International Journal of Information Technology and Management

Vol. X, Issue No. XV, May-2016, ISSN 2249-4510

AN INTERNATIONALLY INDEXED PEER REVIEWED & REFEREED JOURNAL

STUDY ON IMPLICATIONS OF KNOWLEDGE EXTRACTION TO PREDICTION OF STUDENT PERFORMANCE

www.ignited.in

International Journal of Information Technology and Management
Vol. X, Issue No. XV, May-2016, ISSN 2249-4510

Study on Implications of Knowledge Extraction to Prediction of Student Performance

Shweta Singh[1] Dr. Pankaj Kumar[2]

[1]Research Scholar, Shridhar University, Rajasthan

[2]Asst. Prof. (Reader), Dept. Of Computer Science & Engineering, Shri Ram College of Engineering & Management (SRMCEM), Lucknow

Abstract – Identifying variables that predict student's performance may help educators. These variables are influenced by various factors. The study engages factors like students' mathematical background, programming aptitude, problem solving skills, gender, prior experience, high school mathematics grade, locality, previous composters programming experience, and e learning usage.

Keywords: Students, Education, Predict, Performance, Successfully, Control, Applications

----------X----------

1. INTRODUCTION

The Students attributes' data is used to extract knowledge. The automated knowledge extraction model developed through three phases: data preprocessing, attribute selection, and rule extraction. The students' data includes 10 predictive attributes and one target attribute. The predictive attributes are Student ID, High school mathematics grade, mathematical background, problem solving, and programming aptitude, and prior experience, previous computer programming experience, gender, locality, and e learning usage. The target attribute is the Grade (student performance in programming course). Sample of students' dataset is shown in table 1.

Table 1: Sample of students' dataset

➢ Data Preprocessing:

The values of attributes in real world databases may take different shapes, normal values, unique value, single value, almost one value, missing values, continuous values and multi values. The attributes which have normal values contain valuable information and are important in knowledge acquisition. The unique value attributes have a different value for every single record or nearly every record. These attributes identify each record exactly and do not have predictive value. The one value attributes (unary-valued) do not contain information that helps to distinguish between the different records. Therefore they should be ignored for data mining purposes. If a given database includes continuous attributes (real values) then the search based on all possible conjunction values for extracting rules yields to a burdensome computation and consumes much time. This problem can be solved by using a fuzzification process which leads to the reduction of search space. The fuzzy subset of the universe of discourse is described by a membership function (V): [(Tukiainen, Mönkkönen, 2002). Which represents the degree to which and belongs to the set V. A fuzzy linguistic variable, V, is an attribute whose domain contains linguistic values, which are labeled for the fuzzy subsets (Vassilios, Vassilis, 2001). Therefore, the continuous attributes can be transformed into linguistic terms such as; Short (S), Medium (M) and Long (L). A non-overlapping rectangular membership functions may be used and the bounds of each linguistic term can be determined by using the smooth histogram of real values (Wang 2002). The preprocessing stage performs fuzzification process for High school mathematics grade, mathematical background, problem solving, and programming aptitude into nominal values.

➢ Rule Extraction:

Decision trees are a classic method of inductive inference that is still very popular. They are not only easy to implement and use for classification and regression tasks, but also good predictive performance, computational efficiency (Sebastian, (2012). The construction of a decision tree is based on splitting internal nodes recursively. The selection of split attributes on internal nodes is extremely important during the construction process and determines to a large extent the final structure of the decision tree. Many efforts have been made on this aspect and a set of split criteria, such as the Gini index, the information gain and the Chi-Square test, are available. Entropy theory is adopted to select the

appropriate split attribute by the well-known C4.5 algorithm. Let N be the size of the dataset D and Nj the number of samples in class j. Assuming that there are K class labels, the entropy theory states that the average amount of information needed to classify a sample is as follows:

$$Info(D) = \sum_{j=1}^{k} \frac{N_j}{N} \log_2\left(\frac{N_j}{N}\right)$$

When the dataset D is split into several subsets D_1, D2…D_n according to the outcomes of attributes X, the information gain is defined as:

$$Gain(X, D)\ Info(D) - \sum_{j=1}^{k} \frac{N_j}{N} Info(D_j)$$

Where Ni is the number of samples in subset Di ID3 favors all attributes with the largest gain. C4.5 applies Gain ratio, instead of Gain, as following:

$$Gain\text{-}ratio(X, D) = Gain(X, D) \Big/ \left(-\sum_{i=1}^{n} \left(\frac{N^i}{N}\right)\right)$$

C4.5 greedily partitions nodes until a trivial value of the Gain ratio is achieved. A prune procedure is then performed in order to avoid generating a complex tree that over fits the data (Warren, Evangelos 2007. Affendey, *et. al.,* 2010).

A part of Decision tree for the students' dataset is shown in figure 1.

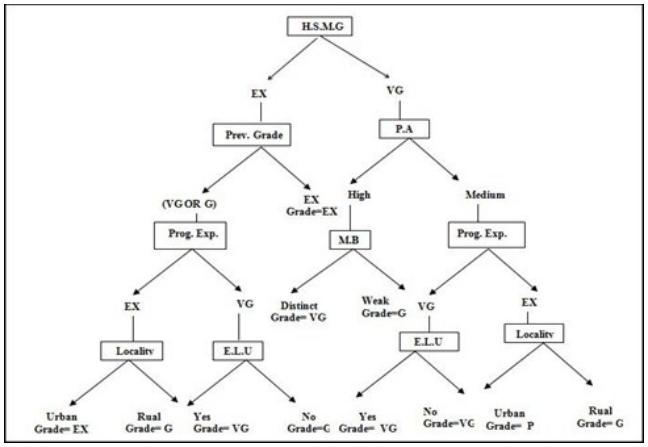

Fig 1: A Part of Decision Tree

2. REVIEW OF LITERATURE:

Most students believe that computer programming is difficult. High failure rates were reported among in introductory programming courses poses an important challenge for freshmen students of many programs. These courses require students to devote a sustained effort during the whole course and a failure to do so may students taking programming courses (Farrell, 2006. Dehnadi, 2009. Kolikant, Pollack, 2002). Although several factors that affect learning to program have been identified over the years, we are still far from a full understanding of why some students learn to program easily and quickly while others flounder. The most frequently mentioned factor is previous computer programming experience. Self-efficacy for programming is influenced by previous programming experience, and student self-efficacy increases substantially during an introductory programming course. Furthermore, students' mental models of programming influence their self-efficacy, and both the mental model and self-efficacy have a direct effect on overall success in an introductory course.(Gerald, *et. al.*). other factors that may affect course success have been less well investigated such as relationship of mathematics or science background to computer programming success. A relationship between student learning styles and learning to program has been found (Doane, William, (2008). There is a body of research on the student's mental model of programming in relation to success in specific programming tasks. In summary, there is a substantial literature on factors affecting student performance (Schuyler, 2008). Predicting student performance in a particular course, or even on assessments within a course, is a difficult but useful undertaking. Given such predictions, a professor can help focus student effort on potential problem areas for particular students given their performance in previous courses. Previous studies have identified a number of predictors? Wilfred W.F. Lau, Allan H.K. Yuen examined the effect of a combination of predictors (gender, learning styles, mental models, prior composite academic ability, and medium of instruction) on programming performance. A.T. Chamillard reported the use of statistical analysis techniques to build predictive models. While many of the generated models did not have sufficient predictive power to be useful, the stronger models and other observations from the analysis provide useful insight into the relationships between the various courses. The models presented use only previous course grades as predictor variables Sally Fincher et al. presented a multi-national, multi-institutional study that investigated introductory programming courses. Report presents a study of possible influencing factors that is distinctive in a number of ways. First, it is both multi-institutional and multi-national, with participants from eleven institutions in three countries. . The study was based on four different diagnostic tasks: a spatial visualization task (a standard paper folding test); a behavioral task used to assess the ability to design and sketch a simple map; a second behavioral task used to assess the ability to articulate a search strategy; and an attitudinal task focusing on approaches to learning and studying (a standard study process questionnaire) (Gerald, *et. al.*).

3. KNOWLEDGE EXTRACTION FOR STUDENT PERFORMANCE PREDICTIONS:

The Students attributes' data is used to extract knowledge. The automated knowledge extraction model developed through three phases: data preprocessing, attribute selection, and rule extraction. The students' data includes 10 predictive attributes and one target attribute. The predictive attributes are Student ID, High school mathematics grade, mathematical background, problem solving, and programming aptitude, and prior experience, previous computer programming experience, gender, locality, and e learning usage. The target attributes is the Grade (student performance in programming course) [18, 19].

➢ **Importance of Data set and Class Labels:**

As the first step in our study, in order to have an experiment in student classification, we selected the student and course data of a LON-CAPA course, PHY183 (Physics for Scientists and Engineers I), which was held at MSU in spring semester 2002. Then we extend this study to more courses. This course integrated 12 homework sets including 184 problems. About 261 students used LON-CAPA for this course. Some of the students dropped the course after doing a couple of homework sets, so they do not have any final grades. After removing those students, 227 valid samples remained. We can predict that the error rate in the first class grouping should be higher than the others; because the sample size among the 9-Classes differs considerably. The present classification experiment focuses on the first six extracted students' features based on the PHY183 Spring 2002 class data.

1. Total number of correct answers. (Success rate)

2. Getting the problem right on the first try, vs. those with high number of submissions. (Success at the first try)

3. Total number of attempts before final answer is derived

4. Total time that passed from the first attempt, until the correct solution was demonstrated, regardless of the time spent logged in to the system. Also, the time at which the student got the problem correct relative to the due date. Usually better students get the homework completed earlier.

5. Total time spent on the problem regardless of whether they got the correct answer or not. Total time that passed from the first attempt through subsequent attempts until the last submission was demonstrated.

Participating in the communication mechanisms vs those working alone LON-CAPA provides online interaction both with other students and with the instructor.

4. CLASSIFIERS:

Pattern recognition has a wide variety of applications in many different fields; therefore it is not possible to come up with a single classifier that can give optimal results in each case. The optimal classifier in every case is highly dependent on the problem domain. In practice, one might come across a case where no single classifier can perform at an acceptable level of accuracy. In such cases it would be better to pool the results of different classifiers to achieve the optimal accuracy. Every classifier operates well on different aspects of the training or test feature vector. As a result, assuming appropriate conditions, combining multiple classifiers may improve classification performance when compared with any single classifier (Petridis, Kaburlasos, 2001. Bhardwaj, Pal 2011).

➢ **Non-tree based classifiers:**

We compare some popular non-parametric pattern classifiers and a single parametric pattern classifier according to their error estimates. Six different classifiers over one of the LON-CAPA data sets are compared. The classifiers used include Quadratic Bayesian classifier, 1-nearest neighbor (1-NN), k-nearest neighbor (k-NN), Parzen-window, multi-layer perceptron (MLP), and Decision Tree. These classifiers are some of the most common classifiers used in practical classification problems. After some preprocessing operations were made on the data set, the error rate of each classifier is reported. Finally, to improve performance, a combination of classifiers is presented.

➢ **Combination of Multiple Classifiers (CMC):**

In combining multiple classifiers we seek to improve classification accuracy. There are different ways one can think of combining classifiers:

The simplest way is to find the overall error rate of the classifiers and choose the one which has the lowest error rate for the given data set. This is called an offline CMC. This may not really seem to be a CMC; however, in general, it has a better performance than individual classifiers. The output of this combination will simply be the best performance. The second method, which is called online CMC, uses all the classifiers followed by a vote. The class getting

maximum votes from the individual classifiers will be assigned to the test sample. This method seems, intuitively, to be better than the previous one. However, when we actually tried this on some cases of our data set, the results were not more accurate than the best result from the previous method. Therefore, we changed the rule of majority vote from "getting more than 50% of the votes" to "getting more than 75% of the votes". We then noticed a significant improvement over offline CMC. The actual performance of the individual classifier and online CMC over our data set. (Wang, Liu, 2012. Hämäläinen, Vinni, 2006. - Hegazy, Moselhi, 1994). Suggest a third method, which is called DSC-LA (Dynamic Selection of Classifiers based on the Local Accuracy estimates).

- **Data Representation and Assessment Tools:**

The predict students' final grades based on the features which are extracted from their (and others') homework data. We design, implement, and evaluate a series of pattern classifiers with various parameters in order to compare their performance in a real data set from the LON-CAPA system. This experiment provides an opportunity to study how pattern recognition and classification theory could be put into practice based on the logged data in LON-CAPA. The error rate of the decision rules is tested on one of the LON-CAPA data sets in order to compare the performance accuracy of each experiment (Tukiainen, Mönkkönen, 2002). Results of individual classifiers, and their combination, as well as error estimates, are presented. The problem is whether we can find the good features for classifying students! If so, we would be able to identify a predictor for any individual student after doing a couple of homework sets. With this information, we would be able to help a student use the resources better (Shute, 1991).

5. CONCLUSION:

Data mining can be used in higher education particularly to improve students' performance. We can apply data mining techniques to discover knowledge using association rules and lift metric. Then we used two classification methods which are Rule Induction and Naïve Bayesian classifier is more useful to predict the performance of student. In line with this decision trees are a classic method of inductive inference that is still very popular. In this paper we found that they are not only easy to implement and use for classification and regression tasks, but also good predictive performance, computational efficiency. We clustered the students into groups using K-Means clustering algorithm, we used outlier detection to detect all outliers in the data.

REFERENCES:

Affendey, L.S., I.H.M. Paris, N. Mustapha, M.N. Sulaiman and Z. Muda, (2010). Ranking of influencing factors in predicting student's academic performance. Inform. Technol. J., 9: pp. 832-837.

Al-Radaideh, Q. A., Al-Shawakfa, E. M., & Al-Najjar, M. I. (2006). Mining student data using decision trees. In the Proceedings of the 2006 International Arab Conference on Information Technology (ACIT' 2006).

Bhardwaj B. K. and Pal S. (April 2011). Data Mining: A prediction for performance improvement using classification, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 4.

Carl Farrell , (2006) Predicting (and Creating) Success in CS1, Issues in Information Systems, Volume VII, No. 1, pp. 259-263.

Delavari N, Beikzadeh M. R. (2004). A New Model for Using Data Mining in Higher Educational System, 5th International Conference on Information Technology based Higher Education and Training: ITEHT '04, Istanbul, Turkey.

Doane, William E. J, (2008) "Predicting student performance in introductory computer programming courses", State University of New York at Albany, ProQuest Dissertations and Theses.

Gerald E. Evans and Mark G. Simkin, "What best predicts computer proficiency?, Communications of the ACM, November 3989 Volume 32 Number 11, pp. 1322-1327.

Golding, P. and O. Donaldson, (2006). Predicting academic performance. Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference T1D-21, San Diego, CA. pp. 1-6.

Hämäläinen, W. and M. Vinni, (2006). Comparison of machine learning methods for intelligent tutoring systems. Proceedings of the 8th International Conference on Intelligent Tutoring Systems, Jhongli, Taiwan, pp. 525-534.

Hegazy, T. and Moselhi, O. (1994). Analogy Based Solution to Markup Estimation Problem. Journal of Computing in Civil Engineering, vol 8(1), pp. 72-87.

Hijazi S. T., and Naqvi R. S. M. M. (2006), —Factors affecting student's performance: A Case of Private College, Bangladesh e-Journal of Sociology, Vol. 3, No. 1.

J. H. Wang, Wen-Jeng Liu (April 2012) and IEEE Transon Systems, Man, and Cybernetics, Vol.32, No.2, pp. 230-238.

J. H. Wang, Wen-Jeng Liu and Lian-Da Lin, (April 2002). "Histogram Based Fuzzy Filter For Image Restoration", IEEE Trans On Systems, Man, and Cybernetics, Vol.32, No.2, pp. 230-238.

Markku Tukiainen and Eero Mönkkönen, (June 2002) "Programming aptitude testing as a prediction of learning to program", 14th Workshop of the Psychology of Programming Interest Group, Brunel University.

Saeed Dehnadi, (May 2009). "A Cognitive Study of Learning to Program in Introductory Programming Courses", thesis submitted to Middlesex University.

Sebastian Nowozin, (2012). "Improved Information Gain Estimates for Decision Tree Induction", Appearing in Proceedings of the 29 th International Conference on Machine Learning, Edinburgh, Scotland, UK.

Stanley TenEyck Schuyler, (2008). "Using Problematizing Ability to Predict Student performance In A First Course In Computer Programming", Robert Morris University, Copyright.

T Warren Liao and Evangelos Triantaphyllou, (2007). Recent advances in data mining of enterprise data algorithms and applications, Series on Computers and Operations Research, Vol. 6.

Valerie J. Shute, (1991). "Who is likely to acquire programming skills?", Educational computing research, Vol. 7(1), pp. 1-24.

Vassilios Petridis, Vassilis G. Kaburlasos, (2001). "Clustering and Classification in Structured Data Domains UsingFuzzy Lattice Neurocomputing (FLN)", IEEETransactions on Knowledge and Data Engineering, March/April, Vol. 13, No. 2.

Vassilios Petridis, Vassilis G. Kaburlasos, (2001). "Clustering and Classification in Structured Data Domains Using Fuzzy Lattice Neuro computing (FLN)", IEEE Transactions on Knowledge and Data Engineering, March/April, Vol. 13, No. 2.

Y.B.-D. Kolikant, S. Pollack, (2002). "Improving mathematically oriented programming skills in computer science studies" fie, vol. 1, pp.T1G3-8, 32nd Annual Frontiers in Education (FIE'02).