CLASSIFICATION TECHNIQES IN EDUCATION DOMAIN

[1]B.Nithyasri [2]K.Nandhini [3]Dr. E.Chandra

[1.]M.Phil Scholar, Department of Computer Science, Dr.N.G.P.Arts and Science College, Coimbatore -48, Tamil Nadu, India
[2]. Ph.D Research Scholar& Head, Department of Computer Science, Dr.N.G.P.Arts and Science, Coimbatore -48, Tamil Nadu, India
[3]. Director, Department of Computer Applications, D,J.Academy for Managerial Excellence, Coimbatore-32, Tamil Nadu, India

***Abstract* - Predicting the performance of a student is a great concern to the higher education managements, where several factors affect the performance. The scope of this paper is to investigate the accuracy of data mining techniques in such an environment. The first step of the study is to gather student's data on technical, analytical, communicational and problem solving abilities. We collected records of 200 Post graduate students of computer science course, from a private Educational Institution conducting various Under Graduate and Post Graduate courses. The second step is to clean the data and choose the relevant attributes. Attributes were classified into two groups "Demographic Attributes" and "Performance Attributes". In the third step, Decision tree and Naive bayes algorithms were constructed and their performances were evaluated. The study revealed that the Decision tree algorithm is more accurate than the Naïve bayes algorithm. This work will help the institute to accurately predict the performance of the students.**

Index Terms: *Naive Bayes, Decision Tree, Data Pruning, Data Mining.*

I. INTRODUCTION

In real world, predicting the performance of the students is a challenging task. The primary goals of Data Mining in practice tend to be Prediction and Description [1]. Predicting performance involves variables like GPA, Entrance Marks, etc. in the student database to predict the unknown or future values of interest. Description focuses on finding human interpretable patterns describing the data, for example; identifying the exceptional students for scholarship and identifying the weak students who are likely to fail.

Educational Data Mining uses many techniques such as Decision Trees, Neural Networks, Naïve Bayes, K- Nearest neighbor, Support Vector Machines and many others. Using these methods many kinds of knowledge can be discovered.

The main objective 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. In this research, the classification task is used to evaluate student's performance and as there are many approaches that are used for data classification, the decision tree and Naïve Bayes method are used here.

Information's like GPA, Entrance Marks were collected from the existing database. Attendance, Aptitude test and Assignment marks were collected from the student's management system, to predict the performance at the end of the semester. This paper investigates the accuracy of Decision tree and Naïve Bayes techniques for predicting student performance.

II. PROPOSED MODEL

This section describes about the process followed to collect and analyze the student data. We then preprocess the data and apply the data mining techniques to discover the performance.

A.TOOL

Various data mining tool were compared to select a suitable platform for our study. We began with a list of data mining tools, from which we have selected the WEKA tool. We then applied the detailed methodology suggested by [2] to identify a number of computational, functional, usability, and support criteria necessary for this project**.** Functionally, WEKA tool supports to build a wider range of Algorithms and also supports for very large data sets, so we decided to use WEKA tool.

B. DATA

The first phase of the study is to collect the data. It is important to select the most appropriate attributes which influence the student performance. For the purpose of the study PG students of "computer science course" provided

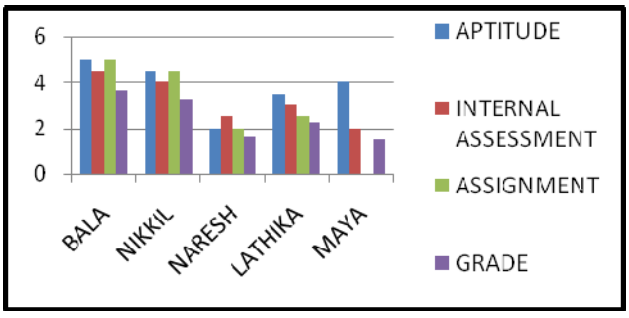
the training set. A total of 200 student's records have been collected. For each semester the students have to produce 5 written assignments, attend 3 internal tests, 5 aptitude tests and must have attendance above 75%. The average assignment marks and aptitude test marks should be >3, should submit at least 3 assignments and attend at least 3 aptitude tests and 2 internal tests to appear in the Final Examination. Along with the above attributes, the cumulative GPA of previous semester marks is also calculated and used.

The attributes were classified into two groups: The "***Demographic Attributes***" group and The "***Performance Attributes***" group. Demographic Attributes group represents the attributes collected from the institute's student's record. The performance attributes like Attendance, Aptitude test, internal test and Assignment marks are obtained from the student's management system. Aptitude test pattern is designed in such a way to test abilities like communication, analytical and logical level of a student. Internal and assignment marks of the students will determine the subject knowledge of a particular semester.

C.*PREPROCESSING THE DATA*

In this phase from the available data relevant groups are formed and cleaned. Information gain for each attribute is calculated. [3] Information gain with respect to set examples is the expected reduction in entropy that results from splitting 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.

D.BAYESIAN NETWORKS

General Bayesian network classifiers are known as Bayesian networks, belief networks or causal probabilistic networks. [4,5]. They draw their roots from a branch of probability and statistics known as decision theory[6],which involves the theory of how to minimize risk and loss when making decisions based on uncertain information.

Moreover, given that quite often data cannot be classified with deterministic correct certainty and associated with every classification problem is a risk/loss function that indicates the severity of an incorrect classification, Bayesian learning involves the process of calculating the most probable hypothesis that would correctly classify an object or piece of data, based on Bye's rule.

Some attractive aspects of Bayesian learning include: each training vector can be used to update probability distributions which in turn affect the probability that a given hypothesis is true; provides more flexibility in that a hypothesis does not get completely ruled out from few examples; and prior knowledge can be easily implemented in the form of prior probability distributions [6].

The structure of a Bayesian network is a graphical illustration of the interactions among the set of variables that it models. It consists of a directed acyclic graph and conditional probability distributions associated with the vertices of the graph. The directed acyclic graph represents the structure of the application domain. Nodes which are usually drawn as circles or ovals represent random variables and arcs represent direct probabilistic dependencies among them. [7, 8]. With every vertex is associated a table of conditional probabilities of the vertex given each state of its parents. We denote the conditional probability table using the notation P (Xi | par(xi)), where lower case xi denotes values of the corresponding random variable Xi and par(xi) denotes a state of the parents of Xi. The graph together with the conditional probability tables define the joint probability distribution contained in the data.

Using the probabilistic chain rule, the joint distribution can be written in the product form:

$$P(x1, x2, x3\ldots) = \prod P(Xi \mid par(xi))$$

Where the product goes from i=1 upto n and n is the number of vertices in the graph.

An example of a simple Bayesian network is given in figure 1. The corresponding joint probability distribution for the figure can be written in the form:

$$P(a, b, c) = P(a \mid b, c)\, P(b \mid c)\, P(c).$$

In a Bayesian network all variables are treated in the same way and any one can be regarded as the class variable Classification. A Bayesian network classifier involves performing probabilistic inference on the Bayesian network using one of the available probabilistic inference algorithms. [9, 10, 11].

For Example,
Let's say we're interested in predicting if a particular student will pass in Math's.

We have data on past student performance. For each student we know:

- If student's GPA > 3.0 (G)
- If student had a strong math background (M)
- If student is a hard worker (H)
- If student passed or failed course

A new student comes along with values G = g, M = m, and H = h, and wants to know if they will likely pass or fail the course.

$$f(g,m,h) = \frac{P(g,m,h,pass)}{P(g,m,h,fail)}$$

If $f(g,m,h) \geq 1$, then classifier predicts pass; otherwise fail.

Table 1 Attribute Categories

Pass				Fail			
GPA>3 (G)	Math? (M)	Hardworker (H)	Prob	GPA>3 (G)	Math? (M)	Hardworker (H)	Prob
0	0	0	0.01	0	0	0	0.28
0	0	1	0.03	0	0	1	0.15
0	1	0	0.05	0	1	0	0.20
0	1	1	0.08	0	1	1	0.14
1	0	0	0.10	1	0	0	0.07
1	0	1	0.28	1	0	1	0.05
1	1	0	0.15	1	1	0	0.08
1	1	1	0.30	1	1	1	0.03

Assume P(pass) = 0.5 and P(fail) = 0.5

Let x = {0,1,0} or {¬G, M, ¬H}

$$f(x) = \frac{P(pass)P(x/pass)}{P(fail)P(x/fail)} = \frac{0.5*0.05}{0.5*0.20} = 0.25$$

Joint Probability Distributions grow exponentially with # of features. For binary-valued features, we need O(2p) Joint Probability Distributions for each class. [13]

For the purpose of the study, the performance attributes like Assignment, Aptitude test, internal test and Attendance were taken as the testing area. A total of 200 students records were collected from the student's management system. For testing the applicability of four scale options for making the possible categories of the identified variables were calculated. The categories of all the attributes were given as (Excellent, good, pass and fail).

E.DECISION TREE

Decision tree is a popular supervised learning classifier that does not require any knowledge or parameter setting. Given a training data, we can induce a decision tree. From a decision tree we can easily create rules about the data. Using decision tree, we can easily predict the classification of unseen records.

Decision tree is a hierarchical tree structure that used to classify classes based on a series of questions or rules about the attributes of the class. The attributes of the classes can be any type of variables from binary, nominal, and quantitative values, while the classes must be qualitative type categorical or binary, or ordinal. Given a data of attributes together with its classes, a decision tree produces a sequence of rules or series of questions that can be used to recognize the class.

F. MEASURING IMPURITY

Given a data table that contains attributes and class of the attributes, we can measure homogeneity (or heterogeneity) of the table based on the classes. We say a table is pure or homogenous if it contains only a single class. If a data table contains several classes, then we say that the table is impure or heterogeneous. There are several indices to measure degree of impurity quantitatively. Most well known indices to measure degree of impurity are entropy, gini index, and classification error.

$$Entropy = \sum_j - p_j \log_2 p_j$$

$$Gini\ Index = 1 - \sum_j p_j^2$$

$$Classification\ Error = 1 - \max\{p_j\}$$

All above formulas contain values of probability of a class j.

One way to measure impurity degree is using entropy.

$$Entropy = \sum_j - p_j \log_2 p_j$$

Entropy of a pure table (consist of single class) is zero because the probability is 1 and log (1) = 0.Entropy reaches maximum value when all classes in the table have equal probability.

Another way to measure impurity degree is using Gini index.

$$Gini\ Index = 1 - \sum_j p_j^2$$

Gini index of a pure table consist of single class is zero because the probability is 1 and 1-(1) ^2 = 0. Similar to Entropy, Gini index also reaches maximum value when all classes in the table have equal probability.

Still another way to measure impurity degree is using index of classification error

$$Classification\ Error = 1 - \max\{p_j\}$$

Similar to Entropy and Gini Index, Classification error index of a pure table (consist of single class) is zero because the probability is 1 and 1-max (1) = 0. The value of classification error index is always between 0 and 1. In fact the maximum Gini index for a given number of classes is always equal to the maximum of classification error index because for a number of classes n, we set probability is equal to p=1/n and maximum Gini index happens at 1-n*(1/n)^2 = 1-1/n, while maximum classification error index also happens at 1-max{1/n}=1-1/n.

To determine the best attribute for a particular node in the tree we use the measure called *Information Gain.* The information gain, *Gain (S, A)* of an attribute *A*, relative to a collection of examples S, is defined as

$$Gain(S.A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Where *Values (A)* is the set of all possible values for attribute *A*, and S_v is the subset of *S* for which attribute *A* has value *v* (i.e., S_v = {s ∈ S | *A(s)* = *v*}). The first term in the equation for *Gain* is just the entropy of the original collection *S* and the second term is the expected value of the entropy after S is partitioned using attribute *A*. The expected entropy described by this second term is simply the sum of the entropies of each subset S_v, weighted by the fraction of examples $|S_v|/|S|$ that belong to S_v. *Gain (S, A)* is therefore the expected reduction in entropy caused by knowing the value of attribute *A*.

The process of selecting a new attribute and partitioning the training examples is now repeated for each non terminal descendant node. Attributes that have been incorporated higher in the tree are excluded, so that any given attribute can appear at most once along any path through the tree. This process continues for each new leaf node until either of two conditions is met:
1. Every attribute has already been included along this path through the tree, or
2. The training examples associated with this leaf node all have the same target attribute value (i.e., their entropy is zero).

Decision tree induction is a typical inductive approach to learn knowledge on classification. In this study we use ID3 Algorithm for constructing the Decision tree. J. Ross Quinlan originally developed ID3 at the University of Sydney. ID3 is based on the Concept Learning System (CLS) algorithm.

ID3 Decision Tree Algorithm

```
function ID3
Input: (R: a set of non-target attributes,
C: the target attribute,
S: a training set) returns a decision tree;
begin
If S is empty, return a single node with value
Failure;
If S consists of records all with the same value
      for the target attribute, return a single
      leaf node with that value;
If R is empty, then return a single node with the
      value of the most frequent values of the
      target attribute that are found in  records
      of S; [in that case there may be errors,
      examples that will be improperly classified];
Let A be the attribute with largest Gain (A, S)
      among attributes in R;
Let {aj| j=1,2, .., m} be the values of attribute
      A;
Let {Sj| j=1,2, .., m} be the subsets of S
      consisting respectively of records with value
      aj for A;
Return a tree with root labeled A and arcs labeled
      a1, a2.. am going respectively to the trees
      (ID3(R-{A}, C, S1), ID3(R-{A}, C, S2),.....,
      ID3(R-{A}, C, Sm);
Recursively apply ID3 to subsets {Sj| j=1,2,  .,
      m}until they are empty
end
```

ID3 searches through the attributes of the training instances and extracts the attribute that best separates the given examples. If the attribute perfectly classifies the training sets then ID3 stops; otherwise it recursively operates on the m (where m = number of possible values of an attribute) partitioned subsets to get their "best" attribute.

III.RESULTS

A total of 50 records were taken for the analysis. The Table 2 shows the test dataset.

Table 2 Test Set

ID	ASSIGNMENT	APTITUDE	ATTENDENCE	GPA	TEST	GRADE
A001	YES	AVG	AVG	GOOD	PASS	PASS
A002	YES	POOR	AVG	GOOD	PASS	PASS
A003	YES	POOR	POOR	GOOD	PASS	PASS
A004	YES	POOR	POOR	POOR	FAIL	FAIL
A005	NO	GOOD	AVG	GOOD	PASS	FAIL
A006	YES	GOOD	GOOD	GOOD	EXCELLENT	EXCELLENT
A007	YES	GOOD	AVG	GOOD	EXCELLENT	EXCELLENT
A008	YES	AVG	AVG	GOOD	PASS	GOOD
A009	YES	AVG	AVG	POOR	EXCELLENT	GOOD
A010	YES	AVG	GOOD	GOOD	PASS	PASS
A011	YES	POOR	POOR	GOOD	EXCELLENT	FAIL
A012	YES	POOR	POOR	POOR	PASS	PASS
A013	YES	POOR	AVG	POOR	PASS	PASS
A014	NO	POOR	POOR	GOOD	EXCELLENT	FAIL
A015	YES	POOR	AVG	POOR	PASS	PASS
A016	NO	POOR	POOR	GOOD	EXCELLENT	FAIL
A017	YES	POOR	AVG	POOR	PASS	PASS
A018	NO	AVG	GOOD	POOR	PASS	PASS
A019	NO	AVG	GOOD	POOR	FAIL	FAIL
A020	YES	AVG	POOR	GOOD	PASS	GOOD
A021	YES	AVG	AVG	POOR	EXCELLENT	GOOD
A022	YES	AVG	GOOD	GOOD	PASS	PASS
A023	YES	POOR	POOR	GOOD	EXCELLENT	FAIL
A024	YES	AVG	AVG	POOR	PASS	PASS
A025	YES	POOR	AVG	GOOD	PASS	PASS
A026	YES	GOOD	AVG	GOOD	EXCELLENT	EXCELLENT
A027	YES	AVG	AVG	GOOD	PASS	GOOD
A028	YES	AVG	AVG	POOR	EXCELLENT	GOOD
A029	YES	AVG	GOOD	GOOD	PASS	PASS
A030	YES	POOR	POOR	GOOD	EXCELLENT	FAIL

Some of the strong rules obtained from the tree are as follows:

TEST = PASS

| APTITUDE = AVG

| | ATTENDENCE = AVG
| | | GPA = GOOD: GOOD
| | | GPA = POOR: PASS
| | ATTENDENCE = POOR: GOOD
| | ATTENDENCE = GOOD: PASS
| APTITUDE = POOR: PASS
| APTITUDE = GOOD: FAIL
TEST = FAIL: FAIL
TEST = EXCELLENT
| APTITUDE = AVG: GOOD
| APTITUDE = POOR: FAIL
| APTITUDE = GOOD: EXCELLENT

Results from Decision Trees using Id3

Time taken to build model	: 0 seconds	
Correctly Classified Instances	: 49	98%
Incorrectly Classified Instances	: 1	2%
Mean absolute error	: 0.0133	
Root mean squared error	: 0.0816	
Relative absolute error	: 3.9216 %	
Root relative squared error	: 19.8789 %	
Total Number of Instances	: 50	

Results from Naïve Bayesian Network classifier

Time taken to build model	: 0 seconds	
Correctly Classified Instances	: 47	94%
Incorrectly Classified Instances	: 3	6 %
Mean absolute error	: 0.0966	
Root mean squared error	: 0.1792	
Relative absolute error	: 28.4077 %	
Root relative squared error	: 43.6242 %	
Total Number of Instances	: 50	
Class PASS: Prior probability	: 0.44	
Class FAIL: Prior probability	: 0.26	
Class EXCELLENT: Prior probability	: 0.09	
Class GOOD: Prior probability	: 0.2	

IV.CONCLUSIONS AND FUTURE WORK

Predicting student performance can be useful to the managements in many contexts. For identifying excellent students for scholarship programs, admissions, and also those who are unlikely to graduate. From the results it is proven that ID3 algorithm is most appropriate for predicting student performance. The error rate is very high for Naïve bayes classifier.ID3 gives 98% prediction for 50 instances which is relatively higher than Naïve Bayes classifier.
This study is an attempt to use classification algorithms for predicting the student performance and comparing the performance of ID3 and Naïve Bayes classifier.

For future work study can be diversified for comparing various courses of under graduates and post graduates of a university with huge datasets.

V. REFERENCES

[1] David Hand, Heikki Mannila, Padhraic Smyth ”Principles of Data Mining”
[2] Collier, K., Carey, B., Sautter, D., and Marjaniemi, C., “A methodology for evaluating and selecting data mining software,” in Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, IEEE, 1999.
[3] Nguyen Thai Nghe , Paul Janecek , and Peter Haddawy “A comparative Analysis of Techniques for Predicting Academic Performance” 37th ASEE/IEEE Frontiers in Education Conference.
[4] Pearl J., Probabilistic reasoning in intelligent systems: networks of plausible inference, (Morgan Kaufmann: San Mateo CA, 1988).
[5] F.V. Jensen., An introduction to Bayesian network (London. U.K: University College London Press, 1996).
[6] F.V. Jensen, Bayesian network basics, AISB Quarterly: Vol 94, 1996, 9-22.
[7] Henrion, Max., Some practical issues in constructing belief networks, Proc. 3rd Conf. on Uncertainty in Artificial Intelligence, Elsevier Science Publishing Company, Inc., New York, NY, 1987, 161-173.
[8] Cheng, J., Bell, D.A. and Liu, W., An algorithm for Bayesian belief network construction from data. Proc. 6th International Workshop on Artificial Intelligence and Statistics, Florida, 1997a, 83-90.
[9] MacKay, David J. C., Information theory, inference and learning algorithms (United Kingdom: Cambridge University Press, 2003).
[10] Pearl, Judea, Causality: Models, Reasoning, and Inference (United Kingdom: Cambridge University Press, 2000).
[11] Lauritzen, Steffen L. & David J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems (with discussion). Journal of the Royal Statistical Society, Series B (Methodological), 50(2), 1988. 157-224
[12] Moore, A. (2001) Bayes Nets for Representing and reasoning about uncertainty. Retrieved April 22, 2008,
[13] Web site: http://www.coral-ab.org/~oates/classes/2006/Machine%20Learning/web/bayesnet.pdf

VI. BIOGRAPHIES

B.Nithyasri received her Master Degree from Bharathiar University and currently doing M.Phil Research at Dr.N.G.P.Arts and Science College, Coimbatore. She has attended more than 5 National and International Conferences. Her research interest lies in the area of Educational Data Mining.

K.Nandhini received her B.Sc., from Bharathiar University, Coimbatore in 1996 and received M.C.A from Bharathidasan University,Tricy in 2001. She obtained her M.Phil., in the area of Data Mining from Bharathidasan University,Tricy in 2004. At present she is working as a Head,for the Department of Computer Science at Dr.N.G.P.Arts and Science College, Coimbatore. She has presented more than 6 research papers in National and

International conferences in the area of Data Mining. Her research interest lies in the area of Data Mining and Artificial Intelligence.

Dr.E.Chandra received her B.Sc., from Bharathiar University, Coimbatore in 1992 and received M.Sc., from Avinashilingam University, Coimbatore in 1994. She obtained her M.Phil., in the area of Neural Networks from Bharathiar University, in 1999. She obtained her PhD degree in the area of Speech recognition system from Alagappa University Karikudi in 2007 . At present she is working as a Head and Assistant Professor at Department of Computer Applications in D. J. Academy for Managerial Excellence, Coimbatore. She has published more than 20 research papers in National, International journals and conferences. She guided for more than 30 M.Phil., research scholars and guiding for more than 10 Ph.D Research Scholars. Her research interest lies in the area of Data Mining, Artificial intelligence, neural networks, speech recognition systems and fuzzy logics.She is an active member of CSI, Society of Statistics and Computer Applications.