Statistics) and PH103 (General Physic). In other courses, if students got A at some courses, they will have Excellent GPA like in rule (9) as we explained it above. In addition to that, the resulting tree from J48 classifier was a huge tree, and to facilitate presenting the tree we used special tool to simplify it (PrefuseTree plugin). We chose to present two branches of the tree where we have found some interesting rules as shown in Figure 2 and Figure 3.

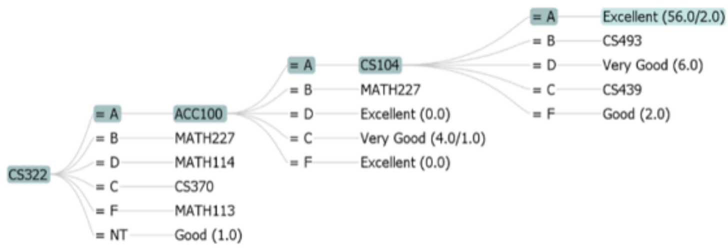

Fig. 2. The rule (CS322=A, ACC100=A, CS104=A => CGPA=Excellent) is highlighted as one of the rules that this part of J48's tree is showing .

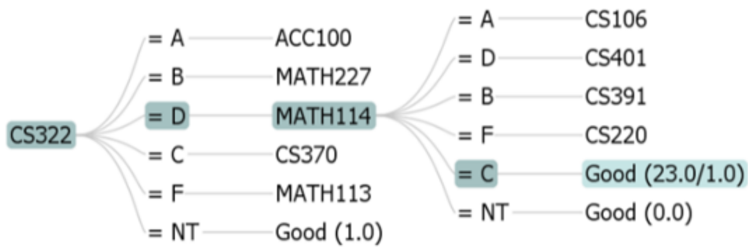

Fig. 3. The rule (CS322=D, MATH114=C=> CGPA=Good) is highlighted as one of the rules that this part of J48's tree is showing .

Figure 2 shows the first part of the J48 tree where the accuracy was 65%. Since the root node is CS322 (which is Operating Systems), we could say that is it the best predictor of CGPA. In addition, an example of rules that could be important is the highlighted rule which covered or applied to 56 students and only 2 students were misclassified by this rule. It means 56 of the students who got A in the 3 courses CS322 (that is, Operating Systems), ACC100 (that is, Principles of Accounting) and CS104 (that is, Discrete Structures) achieved Excellent CGPA.

Moreover, Figure 3 presented another part of J48's tree where the rule that highlighted applied to 23 students and only one student was misclassified by this rule. It means 23 of the students who got D in course CS322 (that is, Operating Systems) and C in course MATH114 (that is, Applied Calculus 2) achieved Good CGPA. When comparing our results to previous work in the area, we have found some interesting results where our work achieved better or worse results in some cases. For example, Al luhaybi *et al.* [22] showed that the best model performance in term of accuracy was the Naïve Bayes algorithm with 84% accuracy, while in our results the best predictors were Naïve Bayes and Hoeffding Tree with accuracy of 91%. This could be because we used a larger dataset, as their dataset included only 129 records and 39 attributes.

On the other hand, the accuracy we achieved was slightly less than the accuracy achieved by Ogor [18] where the best accuracy achieved was 97% as a result of using algorithm C5.0 to predict students' CGPA. That could be because Ogor used a larger dataset (1396 records and 78 attributes) and did not use the 10 folds cross-validation technique as we did. Also, as we know, it is easier to predict a class attribute with fewer variation of class values such as 2 or 3 classes. In our work, the class attribute that we predicted had 4 class values, whereas papers Ogor [18] and Al luhaybi *et al.* [22] predicted class attributes with only 3 classes.

5. Conclusions

In this paper, we have conducted comprehensive comparison of the performance of 15 classification algorithms from different categories to predict the final GPA for computer science students. Our experimental results showed that Nave Bayes and Hoeffding Tree achieved the best performance with Accuracy of 91%. Moreover, we analyzed the rules generated by tree-based and rule-based classifiers and found that some courses can have an indication of final GPA at early levels; That is clearly shown in some courses when students got F or A, a final GPA prediction of Pass or Excellent grade is achieved respectively. Example of these courses are "Operating Systems", "Statistics", "General Physic", "Computer Programming" and "Algorithms".

In comparison to other work in the field, we were able to have a more comprehensive analysis that has produced better accuracy with a higher variation of class values. We hope such prediction models can help in providing a better educational experience. Such prediction of final GPA or grade of a course can help academic advisors and administrators to make better-informed decisions. Also, it can help the students who are at risk to make more efforts to enhance their performance. In the future, we plan to study the effect of Dimensions Reduction methods on the performance of the prediction. We also plan to compare such classifiers using different data subsets to examine the predictive power for their features especially first and second years courses. Furthermore, we want to explore other classification algorithms such as the Deep Neural Network (DNN) either to predict the final GPA or for predicting the grade of one course based on the courses from previous levels.

References

[1] I. A. ALSHAMMARI, M. D. ALDHAFIRI, and Z. AL-SHAMMARI, *A Meta-Analysis of Educational Data Mining on Improvements in Learning Outcomes.*, Coll. Stud. J., **47**(2), pp. 326–333, 2013.

[2] E. FRANK and I. H. WITTEN, *Data Mining: Practical Machine Learning Tools and Techniques*, **54**(2). Morgan Kaufman, 2011.

[3] P. DOMINGOS and G. HULTEN, *Mining high-speed data streams*, in Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining – KDD ’00, 2000, pp. 71–80.

[4] N. LANDWEHR, M. HALL, and E. FRANK, *Logistic model trees*, Mach. Learn., **59**(1–2), pp. 161–205, 2005.

[5] S. B. KOTSIANTIS, *Supervised Machine Learning: A Review of Classification Techniques*, Informatica, **31**, pp. 249–268, 2007.

[6] H. JIAWEI, M. KAMBER, J. HAN, M. KAMBER, and J. PEI, *Data Mining: Concepts and Techniques*, 2012.

[7] C. ROMERO and S. VENTURA, *Educational data mining: A review of the state of the art*, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, **40**(6) pp. 601–618, 2010.

[8] Amirah Mohamed SHAHIRI, Wahidah HUSAIN, and Aini Abdul RASHIDI, *A review on predicting student’s performance using data mining techniques*, Procedia Comput. Sci., **72**(3), pp. 414–422, 2015.

[9] K. VERMA, A. SINGH, and P. VERMA, *A review on predicting student performance using data mining method*, 2016.

[10] M. A. AL-BARRAK and M. S. AL-RAZGAN, *Predicting students’ performance through classification: A case study*, J. Theor. Appl. Inf. Technol., **75**(2), pp. 167175, 2015.

[11] S. B. KOTSIANTIS, C. J. PIERRAKEAS, and P. E. PINTELAS, *Preventing Student Dropout in Distance Learning Using Machine Learning Techniques*, in Knowledge-Based Intelligent Information and Engineering Systems, pp. 267–274, 2003.

[12] G. W. . DEKKER, M. . PECHENIZKIY, and J. M. . VLEESHOUWERS, *Predicting students drop out: A case study*, in EDM’09 - Educational Data Mining 2009: 2nd International Conference on Educational Data Mining, pp. 41–50, 2009.

[13] E. A. AMRIEH, T. HAMTINI, and I. ALJARAH, *Mining Educational Data to Predict Student’s academic Performance using Ensemble Methods*, Int. J. Database Theory Appl., **9**(8), pp. 119–136, 2016.

[14] B.-H. KIM, E. VIZITEI, and V. GANAPATHI, *GritNet: Student Performance Prediction with Deep Learning*, CoRR, vol. abs/1804.0, 2018.

[15] W. WANG, H. YU, and C. MIAO, *Deep Model for Dropout Prediction in MOOCs*, in Proceedings of the 2nd International Conference on Crowd Science and Engineering - ICCSE’17, pp. 26–32, 2017.

[16] J. WHITEHILL, K. MOHAN, D. SEATON, Y. Rosen, and D. Tingley, *Delving Deeper into MOOC Student Dropout Prediction*, Feb. 2017.

[17] Z. IBRAHIM and D. RUSLI, *Predicting students’ academic performance: comparing artificial neural network, decision tree and linear regression*, 21st Annu. SAS Malaysia Forum, 5th, 2007.

[18] E. N. OGOR, *Student Academic Performance Monitoring and Evaluation Using Data Mining Techniques*, in Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007), 2007, pp. 354–359.

[19] HARWATI, A. P. ALFIANI, and F. A. WULANDARI, *Mapping Student’s Performance Based on Data Mining Approach (A Case Study)*, Agric. Agric. Sci. Procedia, **3**, pp. 173–177, 2015.

[20] W. Hämäläinen and M. VINNI, *Classifiers for Educational Data Mining*, Handb. Educ. Data Mining, Data Min. Knowl. Discov. Ser., pp. 57–71., 2010.

[21] G. BADR, A. ALGOBAIL, H. ALMUTAIRI, and M. ALMUTERY, *Predicting Students’ Performance in University Courses: A Case Study and Tool in KSU Mathematics Department*, Procedia Comput. Sci., **82**, pp. 80–89, 2016.

[22] M. Al LUHAYBI, A. TUCKER, and L. YOUSEFI, *The Prediction of Student Failure Using Classification Methods: A Case study*, in Computer Science & Information Technology, 2018, pp. 79–90.

[23] B. JANTAWAN and C.-F. TSAI, *The Application of Data Mining to Build Classification Model for Predicting Graduate Employment*, IJCSIS) Int. J. Comput. Sci. Inf. Secur., **11**(10), 2013.

[24] A. A. AL-SHARGABI and A. N. NUSARI, *Discovering vital patterns from UST students data by applying data mining techniques*, in 2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010, 2010, **2**, pp. 547–551.