International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-5 Issue-7, December 2015

Analysing and Improving Student Performance Using Data Mining and Business Intelligence

Karan Napanda, Sujil Shah, Ojas Kharbe, Sindhu Nair

Abstract*— Academic failures among university students have been the subject of concern in higher education community. Students drop out due to poor academic performance as early as in the first year of their university enrolment. Many interested parties' debate and try to find reasons for this poor performance. Consequently, the ability to predict a student's performance could be useful in many ways to stakeholders of higher education institutions. The proposed system puts forward data mining techniques used to identify the significant variables that affects and influences the performance of undergraduate students. Students' demographic and past academic performance data are then used to study the academic pattern. The knowledge is hidden among the educational data set and it is extractable through various data mining techniques. Such knowledge can be extracted from end semester exams, talents, ethics, grasping power, involvement in extracurricular activities, mid term tests and other educational data sets. Data classification algorithms coupled with decision trees assist in such extraction which can further be analyzed to produce semantic rules to predict student's final performance. The system utilizes semantic web technologies such as ontologies and semantic rules to enhance the quality of the educational content and the delivered learning activities to each student. This proposed system generates a type of confidence among the students and teachers. Hence, the system aims to analyse this extracted such data and mine educational data to produce graphical and statistical results which can help in the improvement of student's performance and also give tutors an overview of the proficiency of the student's learning abilities.

***Index Terms*—Data Mining, ID3, Naïve Bayes, Perceptron Learning rule, Student Performance Analysis**

I. INTRODUCTION

Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large datasets ("big data") involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process in the project is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and updating.

Manuscript published on 30 December 2015.
*****Correspondence Author(s)
Karan Napanda, Department of Computer, D.J. Sanghvi College of Engineering, Mumbai, Maharashtra, India.
Sujil Shah, Department of Computer, D.J. Sanghvi College of Engineering, Mumbai, Maharashtra,India .
Ojas Kharbe, Department of Computer, D.J. Sanghvi College of Engineering, Mumbai, Maharashtra,India.
Prof. Sindhu Nair, Department of Computer, D.J. Sanghvi College of Engineering, Mumbai, Maharashtra,India.

Such operations on data can be used for the improvement of student's performance in a university by analysis. Data mining techniques and ANN (Artificial Neural Network) techniques are used to implement the same. A data set is generated for this purpose, consisting of different educational parameters on which the analysis and the decision making takes place. The techniques are implemented on the data and a comparison is made between the different results that are generated. By comparing using parameters such as accuracy and efficiency, the best result generated is selected for display purposes. The reports generated are in the statistical, graphical form as well as text form which gives an overall analysis of the student's performance and the inference system extrapolates and provides personal suggestions and advice. [1]

II. AN OVERVIEW OF DATA MINING & WHY IT CAN BE USED FOR SUCH ANALYSIS

A. *Data Mining*

A lot of students as well as professors face problems in their academic life, be it related to their grades, their course selection, teaching through the right methods, giving the right amount of notes, motivation and tests to prepare students for the worst scenario. The objective of the project is to develop a system that will analyse a student's academic performance using Data Mining Techniques. The system aims to predict future performance of the students using this analysis. It strives to have in itself a decision making capability to suggest and advise students as well as tutors. It aspires to use data mining techniques for such knowledge generation and inference purpose. Also, within the functioning of the system, since the proposed system uses data mining technique and neural networks, it aspires to compare between the methods and selects the best option. Finally, after generating the results and coming to a conclusion it will generate a graphical/statistical report displaying results regarding the same.[2]

B. *How Data Mining Can Be Used For Analysis*

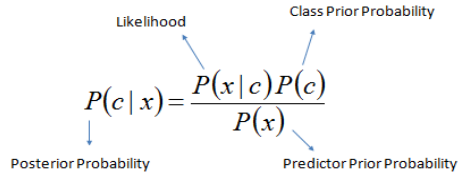
The fact that students these days bother themselves with their grades and especially there is a rising concern among parents and tutors for the downfall in the grades. Also, it is very inconvenient when students resort to very harsh means of treatment on such occasions leading to disastrous consequences. The concept of the project can not only used by the staff and college dean to generate reports and understand patterns but also very useful in a student's life where the ward can make important decision of his life such as selection of optional subjects, which subjects to focus on to improve performance.

Also the concept of the project can be put for large scale implementation for various university courses. To generate patterns of productive output there are different set of algorithms which can be used and thus different conclusions can be computed based on the results generated by Classical Techniques: Statistics, Neighbourhoods and Clustering. Some of these algorithms are Naive Bayesian, ID3 decision making, clustering techniques like k means clustering etc. These techniques will be operated on the data sets and the best solution will be proposed as the result. Also, it will aid in the comparison of the techniques. The proposed system will then make decisions in order as advice and suggestion for the improvement in the student's performance.

III. METHODOLOGIES

A. *Naïve Bayesian*

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. In the statistics and computer science literature, Naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method, Russell and Norvig note that "[Naive Bayes] is sometimes called a Bayesian classifier, a somewhat careless usage that has prompted true Bayesians to call it the idiot Bayes model." Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests. An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- P(c|x) is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

Example: The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target. Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.[3]

B. *ID3*

ID3 is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.

Algorithm

The ID3 algorithm begins with the original set as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set and calculates the entropy (or information gain) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set is then split by the selected attribute (e.g. age < 50, 50 <= age < 100, age >= 100) to produce subsets of the data. The algorithm continues to recur on each subset, considering only attributes never selected before. Recursion on a subset may stop in one of these cases: every element in the subset belongs to the same class (+ or -), then the node is turned into a leaf and labelled with the class of the examples· There are no more attributes to be selected, but the examples still do not belong to the same class (some are + and some are -), then the node is turned into a leaf and labelled with the most common class of the examples in the subset there are no examples in the subset, this happens when no example in the parent set was found to be matching a specific value of the selected attribute, for example if there was no example with age >= 100.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

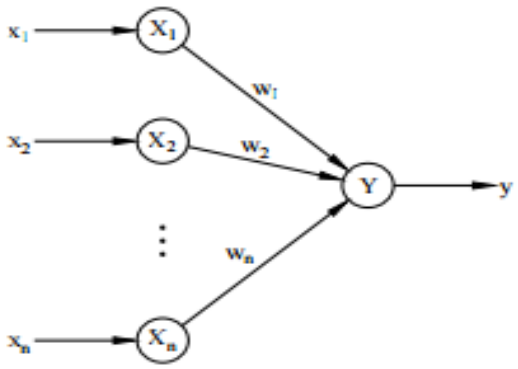
Then a leaf is created, and labelled with the most common class of the examples in the parent set. Throughout the algorithm, the decision tree is constructed with each non-terminal node representing the selected attribute on which the data was split, and terminal nodes representing the class label of the final subset of this branch. **Summary**

1. Calculate the entropy of every attribute using the data set
2. Split the set into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum)
3. Make a decision tree node containing that attribute
4. Recursion on subsets using remaining attributes.[4]

C. Perceptron Learning Rule

In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers: functions that can decide whether an input (represented by a vector of numbers) belongs to one class or another. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm allows for online learning, in that it processes elements in the training set one at a time. A neural network, known as the perceptron, capable of classifying patterns into two or more categories is introduced. Simple Perceptron for Pattern Classification We consider here a NN, known as the Perceptron, which is capable of performing pattern classification into two or more categories. The perceptron is trained using the perceptron learning rule. We will first consider classification into two categories and then the general multiclass classification later. For classification into only two categories, all we need is a single output neuron. Here we will use bipolar neurons. The simplest architecture that could do the job consists of a layer of N input neurons, an output layer with a single output neuron, and no hidden layers. This is the same architecture as we saw before for Hebb learning. However, we will use a different transfer function here for the output neuron:[5]

$$f\theta(x) = \begin{cases} +1, & \text{if } x > \theta, \\ 0, & \text{if } -\theta \le x \le \theta, \\ -1, & \text{if } x < -\theta. \end{cases}$$



IV. PROPOSED SOLUTION

- ***Generating Data Sets***

In this module, we generate a large amount of data set of students on different parameters. The data sets are based on different attributes of a student's academic records (like subjects, attendance, extra- curricular, etc.). The output generated will be a data set on an excel sheet with a .csv extension. The data set will used to extract knowledge in order to be used by the system to apply algorithms on. The data will be extracted from end-semester mark sheets, mid-term tests, ethics and behaviour in class,etc.

- ***Cleaning the Data Set***

This involves elimination of unwanted entries, faulty entries and other faults occurred during the generation of a data set. Such cleaning of data is necessary for it to be processed and worked upon. This results in a data set that is uniform throughout, and actions can be performed with the same. Often, corrupt data set can cause the processing of algorithms to work improperly. Such situations should be avoided and hence refining of the data is necessary for proper processing of data.

- ***Application on the Data Set***

The data set generated in the above steps is then trained upon by using Data Mining and ANN algorithms. Training results in the learning of the system i.e the system learns how to go about a certain set of data and what operations are to be performed. It also builds new attributes/parameters to base its judgements. Extraction is done to gather knowledge by using data mining techniques and ANN techniques to generate knowledge to be analysed. This analysis is made use to produce results

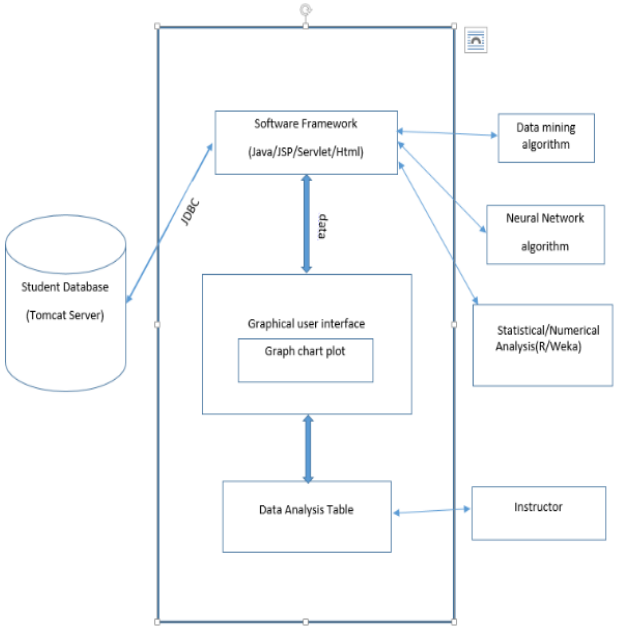
- ***Data Mining Techniques and ANN techniques are applied***

After generation and refining of the data sets, data mining techniques are applied to the generated data set to process information and learn through it. Parallelly, Artificial Neural Network techniques are also applied to generate results and perform analysis. These techniques produce advice and generate statistics to be displayed. The above process trains the system to think but in this module, the systems process the information to map the relation between previous and future courses in order to infer and come to conclusions. These conclusions are either in the graphical or text form.

- ***Report Generation***

The final output is a report generated by the system, by taking a decision. The report is generated by working on the data. It can be a simple course selection advice or advice pertaining to improvement in certain fields of the course. The decision can either be a graph/model which depicts the statistics of a student's performance. Or, it can be a statistical graph depicting relation by making a comparison between all students. The reports can be sent to the respective parents/ guardians as an improvement analysis for the respective student. This will help the student as well as the parents to point out their strengths and weaknesses for certain concepts."

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)



V. CONCLUSION

If this data mining model is applied efficient analysis and usage of the authentic national data can be achieved. In future, this can be used by many admission seekers to analyse and choose the most suitable engineering branch for them. In this work, the data of students from the specific domain was considered. This model can be extended to association rule mining where new interesting hidden patterns can be found from the data. Assuming a student is opting for computer science branch and is given admission in a particular college, it can be investigated whether he has better chances of placement. Finding out the information about dropouts from engineering courses and finding the reasons for it can be a useful future work. Similarly one can investigate whether going for higher studies is giving any extra push for the employment chance of a student. As a summary when data mining techniques are effectively applied over authentic, up-to-date, national level data base, many hidden and useful information can be retrieved, which can be efficiently used both at government and general public level for future planning policies

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” Sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page.

REFERENCES

1. Brijesh Kumar Baradwaj, Saurabh PalMining Educational, Data to Analyze Students‟ Performance, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 6, 2011
2. J. Han and M. Kamber, “Data Mining: Concepts and Techniques,” Morgan Kaufmann, 2000
3. http://www.saedsayad.com/naive_bayesian.htm
4. Mrinal Pandey, Vivek Kumar Sharma, A Decision Tree Algorithm Pertaining to the Student Performance Analysis and Prediction, International Journal of Computer Applications (0975 – 8887) Volume 61– No.13, January 2013.
5. Jia-Lin CHEN and Jyh-Yeong CHANG, Fuzzy Perceptron Learning and Its Application to Classifiers with Numerical Data and Linguistic Knowledge, Neural Networks, 1995. Proceedings., IEEE International Conference

AUTHOR PROFILE

Prof. Sindhu Nair, has a Bachelor’s and a Master’s degree in Computer Engineering. She has several papers which have been published in national and international conferences and publications. She has 12 years of teaching experience in the field of Computer Engineering and 4 years of professional experience in the industry. She is also a member of the ISTE.

Karan Napanda, is a student in the final year of Computer Engineering at D.J. Sanghvi College of Engineering. His interests are in the fields of Data Science, Artificial Intelligence and Machine Learning.

Sujil Shah, is a student in the final year of Computer Engineering at D.J Sanghvi College of Engineering. His interests are in the fields of Artificial Intelligence, Big Data Analytics and Database Management.

Ojas Kharbe, is a student in the final year of Computer Engineering at D.J. Sanghvi College of Engineering. His interests are in the fields of Computer Networks, Computer and Network Security and Big Data.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)