FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 3 No. 1, March, 2019, pp 231 - 239

A COMPARATIVE STUDY OF BASE CLASSIFIERS IN PREDICTING STUDENTS' PERFORMANCE BASED ON INTERACTION WITH LMS PLATFORM

[1]Umar Isa Usman, [1]Salisu Aliyu, [1]Barroon Isma'eel Ahmad, [2]Abubakar Yusuf

[1]Department of Computer Science, Ahmadu Bello University Zaria
[2]Department of Computer Science, Nuhu Bamalli Polytechnic, Zaria.

Corresponding Author's Email: aliyusalisu@abu.edu.ng

ABSTRACT
E-learning platforms known as Learning Management System (LMS) generate huge amount of data that need to be examine in order to derive meaning out of it. This can be achieved using data mining techniques on large educational data, this is a field also known as Educational Data Mining (EDM). One popular application of EDM is prediction of students' performance. This application seems to be difficult due to the diverse nature of the variables that affect performance of students such as culture, family background, psychological history, previous academic performance, parents' economic situation, and previous schooling. In this paper, data mining technique was used to predict students' performance based on their interaction on a LMS. The LMS used is the Modular Object Oriented Dynamic Learning Environment (MOODLE). Datasets containing students' activities on MOODLE were used for the study. The base classifiers used for comparison are: Decision Tree (DT), Naïve Bayes (NB) and K-Nearest Neighbor (KNN). Waikato Environment for Knowledge Analysis (WEKA) was used for data preprocessing, attributes selection evaluation, result analysis and 10-fold cross validation. The results obtained indicates that DT is the best model with 84.1% accuracy which outperforms NB and KNN with accuracies of 83.7% and 76.7% respectively. A correlation analysis showed that the assignment submission attribute was identified as the most significant feature that have the most impact on the prediction of students' performance.

Keywords: Data Mining, E-Learning, LMS

INTRODUCTION

The use of platform independent e-learning systems in educational institutions is growing at an exponential rate (Brusilovsky & Peylo, 2003). Learning Management Systems (LMSs) are common e-Learning systems used in universities, colleges, schools, businesses, and also, by instructors to reduce the traditional face-to-face contact with students (Cole & Foster, 2007). One of the most commonly used LMS is MOODLE, a free learning management system that enables the creation of powerful, flexible and engaging online courses (Rice, 2006). Recent findings revealed that data mining techniques are used by instructors and educational administrators, to improve educational related activities (Cristobal Romero & Ventura, 2007). The field of study that deals with the application of data mining techniques to educational data is referred to as Educational Data Mining (EDM) (Cristóbal Romero & Ventura, 2010). Data mining techniques have been applied to assessment of students' learning performance, providing feedback to both teachers and students of e-learning courses, and the detection of typical students' learning behaviour (Considine & Zappalà, 2002). Prediction of Students' performance is one of the applications of EDM and its objective is to measure the hidden value of students' performance, understanding or grading from other information, attitude or behaviour of students (Cristóbal Romero, López, Luna, & Ventura, 2013). This is a difficult task to address due to the diverse number of variables or attributes that influences the performance of students such as culture, family background, psychological history, previous academic performance, parents economic situation, previous schooling, among others (Anuradha & Velmurugan, 2015). Higher institutions of learning are solving some of their major challenges by adopting data mining techniques. It is essential to note that most recent researches on EDM for students' performance prediction were primarily applied to cases in universities and high school students (Cristóbal Romero et al., 2013) and specifically, in most cases to e-learning or related mode of instruction (Araque, Roldán, & Salguero, 2009).

Several data mining techniques can be used by EDM in building predictive models either to trim down the students failing ratio or to provide recommendations to various stakeholders such as: students, teachers, researchers and administrators, where these recommendations might have a significant impact in improving the learning process (Araque et al., 2009; Patidar, Dangra, & Rawar, 2015).

Classification, which is the most popular data mining technique used (Amrieh, Hamtini, & Aljarah, 2016), is a supervised learning technique that builds a model to classify data items according to a predefined class label (a categorical attribute) based on the values of other attributes (the predicting attributes). The aim of classification is to predict future output based on available data. Classification can be used to predict students' performances in critical courses, such as programming courses in computer science (Al-Barrak & Al-Razgan, 2015). Decision Tree (DT), Naïve Bayes (NB), K-Nearest Neighbor (KNN) and many more are examples of data mining classification algorithms.

Decision Tree (DT)

Decision Tree builds classification model in the form of a tree structure and generate rules (conditional statements) that can easily be understood by humans and easily used within a database to identify a set of records. Most commonly used DT algorithms by researchers in EDM are: Iterative Dichotomiser3 (ID3), Classification and Regression Trees (CART) and C4.5 Algorithms.

ID3 is a recursive algorithm that employs a top down greedy search through the space of possible branches with no backtracking. The ID3 simply uses a fixed set of examples to build a Decision Tree, and later the developed DT will be employed to classify new future samples (Quinlan, 1979). The ID3 constructs a decision tree based on information gain/entropy measures (Araque et al., 2009). According to Aliyu, Musa, and Jauro (2018), when datasets are split to grow decision trees, it is done to reduce impurity and entropy is one way to measure the degree of impurity given by the equation 1.

$$Entropy = \sum_j P_j \, log_2 P_j \qquad (1)$$

where P_j is the probability of each class j in the dataset. The information gain on splitting on an attribute say A, is the difference between the degree of impurity of the parent dataset say S and the weighted summation of impurity degrees of the subset dataset S_i split on attribute A with values v mathematically described as:

$$Information\ Gain(S, A) = H(S) - \sum_v |S_v|/|S| \, H(S_v) \qquad (2)$$

CART was introduced with reference to classification and regression trees, so that it can handle categorical and continuous variables in building decision trees. Classification trees are used to identify the class that a categorical target variable would likely fall into. On the other hand, regression trees handle prediction of continuous target variables. CART works through the recursive partitioning of the training set in order to obtain subsets that are as pure as possible to give the target class. Gini Index is the default impurity measure used in CART for categorical target variables. It is essentially a measure of how well the splitting rule separates the classes contained in the parent node (ElGamal, 2013). Given a training dataset S and that the target attribute takes on j different values, then the Gini index of S is defined as:

$$Gini(S) = 1 - \sum_{i=1}^{j} (P_i)^2 \qquad (3)$$

where P_i is the probability of S belonging to the class i. A Gini split measures the divergences between the probability distribution of the target attributes values which is achieved by selecting the attribute with the maximum gain. The gain by a Gini split on a dataset S and attribute A is given as:

$$Gini\ split(S, A) = Gini(S) - \sum_{i=1}^{j} |S_i|/|S| \, Gini(S_i) \qquad (4)$$

Quinlan (1979) developed C4.5, as an extension of ID3 algorithm in order to handle problems associated with ID3. C4.5 generates decision trees using an approach whereby each node splits the classes based on the gain of information. The C4.5 accepts both continuous and discrete features. It can also handle incomplete data points, as well as, over-fitting problem by a clever bottom-up technique usually known as 'pruning'.

DT algorithm was used by Cristóbal Romero, Ventura, and García (2008) to predict students' final marks based on their usage data in the MOODLE system. Real data from seven MOODLE courses was collected to classify students into two groups: pass and fail. The objective of their research was to classify students with equal final marks into different groups based on the activities carried out in a web-based course. Similarly, commonly used decision tree classifiers: C4.5, ID3 and CART were compared by Yadav and Pal (2012) to find the best classifier for predicting student's performance in First Year engineering exam which showed that the true positive rate of the model for the FAIL class was 0.786 for ID3 and C4.5 decision trees which signifies a successful identification of students who are likely to fail.

Naïve Bayes (NB)

The NB algorithm is a machine learning algorithm that is used for solving classification problems. It is called 'Naïve' because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features. It is a probabilistic classifier that is based on the Bayes Theorem (Raschka, 2014) which states that: "The probability of the event A given the event B is equal to the probability of the event B given A multiplied by the probability of A upon the probability of B". The theorem is mathematically expressed as:

$$P(A \mid B) = (P(B \mid A) \times P(A))/P(B) \qquad (5)$$

In any classification problem, there are multiple features and classes. The aim of the Naïve Bayes classifier is to compute the conditional probability of an object with a feature vector say $x_1, x_2, \ldots, x_n$ belonging to a particular class C_i, for $1 \leq i \leq k$ expressed mathematically as:

$$P(C_i|x_1, x_2, \ldots, x_n) = P(x_1, x_2, \ldots, x_n|C_i) \times P(C_i)/P(x_1, x_2, \ldots, x_n) \qquad (6)$$

Equation 6 can be reduced to:

$$P(C_i|x_1, x_2, \ldots, x_n) = \left(\prod_{j=1}^{j=n} P\left(x_j\middle|C_i\right)\right) \times P(C_i)/P(x_1, x_2, \ldots, x_n) \qquad (7)$$

So, Naïve Bayes classifiers are easy to build and useful particularly for very large dataset, along with simplicity.
According to the work of (Pandey & Pal, 2011), NB classifier can help teacher and students reduce drop out ratio and improve performance level of the students and institution. The application of NB classification on students' database was recommended by (Tair & El-Halees, 2012) where NB was used to predict students' grade with an aim to improve their performance before graduation. In their study, data collected includes 3360 records and 18 attributes for a period of fifteen years (1993 to 2007) and results obtained showed that NB classifier has an acceptable accuracy of 67.5% in predicting graduate students' grades.

K-Nearest Neighbor (KNN)
KNN is a non-parametric classification algorithm used to predict a target label by finding the nearest neighbor classes (Wang, Li, Wang, Liu, & Zhang, 2018). K in KNN refers to the number of nearest neighbors the classifier will use to make its predictions. KNN stores all available cases and classifies new cases based on a similarity measure of the nearest neighbors. The nearest neighbors are computed using any of the distance measures like Euclidean distance, Manhattan distance, Hamming distance etc.
So, given a dataset and an unknown data point, we compute the distance of the unknown data point from all the points in the dataset. The class of the majority neighbors is then selected for the unknown data.
KNN is usually used when there are non-linear decision boundaries between classes. Prasetyawan and Abadi (2017) designed a classification model for predicting students' graduation using the KNN algorithm based on student status, gender and on their first 2 years performance. Raga Jr and Raga (2017) attempts to determine whether Students' Moodle Action Log Data can be used to predict student course performance through the use of various data mining techniques and his experiment showed that the overall best accuracy rating was obtained using the k-nearest-neighbor algorithm with an average accuracy of 72.8%. Kuznar and Gams (2016) studied the prediction of student performance in an online course with the goal of detecting at an early stage of a course those students who have a high chance of failing using the k-nearest neighbor (KNN) and they reported that KNN was able to predict student performance accurately.

Although, several literatures (Kuznar & Gams, 2016; Pandey & Pal, 2011; Prasetyawan & Abadi, 2017; Raga Jr & Raga, 2017; Cristóbal Romero et al., 2008; Tair & El-Halees, 2012; Yadav & Pal, 2012) have demonstrated the viability of base classifiers in predicting students' performance, but only a few have considered comparing the predictive power of these base classifiers given the same dataset.

The aim of this research is to develop predictive models using DT, NB and KNN for predicting students' performance based on their interaction with an LMS as well as to compare the predictive power of the developed models. Despite all the valuable studies that have been carried out in the area of this study, a gap has been identified in the comparison of predictive models for predicting students' performance developed from DT, NB and KNN based on students' activities on a LMS.

MATERIALS AND METHODS

Data Source and Description
The MOODLE log file used in this study was obtained from Institute of Computing and Information Communication Technology (I.C.I.C.T) Ahmadu Bello University Zaria, Nigeria. The file contains records of 515 students' MOODLE activities. In order to carry out the analysis intended for this research, some relevant features representing students learning behavior were extracted from the log file as described in section 2.2.1. Table 1 illustrate a brief description of the data collected. The steps required to prepare the data is as described in the following subsections starting from the data selection, cleaning, transformation and finally, partitioning.

Table 1: Data Description

	Course_View	*Resource_View*	*Assignt_Submit*	*Assignt_View*	*Forum_View*
count	511	505	513	513	500
mean	12.912	7.402	5.159	31.910	0.412
std	18.334	10.688	2.650	25.317	1.1246
min	0	0	0	0	0
max	106	68	14	181	14

Data Preprocessing

Data collection and preprocessing is concerned with preparing the raw data or cleaning it to obtain formatted data suitable for analysis. The preprocessing tasks include removal of: noise, missing values, incomplete and inconsistent data. The major task involved in data preprocessing include: data cleaning, data integration, data transformation. Fig. 1 shows the data collection tasks used in this study.

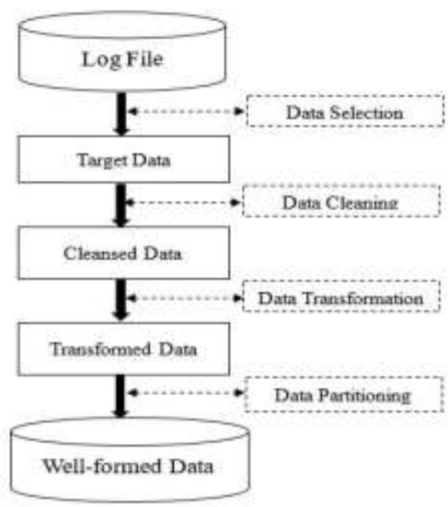


Fig. 1: Data Collection and Preprocessing Tasks

Data Selection

LMS like MOODLE stores huge amount of data in its database, some of which include: administrator's activities, instructor's activities and student's activities. For the purpose of this study, only data related to student's interaction on the e-learning platform was selected because they provide information on the students' participation in the course activities as well as their impact on performance of the students. Table 2 shows the descriptions of the selected features.

Table 2: Attributes Description

S/N	Features	Description
1.	*Course_View*	Number of course views during semester
2.	*Assign_View*	Number of assignment views during semester
3.	*Assign_Submit*	Number of assignment uploads and updates during semester
4.	*Resource_View*	Number of resource views during semester
5.	*Forum_View*	Number of forum views during semester
6.	*Performance*	The students overall score/grade (pass or fail)

Data Cleaning

In this stage, a number of missing values were handled. The missing values were obtained as a result of non-participation of students in some activities. All missing values nominal and numeric attributes were replaced with modes and means from the training datasets using the WEKA filter function called *RaplaceMissingValues*.

Data Transformation

The data transformation technique used in this paper is normalization which is the procedure of bringing data closer to the requirements of the algorithms in order to ease the algorithm's job. WEKA was used to normalize the data using a function called Normalize. It also has a number of built-in functions that support different data transformation tasks making it suitable tool for data normalization.

Data Partitioning

The class label is defined using the final (overall) mark the students obtained in the course. If the total mark is less than 40 (i.e. Total < 40) then the grade is fail, while if it is greater than or equal to 40 (i.e. Total >= 40) then the grade is pass.

Model/Result Validation

A 10-folds cross validation was used to train and validate the developed models. This technique divides the data set into 10 subsets of equal size; nine of the subsets are used for training, while one is left out and used for testing. The process is iterated ten times; the final result is estimated as the average error rate on test samples.

Model Performance Evaluation

The performance of the models was evaluated using a metric called confusion matrix. The matrix is N x N where N is the number of target values (classes), the matrix shows the instances correctly and incorrectly classified by the predictive model compared to the actual outcomes in the data sets which are used to find the evaluation measures: Accuracy, Precision, Recall and F-Measure of the model whose definitions are shown in the equations (8) - (11). Table 3 shows the confusion matrix.

Table 3: Confusion Matrix

	PREDICTED CLASS		
ACTUAL CLASS		POSITIVE	NEGATIVE
	POSITIVE	TP	FP
	NEGATIVE	FN	TN

i. Accuracy: this is the proportion of total number of predictions that were correct. The higher prediction accuracy is, the better the model.

$$\text{Accuracy (A\%)} = (TP + TN)/ (TP + TN + FP + FN) * 100 \qquad (8)$$

ii. Positive predictive value/Precision: this is percentage of instances that the classifiers marked and classified in the class, and are actually in the class.

$$\text{Precision (P \%)} = TP/(TP+FP) * 100 \qquad (9)$$

iii. Recall or Sensitivity: this is the proportion of actual positive cases which are identified correctly.

$$\text{Recall (R \%)} = TP/(TP+FN) * 100 \qquad (10)$$

iv. F-Measure is the accuracy of harmonic mean of precision and recall that is the weighted average of the class.

$$\text{F-Measure (F\%)} = (2*\text{Precision}*\text{Recall})/ (\text{Precision} + \text{Recall}) *100 \qquad (11)$$

Significant Attributes

As one of the objectives of this study was to identify the variables/attributes that are more important which are indicators of high performance of students. The WEKA Attribute Selection Evaluator was used to show the relationship between the features and the class level (Grade).

RESULTS AND DISCUSSION

The experiment was conducted using WEKA using a 10-fold cross validation to train and validate the model after which its performance was measured using the data in the confusion matrix. The results obtained for the evaluation of the base classifiers is as shown in Table 4 and graphically in Fig. 2.

Table 4: Evaluation Measures for base classifiers

Evaluation Measure	Naïve Bayes (NB)	Decision Tree (DT)	Nearest Neighbor (KNN)
Accuracy	83.7	84.1	76.7
Precision	84.5	84.8	85.1
Recall	98.9	98.9	87.8
F-Measure	91.1	91.3	86.4

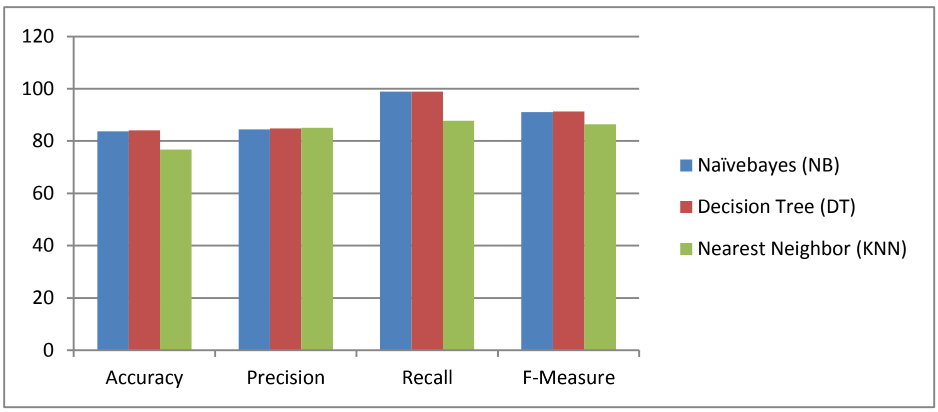


Fig. 2: Evaluation Metrics for base classifiers

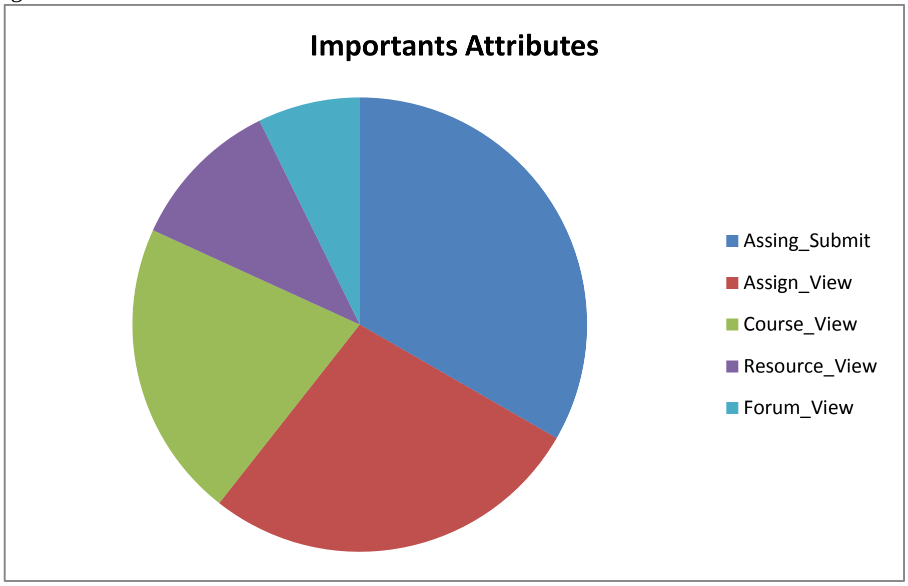
From the results obtained, it is clear that when different algorithms were applied on the dataset, the results are distinct based on the evaluation metrics. Table 4 shows the classification results using the following data mining techniques: NB, KNN and DT. It indicates that DT model outperforms other data mining techniques with 84.1% accuracy which means 433 students were correctly classified to the actual class labels (pass and fail) while 82 students were incorrectly classified followed by the NB with 83.7% accuracy and then KNN with 76.7 % accuracy. The precision, Recall and F-Measure for the three classifiers are shown in Table 4. Fig. 2 is a graphical representation of the evaluation metrics for the base classifiers. The results prove the strong effect of the considered attributes on the prediction of students' performance and the suitability of the DT model for such purpose.

To determine the important features that have the most significant impact on the prediction of the result, the WEKA selection attributes: Correlation Evaluator and the ranker search method was used. The results on Table 5 and Fig. 3 were obtained.

Table 5: Important Features

Features	Relationship	Rank
Assign_Submit	0.55	1
Assign_View	0.45	2
Course_View	0.35	3
Resource_View	0.18	4
Forum_View	0.12	5

Fig. 3: Attributes Correlation Chart



From Table 5, it can be observed that, the attribute *Assign_Submit* has a strong positive correlation with the score, having value of correlation coefficient of 0.55; hence, this implies that students who submit their assignments has high chances of performing better that those who do not. The attributes *Assign_View* and *Course_view* have medium correlations with the score, having values of 0.45, and 0.35 respectively. Finally, the attributes *Resource_View* and *Forum_View* both have weak correlations with the score. Fig. 3 also shows the important attributes represented by the regions in the pie chart.

CONCLUSION

Dataset containing records of students' activities on Moodle was used to train and test students' performance predictive model of some base classifiers: DT, NB and KNN. Accuracy, Precision, Recall and F-Measure were performance measures used to evaluate the performance of the base classifiers. This study clearly concludes that DT is the best classifier among the three algorithms used and therefore, it can be used to predict students' performance prior to examinations. The study, also attempts to find out the most significant attributes that have most impact on students' performance prediction of which Assign_Submit was obtained. Education decision makers should be encouraged to advice students to participate fully in the above attributes as they are indicators of good performance. In the future, we intend to improve the current study through the use of ensemble techniques to improve the performance of the base classifiers or the use of comprehensive datasets with varieties of features in order to obtained more conclusive results or even modifying any of the base classifiers in an attempt to improve their performance.

REFERENCES

Al-Barrak, M. A., & Al-Razgan, M. S. (2015). Predicting Students' performance Through Classification: A Case Study. *Journal of Theoretical & Applied Information Technology, 75*(2).

Aliyu, S., Musa, H., & Jauro, F. (2018). *Performance comparison of two Decision tree algorithms based on splitting criteria for predicting child birth delivery type.* Paper presented at the 1st International Conference on Education and Development, Baze University, Abuja-Nigeria.

Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining Zeducational data to predict Student's academic performance using ensemble methods. *International Journal of Database Theory and Application, 9*(8), 119-136.

Anuradha, C., & Velmurugan, T. (2015). A comparative analysis on the evaluation of classification algorithms in the prediction of students performance. *Indian Journal of Science and Technology, 8*(15).

Araque, F., Roldán, C., & Salguero, A. (2009). Factors influencing university drop out rates. *Computers & Education, 53*(3), 563-574.

Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education (IJAIED), 13*, 159-172.

Cole, J., & Foster, H. (2007). *Using Moodle: Teaching with the popular open source course management system*: " O'Reilly Media, Inc.".

Considine, G., & Zappalà, G. (2002). The influence of social and economic disadvantage in the academic performance of school students in Australia. *Journal of sociology, 38*(2), 129-148.

ElGamal, A. (2013). An educational data mining model for predicting student performance in programming course. *International Journal of Computer Applications, 70*(17), 22-28.

Kuznar, D., & Gams, M. (2016). *Metis: system for early detection and prevention of student failure.* Paper presented at the 6thInternational Workshop on Combinations of Intelligent Methods and Applications (CIMA 2016).

Pandey, U. K., & Pal, S. (2011). Data Mining: A prediction of performer or underperformer using classification. *arXiv preprint arXiv:1104.4163*.

Patidar, P., Dangra, J., & Rawar, M. (2015). Decision tree C4. 5 algorithm and its enhanced approach for educational data mining. *Engineering Universe for Scientific Research and Management, 7*(2).

Prasetyawan, P., & Abadi, F. (2017). *Application Development of Student's Graduation Classification Model based on The First 2 Years Performance using K-Nearest Neighbor.* Paper presented at the International Conference on Engineering and Technology Development (ICETD).

Quinlan, J. R. (1979). *Induction over Large Data Bases*. Retrieved from

Raga Jr, R. C., & Raga, J. D. (2017). Monitoring Class Activity and Predicting Student Performance Using Moodle Action Log Data. *International Journal of Computing Sciences Research, 1*(3), 1-16.

Raschka, S. (2014). Naive bayes and text classification i-introduction and theory. *arXiv preprint arXiv:1410.5329*.

Rice, W. (2006). Moodle e-learning course development. a complete guide to successful learning using moodleQ3. *Packt Publ*, 13.

Romero, C., López, M.-I., Luna, J.-M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. *Computers & Education, 68*, 458-472.

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert systems with applications, 33*(1), 135-146.

Romero, C., & Ventura, S. (2010). 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), 601-618.

Romero, C., Ventura, S., & García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education, 51*(1), 368-384.

Tair, M. M. A., & El-Halees, A. M. (2012). Mining educational data to improve students' performance: a case study. *International Journal of Information, 2*(2), 140-146.

Wang, W., Li, Y., Wang, X., Liu, J., & Zhang, X. (2018). Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers. *Future Generation Computer Systems, 78*, 987-994.

Yadav, S. K., & Pal, S. (2012). Data mining: A prediction for performance improvement of engineering students using classification. *arXiv preprint arXiv:1203.3832*.