Advances in Computational Research
ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 7, Issue 1, 2015, pp.-239-242.
Available online at http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BPJ0000187

APPLYING NAIVE BAYESIAN CLASSIFIER FOR PREDICTING PERFORMANCE OF A STUDENT USING WEKA

TRIBHUVAN A.P.[1]*, TRIBHUVAN P.P.[2] AND GADE J.G.[3]

[1]Department of Computer Science, Marathwada Institute of Technology, Aurangabad- 431 028, MS, India.
[2]Deogiri Institute of Engineering and Management Studies, Aurangabad- 431 005, MS, India.
[3]Department of Industrial Automation, J.E.S. College, Jalna - 431 203, MS, India
*Corresponding Author: Email- amrapaliprakash512@gmail.com

Received: December 18, 2014; Revised: January 05, 2015; Accepted: January 15, 2015

Abstract- The advances in computing and information storage have provided vast amounts of data. The challenge has been to extract knowledge from this raw data that has guide to new methods and techniques such as data mining that can link the knowledge gap. This paper aimed to review these new data mining techniques and predicting the performance of a student is a great concern to the higher education managements, where quite a few factors affect the performance. The scope of this paper is to explore the accuracy of data mining techniques. We collected records of 100 under graduate students from a private Educational Institution conducting various Under Graduate courses of Information Technology. Decision tree and Naive bayes algorithms were evaluated by using WEKA tool to discover the performance. Decision tree algorithm is more accurate than the Naive bayes algorithm. This work will help the Educational Institution to precisely predict the performance of the students.

Keywords- Naive bayes, Classification, Decision Tree, Data Mining

Citation: Tribhuvan A.P., Tribhuvan P.P. and Gade J.G. (2015) Applying Naive Bayesian Classifier for Predicting Performance of a Student Using WEKA. Advances in Computational Research, ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 7, Issue 1, pp.-239-242.

Introduction

Data mining software applications includes various techniques that have been developed by both business and research centers. These techniques have been used for industrial, education, commercial and scientific purposes. In real world, predicting the performance of the students is a challenging task. The primary goal of Data Mining in practice tends to be Prediction and Description [1]. Predicting performance of student involves variables like attendance of student, aptitude, assignment, submission, class test marks, GPA, grade etc. in the student test record database.

Data mining involves many different algorithms to accomplished different tasks. All of these algorithms attempt to fit model to the data and examine the data and determine a model that is closest to the characteristics of the data being examined. The model that is created can be either predictive or descriptive in nature. A productive model makes a prediction about values of data using known results found from different data. Predictive model data mining tasks include classification, regression, time series analysis and prediction. Classification maps data into predefined groups or classes [2]. Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known). “How is the derived model presented?” The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks. A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions [3].

The main aim of this paper is to use data mining methodologies to study students’ performance in the courses. Data mining provides many tasks that could be used to study the student performance. Here the classification tasks is used to evaluate student’s performance and as there are many approaches that are used for data classification, the decision tree and Naive Bayes method are used [4]. Decision trees can easily be converted to classification rules Decision tree algorithms, such as ID3 (Iterative Dichotomiser), C4.5, and CART (Classification and Regression Trees) [3]. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attribute [3]. This paper explores the accuracy of Decision tree and Naive Bayes techniques for predicting student performance.

Proposed System

This section describes about the procedure followed to collect and analyze the student data. After the preprocessing on the training data set we apply the data mining techniques to predict the performance of student.

Data Mining Tool

We have selected the WEKA tool. We then applied the detailed methodology suggested by [5] to identify a number of computational, functional, usability, and support criteria necessary for this project.. A variety of formats: WEKA's ARFF format, CSV format, C4.5 format, or serialized Instances format. We select ARFF format here. Practically, WEKA tool supports to build a broad range of algorithms and also supports for very large data sets, so we decided to use WEKA tool.

Training Dataset

The first step in this project is to collect data. It is important to select the most suitable attributes which influence the student performance. We have training set of 100 under graduate students from a private Educational Institution conducting various Under Graduate courses of Information Technology. For each semester the students have to produce 2 home assignments, attend 2 internal tests, weekly aptitude tests and must have attendance above 75% along with attribute GPA of previous semester marks is also calculated and used to appear in the Final Semester Examination.

Preprocessing

In preprocessing on available data relevant classes are formed and cleaned. Information get for each attribute is calculated. Information get with respect to set examples is the expected reduction in entropy that results from opening a set of examples using the values of that attribute. This is used in constructing the Decision tree.

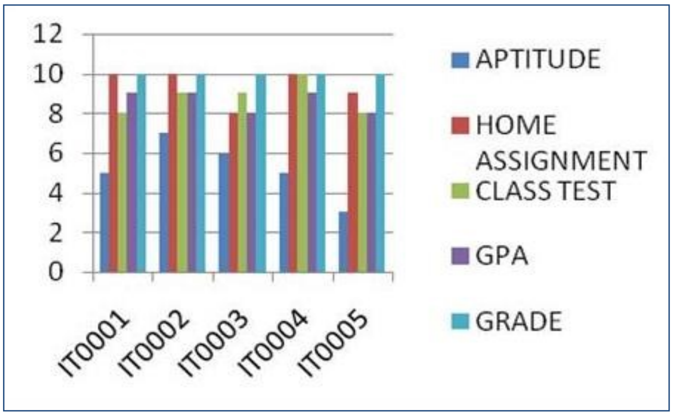

Fig. 1- Sample of Visualization

By using the preprocessing technique visualization, we can get some knowledge about data.

Bayesian Classification

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem, described below. Studies comparing classification algorithms have found a simple Bayesian classifier known as the *Naive Bayesian classifier* to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called *class conditional independence*. It is made to simplify the computations involved and, in this sense, is considered "naive." *Bayesian belief networks* are graphical models, which unlike Naive Bayesian classifiers, allow the representation of dependencies among subsets of attributes.Bayesian belief networks can also be used for classification [3].

Decision Tree

A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules. A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. There are many other methods for constructing classification models, such as naive Bayesian classification, support vector machines, and *k*-nearest neighbor classification [3].

ID3 Decision Tree

In our implementation it first checks the training data for a non-nominal class, missing values, or any other attribute that is not nominal, because the ID3 algorithm can't handle these. It then makes a copy of the training set (to avoid changing the original data) and calls a method:weka.classifiers.trees.Id3

Naive Bayesian Classifiers

Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called *class conditional independence*. It is made to simplify the computations involved and, in this sense, is considered "naive." *Bayesian belief networks* are graphical models, which unlike Naive Bayesian classifiers allow the representation of dependencies among subsets of attributes. Bayesian belief networks can also be used for classification [3]. Method is: weka.classifiers.bayes.NaiveBayes

Result

A total of 50 records were taken for the analysis. The flat file is used in arff (Attribute-Relation File Format). The [Fig-2] shows the test dataset viewed in ARFF-Viewer of WEKA.

The Result is Split into Several Sections

- Run information. A list of information giving the learning scheme options, relation name, instances, attributes and test mode that were involved in the process.
- Classifier model (full training set). A textual representation of the classification model that was produced on the full training data.
- The results of the chosen test mode are broken down thus.
- Summary. A list of statistics summarizing how accurately the classifier was able to predict the true class of the instances under the chosen test mode.

Some of the Strong Rules Obtained from the Tree are as follows:

Results from Decision Trees using Id3

=== Run information ===

Scheme: weka.classifiers.trees.Id3

Relation: Test_Record

Instances: 50

Attributes: 6

ATTENDENCE
APTITUDE
ASSIGNMENT
TEST
GPA
GRADE

Test mode: evaluate on training data

=== Classifier model (full training set) ===

Id3

GPA = GOOD: GOOD
GPA = AVG
| APTITUDE = GOOD: AVG
| APTITUDE = AVG: AVG
| APTITUDE = POOR: null
GPA = POOR: POOR

Time taken to build model: 0 seconds

=== Evaluation on training set ===

=== Summary ===

Correctly Classified Instances	46	92%
Incorrectly Classified Instances	4	8%
Mean absolute error	0.0711	
Root mean squared error	0.1886	
Relative absolute error	17.6446 %	
Root relative squared error	42.1282 %	
Total Number of Instances	50	

Results from Naive Bayesian Network classifier

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes
Relation: Test_Record
Instances: 50
Attributes: 6

ATTENDENCE
APTITUDE
ASSIGNMENT
TEST
GPA
GRADE

Test mode: evaluate on training data

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class GOOD: Prior probability = 0.53
APTITUDE: Discrete Estimator. Counts = 10 12 8 (Total = 30)
ASSIGNMENT: Discrete Estimator. Counts = 28 1 (Total = 29)
TEST: Discrete Estimator. Counts = 28 1 (Total = 29)
GPA: Discrete Estimator. Counts = 28 1 1 (Total = 30)
GRADE: Discrete Estimator. Counts = 28 1 1 (Total = 30)
Class AVG: Prior probability = 0.26
APTITUDE: Discrete Estimator. Counts = 9 6 1 (Total = 16)
ASSIGNMENT: Discrete Estimator. Counts = 14 1 (Total = 15)
TEST: Discrete Estimator. Counts = 14 1 (Total = 15)
GPA: Discrete Estimator. Counts = 1 14 1 (Total = 16)
GRADE: Discrete Estimator. Counts = 1 14 1 (Total = 16)
Class POOR: Prior probability = 0.21
APTITUDE: Discrete Estimator. Counts = 6 1 6 (Total = 13)
ASSIGNMENT: Discrete Estimator. Counts = 5 7 (Total = 12)
TEST: Discrete Estimator. Counts = 5 7 (Total = 12)
GPA: Discrete Estimator. Counts = 1 5 7 (Total = 13)
GRADE: Discrete Estimator. Counts = 1 5 7 (Total = 13)

Time taken to build model: 0 seconds

=== Evaluation on training set ===

=== Summary ===

Correctly Classified Instances	46	92%
Incorrectly Classified Instances	4	8%
Mean absolute error	0.0564	
Root mean squared error	0.2253	
Relative absolute error	13.99%	
Root relative squared error	50.3395%	
Total Number of Instances	50	

ARFF-Viewer - D:\weka-3-5-3\weka-3-5-3\data\Test_Record.arff

File Edit View

Test_Record.arff

Relation: Test_Record

No.	RollNo Nominal	ATTENDENCE Nominal	APTITUDE Nominal	ASSIGNMENT Nominal	TEST Nominal	GPA Nominal	GRADE Nominal
1	IT0001	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
2	IT0002	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
3	IT0003	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
4	IT0004	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
5	IT0005	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
6	IT0006	AVG	AVG	YES	PASS	AVG	GOOD
7	IT0007	POOR	GOOD	YES	PASS	AVG	GOOD
8	IT0008	AVG	GOOD	YES	PASS	AVG	GOOD
9	IT0009	AVG	GOOD	YES	PASS	AVG	GOOD
10	IT0010	POOR	POOR	NO	FAIL	POOR	FAIL
11	IT0011	POOR	POOR	NO	FAIL	POOR	FAIL
12	IT0012	AVG	AVG	YES	PASS	AVG	GOOD
13	IT0013	GOOD	GOOD	YES	PASS	GOOD	EXCELLENT
14	IT0014	GOOD	GOOD	YES	PASS	GOOD	EXCELLENT
15	IT0015	GOOD	GOOD	YES	PASS	GOOD	EXCELLENT
16	IT0016	GOOD	GOOD	YES	PASS	GOOD	EXCELLENT
17	IT0017	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
18	IT0018	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
19	IT0019	GOOD	AVG	YES	PASS	GOOD	EXCELLENT
20	IT0020	GOOD	POOR	YES	PASS	GOOD	EXCELLENT
21	IT0021	GOOD	POOR	YES	PASS	GOOD	EXCELLENT
22	IT0022	GOOD	POOR	YES	PASS	GOOD	EXCELLENT
23	IT0023	GOOD	POOR	YES	PASS	GOOD	EXCELLENT
24	IT0024	GOOD	POOR	YES	PASS	GOOD	EXCELLENT
25	IT0025	POOR	POOR	NO	FAIL	POOR	FAIL
26	IT0026	AVG	GOOD	YES	PASS	AVG	GOOD
27	IT0027	POOR	GOOD	NO	FAIL	POOR	FAIL
28	IT0028	GOOD	GOOD	YES	PASS	GOOD	EXCELLENT
29	IT0029	GOOD	GOOD	YES	PASS	GOOD	EXCELLENT
30	IT0030	GOOD	GOOD	YES	PASS	GOOD	EXCELLENT

Fig. 2- Test Dataset viewed in ARFF-Viewer of WEKA.

Conclusion

Predicting student performance can be useful to the managements in many environments. For identifying good students for admissions, and also those who are appear in the Final Examination.

From the results it is proven that ID3 algorithm is most appropriate for predicting student performance. The error rate is high for Naive bayes classifier. ID3 gives 92% prediction for 50 instances which is relatively higher than Naive Bayes classifier. This study is an attempt to use classification algorithms for predicting the student performance and comparing the performance of ID3 and Naive Bayes classifier.

Conflicts of Interest: None declared.

References

[1] Hand D.J., Mannila H. & Smyth P. (2001) *Principles of data mining*, MIT press.

[2] Dunham M.H. (2006) *Data mining: Introductory and advanced topics*, Pearson Education India.

[3] Jiawei H. & Kamber M. (2001) *Data mining: concepts and techniques*, San Francisco, CA, itd: Morgan Kaufmann, 5.

[4] Nithyasri B., Nandhini K. & Chandra E. (2010) *International Journal on Computer Science and Engineering*, 2(5), 1679-1684.

[5] Collier K., Carey B., Sautter D. & Marjaniemi C. (1999) *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences*, 11.