International Journal of Engineering and Technology Innovation, vol. 7, no. 1, 2017, pp. 11 - 23

Slow Learner Prediction using Multi-Variate Naïve Bayes Classification Algorithm

Shiwani Rana*, Roopali Garg

Department of Information Technology, UIET, Panjab University, Chandigarh, India.

Received 05 September 2016; received in revised form 21 November 2016; accepted 02 December 2016

Abstract

Machine Learning is a field of computer science that learns from data by studying algorithms and their constructions. In machine learning, for specific inputs, algorithms help to make predictions. Classification is a supervised learning approach, which maps a data item into predefined classes. For predicting slow learners in an institute, a modified Naïve Bayes algorithm implemented. The implementation is carried sing Python. It takes into account a combination of likewise multi-valued attributes. A dataset of the 60 students of BE (Information Technology) Third Semester for the subject of Digital Electronics of University Institute of Engineering and Technology (UIET), Panjab University (PU), Chandigarh, India is taken to carry out the simulations. The analysis is done by choosing most significant forty-eight attributes. The experimental results have shown that the modified Naïve Bayes model has outperformed the Naïve Bayes Classifier in accuracy but requires significant improvement in the terms of elapsed time. By using Modified Naïve Bayes approach, the accuracy is found out to be 71.66% whereas it is calculated 66.66% using existing Naïve Bayes model. Further, a comparison is drawn by using WEKA tool. Here, an accuracy of Naïve Bayes is obtained as 58.33 %.

Keywords: classification, clustering, confusion matrix, multi-variate, naïve Bayes, supervised machine learning, unsupervised machine learning, WEKA tool

1. Introduction

Machine learning is a type of intelligent learning, which provides computers with the ability to design and develop algorithms. It focuses on the advancement of computer programs that can train themselves to grow and change when exposed to new data. A machine-learning program detects patterns in data and includes different combinations of logic [1].The objective of machine learning is to program PCs to utilize case information or information from previous experience to tackle a given issue. Two basic types of machine learning are as follows:

Supervised Machine Learning: Supervised Machine Learning Algorithm generates a function, which maps inputs to desired outputs. These algorithms are trained using labeled examples [3]. Supervised algorithms can be applied on various data sets based on two functionalities: Training and Prediction

Training: During the training phase, a conversion of input value to a feature set is done using feature extractor. Feature sets contain the general information about each input, which can be used in the classification by different machine learning algorithms to generate a model [4].

* Corresponding author. E-mail address: shiwanirana40@gmail.com

Prediction: During the prediction phase, a conversion of unseen input values to a feature set is done using same feature extractor. Further, these feature sets are used as inputs to various models to generate predicted labels.

Unsupervised Machine Learning: Unsupervised machine learning algorithm models a set of inputs that are used for data that has no historical labels. In supervised machine learning, each example contains input objects (a vector) without output values (targets). It analyzes the training *data*. *Then* separates and groups (also known as clustering) *the data* with the similarity metric [5]. The goal of the algorithm is to discover the data and find some arrangement within. Unsupervised machine learning works well on transactional data.

***Benefits of Machine Learning*:**

- *Rapid processing, analysis, and predictions*: The speed at which machine learning can consume data and identify relevant data makes the ability to act in real time a reality [6]. For example, machine learning can constantly optimize the next best offer for the customer. This can be understood as to what the customer might see at noon may be different from what that same customer sees at 1 PM.
- *Huge data inputs from infinite sources*: Machine learning has the capability to consume infinite amounts of detailed data to constantly review and adjust the messages based on very recent customer behaviors. Once a model is trained from a full set of data sources, it can identify the most relevant variables, limiting long and complicated integrations and allowing for focused data feeds [7].
- *Action systems*: The systems can act on the outputs of machine learning and can thus make the marketing message much more dynamic. For example, newly obtained information may suggest surfacing a retention offer to a specific customer. Perhaps, on the other hand, no offer at all, if the behavior suggests that the customer might not require one to create a conversion event.
- *Learning from past behaviors*: A major advantage of machine learning is that models can learn from past predictions and outcomes, and continually improve their predictions based on new and different data. A simple example is whether the weather at a particular moment has a correlated effect on conversion behavior.

***Disadvantages of Machine Learning*:**

- *Limited*: It is not a guarantee that machine-learning algorithms will always work in every case imaginable. Sometimes or most of the times machine learning will fail. Therefore, it requires some understanding of the problem at hand in order to apply the correct machine-learning algorithm.
- *Large data requirements*: Some machine learning algorithms require a large amount of training data. It might be cumbersome to work with or collect such large amounts of data. Fortunately, there is much training data for image recognition purposes.

Naïve Bayes algorithm (NB)

The Naive Bayes algorithm [8] is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. It is a simple method, which uses Bayesian theorem for classification. It was named after Thomas Bayes who proposed Bayes theorem. Bayes Theorem can be written as [9]:

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)} \tag{1}$$

where P (x) is probability of x. P (x|y) is conditional probability of x given y. x, y can be A or B.

It is called naïve because it simplifies the problems relying on two important assumptions. It assumes that the predictive attributes are conditionally independent with known classification. In addition, there is an assumption that there are no hidden attributes.

Naive Bayes algorithm is for classification. It is a type of supervised learning where the class is known for a set of a training data points(already known data sets) and need to propose the class for any other given data points. The complexity for Naïve Bayes Algorithm is O (log n) [10]. The steps of Naïve Bayes algorithm are:

Step 1: Calculate the prior probability based on previous experience, often used to predict outcomes before they actually happen.

$$Prior\ Probability\ of\ specific\ objects = \frac{No.\ of\ specific\ objects}{Total\ No.\ of\ objects} \quad (2)$$

Step 2: After calculating the prior probability, a new object (O) is ready to be classified. For calculating likelihood, assume that the more objects in the vicinity of the new object, the more likely that the new cases belong to that particular class.

$$Likelihood\ of\ A belonging\ to\ O\ particular\ class = \frac{No.\ of\ objects\ of\ a\ particular\ class\ in\ the\ vicinity\ of\ O}{Total\ No.\ of\ objects\ of\ that\ class} \quad (3)$$

Step 3: Using the Bayes Rule, the final classification is obtained by combining both the prior and likelihood probabilities and is known as posterior probability.

$$Posterior\ Probability\ of\ O\ belonging\ to\ a\ specific\ class = Prior\ Probability\ of\ specific\ objects * Likelihood\ of\ O\ belonging\ to\ a\ particular\ class \quad (4)$$

***Advantages of Naïve Bayes*:**

- Naive Bayes can be used for both binary and multiclass classification problems.
- The Naive Bayes algorithm affords fast, highly scalable model building.

***Disadvantages of Naïve Bayes*:**

- This algorithm assumes independence of features but in practice, this assumption rarely holds.

Motivation of the research: The slow learner prediction [11] is the branch of the automatic predictive method for the students learning abilities. The student's performance based slow learner method plays a significant role in nourishment of the skills of the students with slow learning ability to minimize the adverse future effects of the slow learning problem [12]. The early stage detection can help the institutions to identify and evaluate the individual performance of the students and helps to incorporate the special care on the slow learners [13]. To summarize, the motivation of this paper is to show the weakness of a student not only by analyzing the current marks scored by him/her but also the in-depth information, which is helpful in pointing out why is a student a slow learner.

The paper is organized in the following sections. Section II describes the Data Collection and Preparation. Here, the most significant forty-eight different attributes of the students are discussed for an in-depth analysis. Section III deals with the Proposed Work, which gives an overview of the problem with the work done in the research. Section IV presents the Methodology used in the research. The Implementation details and the results are included in Section V. Further, the paper is appended with Conclusion and Future scope in Section VI.

2. Data Collection and Preparation

The data set used for classification is shown in the Table 3.1. It contains 48 attributes (variable_names) and 60 instances (number of students). It is in Attribute-Relation File Format (ARFF) format. In addition to some specific information (like marks in different subjects), other detailed information is also collected through the survey. Further, other necessary information is collected in order to improve the prediction accuracy [14]. The prediction will be more accurate with the detailed information about a student.

Table 1 Student related Data variables

Variable-name (Attribute)		
MinorI	MinorII	Final_Exam
Assign (assignment)	Grades	Quiz
Total_Marks	Gender	Family_System
F_Profession	M_Profession	P_AnnualIncome
(Father's Profession)	(Mother's Profession)	(Parent's annual income)
Parental_Status	Medium_12th	SchoolType_12th
Institution_10th	Mode_of_transportation_to_school	Category
BoardType_11th	Institution_11th	SchoolType_10th
Medium_10th	CGPA_10th	BoardType_10th
SchoolType_11th	Medium_11th	Percentage_11th
Percentage_12th	BoardType_12th	Institution_12th
Private_Tutions	Area_at_school_level	Computer_at_home
Having_net_access	MathsMarks_11th	MathsMarks_12th
Given_any_enterance_exam	If_yes_name	Entrance_Exam_rank
(if given any entrance exam)		
Entrance_exam_year	Mode_of_admission_in_University	CGPA_1stYear
Interest_in_sports	After_graduation	decisionlogic

2.1. ARFF

ARFF stands for Attribute-Relation File Format. It is fundamentally an ASCII (American Standard Code for Information Interchange) content document, which is utilized for depicting a rundown of occasions sharing an arrangement of qualities. ARFF files have two distinct sections: Header Information and Data Information.

The Header of the ARFF file format further contains:

- Relation name
- Attributes (the columns in the data) and their data-types.

The @relation Declaration: The relation name is defined as the first line in the ARFF file. The format of Relation contained in Header is:

@relation <relation-name>

where <relation-name> is a string. The string must be quoted if the name includes spaces.

The @attribute Declarations:

Characteristic presentations appear as a requested succession of @attribute declarations. Every property in the information set has its own particular @attribute declaration, which remarkably characterizes the name of that characteristic and its information. The request in which the characters are proclaimed shows the segment position in the information area of the rec ord.

For instance, if an attribute is the third one pronounced then it is normal that every o ne of those character qualities will be found in the third comma-delimited section [15-17]. The format for the @attribute statement is:

@attribute <attribute-name><datatype>

where the <datatype> can be any one of these types:

- numeric
- integer is treated as numeric
- real is treated as numeric
- <nominal-specification>
- string
- date [<date-format>]
- relational for multi-instance data (for future use)

The keywords numeric, real, integer, string and date are case insensitive.

The ARFF Data section of the file contains the data declaration line and the actual instance lines.

The @data Declaration is a single line denoting the start of the data segment in the file. The format is: @data

3. Proposed Work

The existing model is not based upon the multi-attribute based relationship evaluation for the learning capabilities of the students [18]. The existing model is not capable of computing the multi-relationships across the database to derive the multi-directional access to the hidden data in the submitted dataset. The proposed model can be made capable by adding the multi-attribute based multiple probability computational support for the input data processing. A limited number of factors have been analyzed in the existing model, which limits the performance of the slow learner classification model. The performance of the existing model can be improved by looking at multi-variate data features for the in-depth analysis, which directly affects the depth of the multiple perception analysis modules [19]. The proposed model can utilize the pattern discovery algorithm to discover the un-processed patterns to detect the slow learner students. The proposed system is developed using improved and detailed data mining for the slow learner evaluation using Naïve Bayes Classifier. The proposed system will be capable of performing the deep analysis over the student data obtained from high school, which may contain the information about the user performance in the various feature enhancements [20-21]. The simulations were conducted using the proposed model by employing the dataset of the students of BE (Information Technology) Third Semester for the subject of Digital Electronics at University Institute of Engineering and Technology (UIET), Panjab University (PU), Chandigarh, India. The experimental results show that the modified Naïve Bayes model has outperformed the Naïve Bayes Classifier in accuracy but requires significant improvement in the terms of elapsed time.

4. Methodology

4.1. Modified Naïve Bayes:

The proposed Naïve Bayes model structure has been defined with the following attributes, as listed below, for the processing of the student data. It has been designed using the python-programming interface [22].

- Each data row is processed with the singular dimension and is defined with the feature vector of the database values F = (v1, v2, v3 …. vn). These are further processed for the multiple averaging based results, which are utilized to compute the final classification decisions.
- For the processing of the multi-class data (such as C1, C2 … Cm), the unclassified data is utilized for the testing. The prediction is done after analysing the multi-directional analysis of the input database of the student dataset based upon the multiple averaging factors.

4.2. Algorithm: Naive Bayes Classifier for Student learning prediction

- Obtain the pre-defined classified information from the input dataset in the form of the pre-classification defined with "yes" and "no" probabilities, which are also known as the Pyes and Pno values obtained from the training data [23].
- Then, the iteration is run for each of the test record in the student dataset
- Iterate till each of the test record
    - Iterate for each and every attribute
        - Categorize each and every attribute
        - Calculate the primary result types computed with the following equation:
            - $Result(y) = Resul\_types * Probability\ of\ the\ yes\ attribute$ (5)
            - $Result(n) = Resul\_types * Probability\ of\ the\ no\ attribute$ (6)
        - Calculate the overall results of positive samples with Result(y) = Result(y) * P(y)
        - Calculate the overall results of negative samples with Result(y)= Result(y) * P(n)
        - Compute the probabilities in the combinations of the attributes
        - Normalize the probability values by computing the singular averaging factor
        - Update the classification vector
- Compute and return the final decision vector for each of the entry in the dataset
- Compute the performance models based upon the precision, recall, and other performance measures
- Return the program

5. Implementation and Results

The modified Naïve Bayes Classification algorithm is applied to the data set of UIET students. Slow learners are to be classified from the given data set. The python coding for the naïve bayes is done on Ubuntu platform.

The decision logic, that is, the actual labels are taken to be :

'yes' - for slow learner
'no'- for not being slow learner

After the assignment of labels, the classification is being processed which gives the prediction by compairing the actual and predicted labels.

Actual labels are to be predicted by the user.Predicted labels are to be predicted by the classification algorithm.

The existing model finds the individual probability but the modified model finds the probability in combination.

Performance Metrics

The confusion matrixis used to measure the performance of two-class problem for the given data set. The confusion matrix consists of Correctly Classified Instances and Incorrectly Classified Instances. The Correctly classified instances are composed of TP and TN. The Incorrectly classified instances are composed of FP and FN.

According to these design requirements, the specialized joints and members include cam pairs, gear pairs, the frame, the input, and the output. Their symbols and representations are listed in Table 1. The whole process proceeds according to the follows:

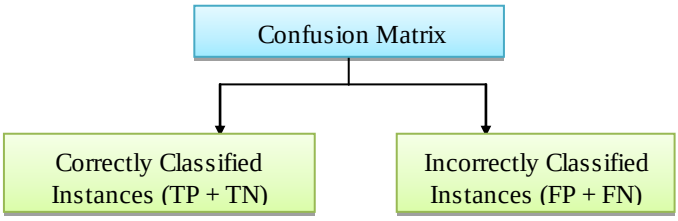

Fig. 1 Confusion Matrix

- TP Rate**:** These are the cases in which this research predicted 'yes', students are slow learners. The algorithm predicts as slow learners as well.

$$TP = \frac{TP}{TP + FN} \tag{7}$$

- FP Rate: These are the cases in which this research predicted 'yes', students are slow learners; but they are predicted as not being slow learners by the Naïve Bayes classifier. These types of cases are also known as 'Type I error' [24].

$$FP = \frac{FP}{FP + TN} \tag{8}$$

- TN Rate: This represents those cases in which the research predicted that the students are not slow learners. The algorithm also predicts these students as not being slow learners.

$$TN = \frac{TN}{TN + FP} \tag{9}$$

- FN Rate: These are the cases in which the research predicted 'no' i.e. the students are not slow learners; but they are predicted as slow learners by the algorithm. These types of cases are known as 'Type II error' [25].

$$FN = \frac{FN}{FN + TP} \tag{10}$$

The performance metrics values are evaluated and plotted for both Existing Naïve Bayes and Modified Naïve Bayes algorithm. The performance metrics under consideration are- Confusion Matrix, Precision, Recall, F-Measure and Accuracy.

5.1. Confusion Matrix Instances

Fig. 2 shows the difference between correctly classified and incorrectly classified instances over Naïve Bayes and Modified Naïve Bayes model. Naïve Bayes is shown with blue colored line and Modified Naïve Bayes is shown with red colored line. Correctly classified and incorrectly classified are over horizontal line and Instances are shown over the vertical line.

Table 2 Confusion Matrix

Slow Learner Prediction	Predicted: No	Predicted: Yes
Actual: No	TP:35 (Existing Naïve Bayes) :42 (Modified Naïve Bayes)	FN:5 (Existing Naïve Bayes) :2 (Modified Naïve Bayes)
Actual: Yes	FP:15 (Existing Naïve Bayes) :15 (Modified Naïve Bayes)	TN:5 (Existing Naïve Bayes) :1 (Modified Naïve Bayes)

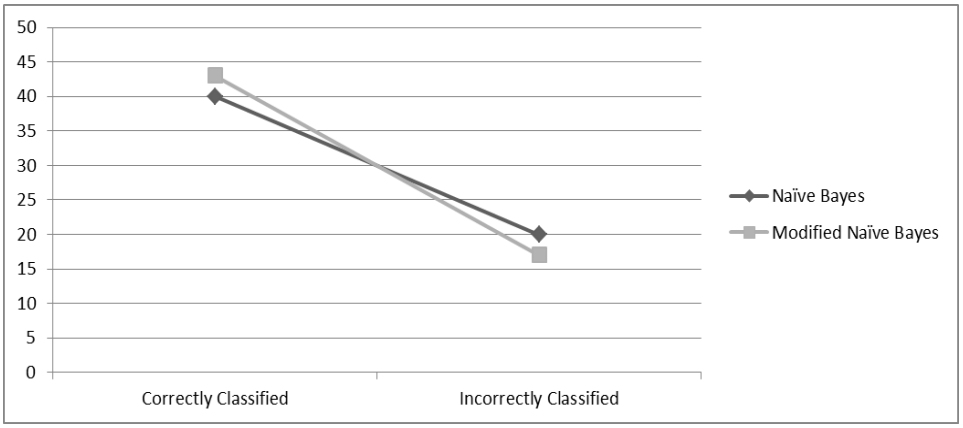

Fig. 2 Comparison based on correctly classified instances and incorrectly classified instances

5.2. *Precision*

Precision explains that 'how many selected items are relevant'

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

where TP: True Positive (Naïve Bayes: 35 - Modified Naïve Bayes: 42)

and FP: False Positive (Naïve Bayes: 15 - Modifies Naïve Bayes: 15)

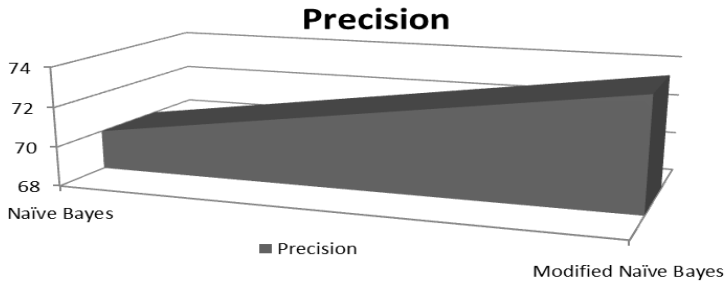

Fig. 3 Comparison based on Precision

Fig. 3 shows the percentage of precision between the two models. Vertical line represents the Percentage, and horizontal line depicts the models. Left hand side of the horizontal line shows the precision percentage of the Naïve Bayes model which is 70% and right hand side shows the precision percentage of Modified Naïve Bayes model, which is 73.68%.

Precision actually defines the closeness of the values to each other.

$$\text{Precision Percentage (Naïve Bayes)} = \frac{35}{35+15} \times 100 = 70\%$$

$$\text{Precision Percentage (Modified Naïve Bayes)} = \frac{42}{42+15} \times 100 = 73.68\%$$

This shows that Modified Naïve Bayes algorithm performs better in terms of Precision i.e. it is efficient in deciding the number of selected item which are relevant.

5.3. *Recall*

Recall gives 'how many relevant items are selected?'

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

where TP: True Positive (Naïve Bayes: 35 - Modified Naïve Bayes: 42)

and FN: False Positive (Naïve Bayes: 5 - Modifies Naïve Bayes: 2)

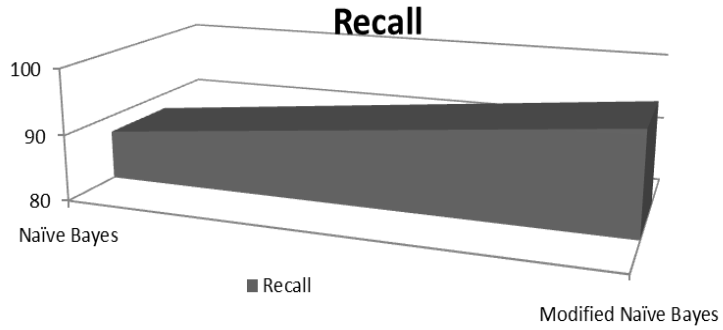

Fig. 4 Comparison based on Recall

Fig. 4 shows the percentage of Recall between the two models. Vertical line represents the Percentage, and horizontal line depicts the models. Left hand side of the horizontal line shows the recall percentage of the Naïve Bayes model which is 87.5% and right hand side shows the recall percentage of Modified Naïve Bayes model, which is 95.45%.

$$\text{Recall Percentage (Naïve Bayes)} = \frac{35}{35+5} \times 100 = 87.5\%$$

$$\text{Recall Percentage (Modified Naïve Bayes)} = \frac{42}{42+2} \times 100 = 95.45\%$$

Here again, the Modified Naïve Bayes algorithm performs better in terms of Recall i.e. number of relevant items that can be selected.

5.4. F-Measure

F-Measure conveys the balance between the precision and the recall.

$$F - Measure = \frac{2\ x\ P\ x\ R}{P+R} \tag{13}$$

where P: Precision (Naïve Bayes: 70% - Modified Naïve Bayes: 73.68%)
and R: Recall (Naïve Bayes: 87.5% - Modified Naïve Bayes: 95.45%)

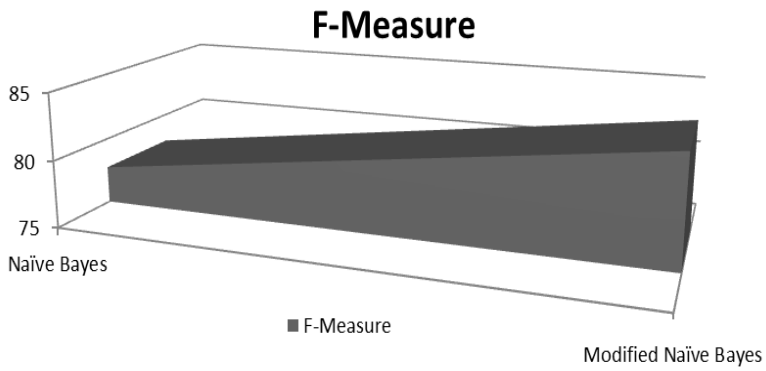

Fig. 5 Comparison based on F-Measure

Fig. 5 shows the percentage of F-Measure between the two models. Vertical line represents the Percentage, and horizontal line depicts the models. Left hand side of the horizontal line shows the F-Measure percentage of the Naïve Bayes model which is 77.77% and right hand side shows the F-Measure percentage of Modified Naïve Bayes model, which is 83.16%. They can be obtained by using Eq. (13), i.e.:

$$\text{F-Measure Percentage (Naïve Bayes)} = \frac{2 \times 70 \times 87.5}{70+87.5} = 77.77\%$$

$$\text{F-Measure Percentage (Modified Naïve Bayes)} = \frac{2 \times 73.68 \times 95.45}{73.68+95.45} = 83.16\%$$

5.5. Accuracy

Accuracy is defined as the percentage of proportion of the total number of predictions that are correct [26].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Fig. 6 shows the percentage of Accuracy between the two models. Vertical line represents the Percentage, and horizontal line depicts the models. Left hand side of the horizontal line shows the Accuracy percentage of the Naïve Bayes model which is 66.66% and right hand side shows the Accuracy percentage of Modified Naïve Bayes model, which is 71.66%. Both are found by using Eq. (14), i.e.:

$$\text{Accuracy Percentage (Naïve Bayes)} = \frac{35+5}{35+5+15+5} \times 100 = 66.66\%$$

$$\text{Accuracy Percentage (Modified Naïve Bayes)} = \frac{42+1}{42+1+15+2} \times 100 = 71.66\%$$

The existing model finds the individual probability but the modified model finds the probability in combination.

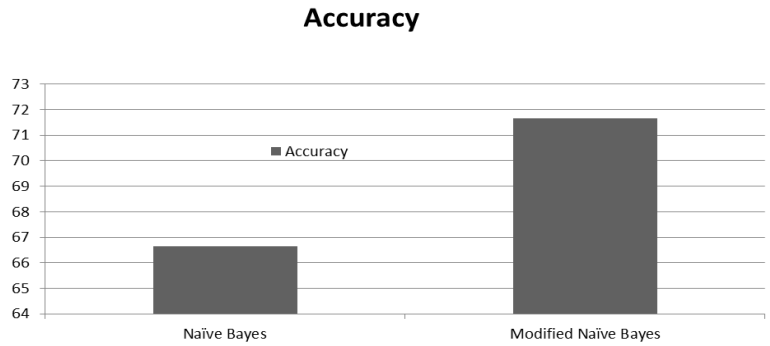

Fig. 6 Comparison based on Accuracy

Comparison of Naïve Bayes and Modified Naïve Bayes classification algorithms is presented in Table 3. The comparison is drawn using features like- Correctly classified instances, incorrectly classified instances, Precision, Recall and F-Measure [27].

Table 3 Comparison of the Existing and Naïve Bayes Modified Model

Type of classification Algorithm	Correctly Classified Instances (TP + TN)	Incorrectly Classified Instances (FP + FN)	Precision	Recall	F-Measure	Accuracy
Naïve Bayes	40	20	70%	87.5%	77.77%	66.66%
Modified Naïve Bayes	43	17	73.68%	95.45%	83.16%	71.66%
Naïve Bayes (WEKA)	35	25	80%	60.9%	69.15%	58.33%

Table 3 presents the comparison of existing and Modified Naïve Bayes model on the data set of 60 students of BE, third semester at UIET, Panjab University, Chandigarh for the subject of Digital Electronics. The main attributes were grades, marks in sessional and total marks without the in-depth information. The correctly classified instances are calculated as 40, and the rest 20 are classified incorrectly [28]. This means 40 students are those who's actual and predicted labels same and 20 students are those whose predicted labels do not match with the actual labels and same is the case for modified Naïve Bayes model. However, in the modified model, the in-depth attributes with the main attributes are considered. Here, a more accurate result is obtained. In this case, the correctly classified instances and incorrectly classified instances are 43 and 17, respectively, instead of 40 and 20 as obtained earlier.

Further, the table shows the prediction results of 60 students using Naïve Bayes algorithm but by employing WEKA tool [29]. The correctly classified instances and incorrectly classified instances are calculated as 35 and 25, respectively. This means 35 students are those who's actual and predicted labels are same, and 25 students are those whose predicted labels do not match with the actual labels. In addition, all the performance metric values are lower in this case.

Fig. 7 justifies the values of Naïve Bayes Classifier using WEKA in Table 3. In this figure, 'no' means the instances/students are not slow learners.

```
Correctly Classified Instances          35               58.3333 %
Incorrectly Classified Instances        25               41.6667 %
Kappa statistic                          0.0854
Mean absolute error                      0.4526
Root mean squared error                  0.578
Relative absolute error                124.2565 %
Root relative squared error            136.0607 %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.5       0.391       0.28      0.5       0.359      0.491    yes
                 0.609     0.5         0.8       0.609     0.691      0.491    no
Weighted Avg.    0.583     0.475       0.679     0.583     0.614      0.491

=== Confusion Matrix ===

  a  b   <-- classified as
  7  7 |  a = yes
 18 28 |  b = no
```

Fig. 7 Naïve Bayes Classifier Output using WEKA

6. Conclusion and Future Scope

In this research, a Modified Naïve Bayes classification algorithm is developed in Python. It is utilized for the prediction on the dataset of 60 students of BE (Information Technology) Third Semester of UIET, PU, Chandigarh, for predicting and examining the performance of students and even analyzing the slow learners among them. In this study, a model was created taking into account some chosen student related information variables gathered from a survey (real world data). This study is extremely helpful to recognize the proportion of moderate learners, to rectify the disappointments at an early stage and initiate steps to enhance the weaker students in a better way. A modified Naïve Bayes classification model is then compared with the previous Naïve Bayes by using the same data set of 60 students with their detailed information including 48 attributes. A comparison of the existing and modified Naïve Bayes algorithm is done based on five main parameters such as Precision, Recall, F-Measure, Confusion Matrix and Accuracy. Modified Naïve Bayes algorithm works by combining the likewise attributes and then finding the probability. By using this approach, the accuracy is obtained to be 71.66% whereas it is 66.66% for existing Naïve Bayes model. By using WEKA tool, the accuracy is found to be 58.33%.

Therefore, the modified algorithm is actually developed to predict the slow learners among the students not only by their marks or grades but also by using additional personal information. One can find the nature of a student by this additional information and can accurately predict the performance. Further, the corrective steps can be taken well in time.

In future, the research can be extended to include more number of students with more data about each student (i.e. profile and educational modules). The present study has taken into account the performance of students for the subject of Digital Electronics. The analysis can be carried out for other subjects, like microprocessor, Algorithm Analysis. In addition, subjects having Digital Electronics as its pre-requisite can lead to interesting research outcomes. The proposed algorithm can be compared with some of the improved naive Bayes algorithms such as tree augmented naive Bayes (TAN), hidden naive Bayes (HNB), Averaged One-Dependence Estimators (AODE), Weighted Average of One-Dependence Estimators (WAODE).

Acknowledgments

The authors thank Panjab University for carrying out the research work. A special appreciation for BE (IT) students of 3rd sem, UIET for their constant support in providing the data needed for the investigation. Deep regards are due for anonymous Reviewers for their valuable feedback and suggestions, which has led to refining the research work further.

References

[1] S. Singh and S. P. Lal, "Educational courseware evaluation using machine learning techniques," Proc. IEEE Conference on e-Learning, e-Management and e-Services (IC3e 13), IEEE Press, Dec. 2013, pp. 73-78.

[2] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," Science, vol. 349, pp. 255-260, July 2015.

[3] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of machine learning, London: MIT Press, 2012.

[4] H. Bydovska and L. Popelínský, "Predicting student performance in higher education," Proc. IEEE Workshop on Database and Expert Systems Applications (DEXA 13), IEEE Press, Aug. 2013, pp. 141-145.

[5] C. Anuradha and T. Velmurugan, "A data mining based survey on student performance evaluation system," Proc. IEEE International Conference on Computational Intelligence and Computing Research (ICCIC 14), IEEE Press, Dec. 2014, pp. 452-456.

[6] K. Koile, A. Rubin, S. Chapman, M. Kliman, and L. Ko, "Using machine analysis to make elementary students' mathematical thinking visible," Proc. International Conference on Learning Analytics & Knowledge (LAK 16), ACM, Apr. 2016, pp. 524-525.

[7] B. M. McLaren, et al., "Using machine learning techniques to analyze and support mediation of student e-discussions," Frontiers in Artificial Intelligence and Applications, vol. 158, pp. 331-338, Jun. 2007.

[8] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," Machine Learning, vol. 29, pp. 131-163, Nov. 1997.

[9] L. Jiang, H. Zhang, Z. Cai, and D. Wang, "Weighted average of one-dependence estimators," Journal of Experimental and Theoretical Artificial Intelligence, vol. 24, pp. 219-230, Jun. 2012.

[10] L. Jiang, H. Zhang, and Z. Cai, "A novel Bayes model: Hidden Naive Bayes," IEEE Transactions on Knowledge and Data Engineering, vol. 21, pp. 1361-1371, Oct. 2009.

[11] A. B. E. D. Ahmed and I. S. Elaraby, "Data mining: a prediction for student's performance using classification method," World Journal of Computer Application and Technology, vol. 2, pp. 43-47, Feb. 2014.

[12] A. Peña-Ayala, "Educational data mining: a survey and a data mining-based analysis of recent works," Expert Systems with Applications, vol. 41, pp. 1432-1462, Mar. 2014.

[13] M. I. Lopez, J. M. Luna, C. Romero, and S. Ventura, "Classification via clustering for predicting final marks based on student participation in forums," International Conference on Educational Data Mining (EDM 12), Jun. 2012, pp. 148-151.

[14] M. Mayilvaganan and D. Kalpanadevi, "Comparison of classification techniques for predicting the performance of student's academic environment," Proc. IEEE International Conference Communication and Network Technologies (ICCNT 14), IEEE Press, Dec. 2014, pp. 113-118.

[15] J. Willems, "Using Learning styles data to informe-learning design: a study comparing undergraduates, postgraduates and e-educators," Australasian Journal of Educational Technology, vol. 27, pp. 863- 880, Jan. 2011.

[16] M. Peng, et al., "Central topic model for event-oriented topics mining in microblog stream," Proc. International Conference on Information and Knowledge Management (CIKM 15), ACM, Oct. 2015, pp. 1611-1620.

[17] J. Huang, et al., "A probabilistic method for emerging topic tracking in microblog stream," World Wide Web Internet and Web Information Systems, vol. 23, pp. 1-26, Apr. 2016.

[18] S. Rana and R. Garg, "Evaluation of student's performance of an institute using clustering algorithms," International Journal of Applied Engineering Research, vol. 11, pp. 3605-3609, May 2016.

[19] S. Singh and V. Kumar, "Performance analysis of engineering students for recruitment using classification data mining techniques," International Journal of Computer Science and Engineering Technology, vol. 3, pp. 31-37, Feb. 2013.

[20] R. Sison and M. Shimura, "Student modeling and machine learning," International Journal of Artificial Intelligence in Education, vol.9, pp. 128-158, July 1998.

[21] H. Lakkaraju, et al., "A machine learning framework to identify students at risk of adverse academic outcomes," Proc. International Conference on Knowledge Discovery and Data Mining (KDD 15), ACM, Aug. 2015, pp. 1909-1918.

[22] G. Kaur and N. Oberoi, "Naive Bayes classifier with modified smoothing techniques for better spam classification," International Journal of Computer Science and Mobile Computing, vol. 3, pp. 869-878, Oct. 2014.

[23] S. Sivakumari, R. P. Priyadarsini, and P. Amudha, "Accuracy evaluation of C4.5 and Naive Bayes classifiers using attribute ranking method," International Journal of Computational Intelligence Systems, vol. 2, pp. 60-68, Mar. 2009.

[24] P. Meedech, N. Iam-On, and T. Boongoen, "Prediction of student dropout using personal profile and data mining approach," Intelligent and Evolutionary Systems, vol. 5, pp. 143-155, May 2016.

[25] S. Kotsiantis, C. Pierrakeas, and P. Pintelas, "Predicting students' performance in distance learning using machine learning techniques," Applied Artificial Intelligence, vol. 18, pp. 411-426, May 2004.

[26] C. G. Nespereira, E. Elhariri, N. El-Bendary, A. F. Vilas, and R. P. D. Redondo, "Machine learning based classification approach for predicting student's performance in blended learning," Proc. International Conference on Advanced Intelligent System and Informatics (AISI 15), Springer, Nov. 2015, pp. 47-56.

[27] C. Romero, M. I. López, J. M. Luna, and S. Ventura, "Predicting students' final performance from participation in on-line discussion forums," Computers and Education, vol. 68, pp. 458-472, Oct. 2013.

[28] G. I. Webb, J. R. Boughton, and Z. Wang, "Not so Naive Bayes: aggregating one-dependence estimators," Machine learning, vol. 58, pp. 5-24, Jan. 2005.

[29] R. Kohavi, "Scaling up the accuracy of Naïve Bayes classifiers: a decision-tree hybrid," Proc. International Conference on Knowledge Discovery and Data Mining (KDD 96), ACM, Aug. 1996, pp. 202-207.