International Journal of Research in Engineering, Science and Management
Volume-2, Issue-1, January-2019
www.ijresm.com | ISSN (Online): 2581-5792

Student's Grade Prediction

Nameerah Kazi[1], Rahul Yadav[2], Ujwala Chinta[3], Deepesh Sharma[4]
[1]*Diploma Student, Department of Computer Engineering, Zagdu Singh Charitable Trust's Thakur Polytechnic, Mumbai, India*

***Abstract*: An educational institution needs to have an approximate prior knowledge of enrolled students to predict their performance in future academics. This helps them to identify promising students and also provides them an opportunity to pay attention to and improve those who would probably get lower grades. As a solution, we have developed a system which can predict the performance of students from their previous performances using concepts of data mining techniques under Classification. We have analyzed the data set containing information about students, such as gender, marks scored in the board examinations of classes X and XII, marks and rank in entrance examinations and results in first year of the previous batch of students.C4.5 classification algorithms on this data, we have predicted the general and individual performance of freshly admitted students in future examinations.**

***Keywords*: Classification, C4.5, Data Mining, Educational Research, Predicting Performance**

1. Introduction

Every year, educational institutes admit students under various courses from different locations, educational background and with varying merit scores in entrance examinations. Moreover, schools and junior colleges may be affiliated to different boards, each board having different subjects in their curricula and also different level of depths in their subjects. Analyzing the past performance of admitted students would provide a better perspective of the probable academic performance of students in the future. This can very well be achieved using the concepts of data mining. For this purpose, we have analysed the data of students enrolled in first year of engineering. This data was obtained from the information provided by the admitted students to the institute. It includes their full name, gender, application ID, scores in board examinations of classes X and XII, scores in entrance examinations, category and admission type.C4.5 algorithms after pruning the dataset to predict the results of these students in their first semester as precisely as possible.

2. Literature survey

A. Data mining

Data mining is the process of discovering interesting knowledge, such as associations, patterns, changes, significant structures and anomalies, from large amounts of data stored in databases or data warehouses or other information repositories. It has been widely used in recent years due to the availability of huge amounts of data in electronic form, and there is a need for turning such data into useful information and knowledge for large applications. These applications are found in fields such as Artificial Intelligence, Machine Learning, Market Analysis, Statistics and Database Systems, Business Management and Decision Support.

1) Classification

Classification is a data mining technique that maps data into predefined groups or classes. It is a supervised learning method which requires labelled training data to generate rules for classifying test data into predetermined groups or classes. It is a two-phase process. The first phase is the learning phase, where the training data is analyzed and classification rules are generated. The next phase is the classification, where test data is classified into classes according to the generated rules. Since classification algorithms require that classes be defined based on data attribute values, we had created an attribute "class" for every student, which can have a value of either "Pass" or "Fail".

2) Clustering

Clustering is the process of grouping a set of elements in such a way that the elements in the same group or cluster are more similar to each other than to those in other groups or clusters. It is a common technique for statistical data analysis used in the fields of pattern recognition, information retrieval, bioinformatics, machine learning and image analysis. Clustering can be achieved by various algorithms that differ about the similarities required between elements of a cluster and how to find the elements of the clusters efficiently. Most algorithms used for clustering try to create clusters with small distances among the cluster elements, intervals, dense areas of the data space or particular statistical distributions.

B. Selecting classification over clustering

In clustering, classes are unknown aprior and are discovered from the data. Since our goal is to predict students' performance into either of the predefined classes - "Pass" and "Fail", clustering is not a suitable choice and so we have used classification algorithms instead of clustering algorithms.

C. Issues regarding classification

1) Missing data

Missing data values cause problems during both the training phase and to the classification process itself. For example, the reason for non-availability of data may be due to:

- Equipment malfunction

- Deletion due to inconsistency with other recorded data
- Non-entry of data due to misunderstanding
- Certain data considered unimportant at the time of entry
- No registration of data or its change

This missing data can be handled using following approaches:

- Data miners can ignore the missing data
- Data miners can replace all missing values with a single global constant
- Data miners can replace a missing value with its feature mean for the given class
- Data miners and domain experts, together, can manually examine samples with missing values and enter a reasonable, probable or expected value

In our case, the chances of getting missing values in the training data are very less. The training data is to be retrieved from the admission records of a particular institute and the attributes considered for the input of classification process are mandatory for each student. The tuple which is found to have missing value for any attribute will be ignored from training set as the missing values cannot be predicted or set to some default value. Considering low chances of the occurrence of missing data, ignoring missing data will not affect the accuracy adversely.

2) Measuring accuracy

Determining which data mining technique is best depends on the interpretation of the problem by users. Usually, the performance of algorithms is examined by evaluating the accuracy of the result. Classification accuracy is calculated by determining the percentage of tuples placed in the correct class. At the same time there may be a cost associated with an incorrect assignment to the wrong class which can be ignored.

3) C4.5

C4.5 is a well-known algorithm used to generate a decision trees. The decision trees generated by the C4.5 algorithm can be used for classification, and for this reason, C4.5 is also referred to as a statistical classifier. The C4.5 algorithm made a number of changes to improve ID3 algorithm . Some of these are:

- Handling training data with missing values of attributes
- Handling differing cost attributes
- Pruning the decision tree after its creation
- Handling attributes with discrete and continuous values

Let the training data be a set S = s1, s2 ... of already classified samples. Each sample Si = xl , x2... is a vector where xl , x2 ... represent attributes or features of the sample. The training data is a vector C = c1, c2..., where c1, c2... represent the class to which each sample belongs to.

At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits data set of samples S into subsets that can be one class or the other. It is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute factor with the highest normalized information gain is considered to make the decision. The C4.5 algorithm then continues on the smaller sub-lists having next highest normalized information gain.

3. Technologies used

A. PHP and the Code Igniter Framework

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general purpose server side scripting language that is especially suited for web development and can be embedded into HTML.

CodeIgniter is a well-known open source web application framework used for building dynamic web applications in PHP. Its goal is to enable developers to develop projects quickly by providing a rich set of libraries and functionalities for commonly used tasks with a simple interface and logical structure for accessing these libraries. CodeIgniter is loosely based on the Model-View-Controller (MVC) pattern and we have used it to build the front end of our implementation.

B. MySQL

MySQL is the most popular open source RDBMS which is supported, distributed and developed by Oracle. In the implementation of our web application, we have used it to store user information and students' data.

C. Rapidminer

Rapid Miner is an open source data mining tool that provides data mining and machine learning procedures including data loading and transformation, data preprocessing and visualization, modelling, evaluation, and deployment. It is written in the Java programming language and makes use of learning schemes and attribute evaluators from the WEKA machine learning environment and statistical modelling schemes for the R-Project. We have used Rapid Miner to generate decision tree of C4.5 algorithms.

4. Implementation

We had divided the entire implementation into five stages. In the first stage, information about students who have been admitted to the second year was collected. This included the details submitted to the college at the time of enrolment. In the second stage, extraneous information was removed from the collected data and the relevant information was fed into a database. The third stage involved applying the C4.5 algorithms on the training data to obtain decision trees of both the algorithms. In the next stage, the test data, i.e. information about students currently enrolled in the first year, was applied to the decision trees. The final stage consisted of developing the front end in the form of a web application.

These stages of implementation are:

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-1, January-2019
www.ijresm.com | ISSN (Online): 2581-5792

A. *Student database*

We were provided with a training dataset consisting of information about students admitted to the first year. This data was in the form of a Microsoft Excel 2003 spreadsheet and had details of each student such as full name, application ID, gender, caste, percentage of marks obtained in board examinations of classes X and XII, percentage of marks obtained in Physics, Chemistry and Mathematics in class XII, marks obtained in the entrance examination, admission type, etc. For ease of performing data mining operations, the data was filled into a MySQL database.

B. *Data preprocessing*

Once we had details of all the students, we then segmented the training dataset further, considering various feasible splitting attributes, i.e. the attributes which would have a higher impact on the performance of a student. For instance, we had considered 'location' as a splitting attribute, and then segmented the data according to students' locality. A snapshot of the student database is shown in Figure 2. Here, irrelevant attributes such as students residential address, name, application ID, etc. had been removed. For example, the admission date of the student was irrelevant in predicting the future performance of the student. The attributes that had been retained are those for merit score or marks scored in entrance examination, gender, percentage of marks scored in Physics, Chemistry and Mathematics in the board examination of class XII and admission type. Finally, the "class" attribute was added and it held the predicted result, which can be either "Pass" or "Fail".

Since the attributes for marks would have discrete values, to produce better results, specific classes were defined. Thus, the "merit" attribute had a value "good" if the merit score of the student was 120 or above out of a maximum score of 200, and was classified as "bad" if the merit score was below 120. Also, the value that can be held by the "percentage" attribute of the student are three - "distinction" if the percentage of marks scored by the student in the subjects of Physics, Chemistry and Mathematics was 70 or above, "first_class" if the percentage was less than 70 and greater than or equal to 60, then it was classified as "second_class" if the percentage was less than 60. The attribute for admission type is labelled "type" and the value held by a student for it can be either "AI" (short for All-India), if the student was admitted to a seat available for All-India candidates, or "OTHER" if the student was admitted to another seat.

sr_no	merit_no	merit_marks	app_id	name	gender	cast	location	percent	type
1	328	153.00	EN10256634	AKSHAY DEENATH	Male	Open	Mumbai	95.66	AI
2	725	152.00	EN10279670	YEMPALLE SUSHMA BASWARAJ	Female	Open	Mumbai	88.66	AI
3	1066	143.00	EN10288511	KIRAN SUSHIL GRIFFITHS	Male	Open	Mumbai	96.00	AI
4	1294	136.00	EN10157654	WALCHALE ABHIJEET SUHAS	Male	Open	Mumbai	82.00	AI
5	1419	132.00	EN10255786	KUNAL JADHAV	Male	Open	Mumbai	80.33	AI
6	2166	109.00	EN10297182	KAKKHELE ARVINDRAKUMAR VITTHAL	Male	NT 3 (NT-D)	Mumbai	83.66	GRFDH
7	3290	156.00	EN10172564	TALAWADEKAR ADITYA SHYAM	Male	OBC	Mumbai	89.33	GOBCH
8	5533	144.00	EN10254877	SONAWANE NIKHIL RAJENDRA	Male	SBC/OBC	Mumbai	85.66	GOBCH
9	6882	140.00	EN10196064	PATIL SUMEET BHAGWAN	Male	OBC	Mumbai	88.33	GOBCH
11	1456	168.00	EN10195904	LOHOTE PRANIT TANAJI	Male	Open	Mumbai	92.00	GOPENH
12	2158	162.00	EN10216645	IYER SIDDHARTH SUNDARAM	Male	Open	Mumbai	92.66	GOPENH
13	2619	160.00	EN10258191	GEORGE NISHANT JOSEPH	Male	Open	Mumbai	94.66	GOPENH

merit	gender	percent	type	class
good	Male	distinction	AI	pass
good	Female	distinction	AI	pass
good	Male	distinction	AI	pass
good	Male	distinction	AI	pass
good	Male	distinction	AI	pass
bad	Male	distinction	OTHER	pass
good	Male	distinction	OTHER	pass
good	Male	distinction	OTHER	pass
good	Male	distinction	OTHER	fail
good	Male	distinction	OTHER	pass
good	Male	distinction	OTHER	pass

Fig. 1. Preprocessed student database

C. *Data processing using rapidminer*

The next step was to feed the pruned student database as input to RapidMiner. This helped us in evaluating interesting results by applying classification algorithms on the student training dataset. The results obtained are shown in:

1) *C4.5 algorithm*

The C4.5 algorithm too generates a decision tree, and we obtained one from RapidMiner. This tree, shown in Figure 2, has fewer decision nodes as compared to the tree for improved ID3.

2) *Implementing the performance prediction web application*

RapidMiner helped significantly in finding hidden information from the training dataset. These newly learnt predictive patterns for predicting students' performance were then implemented in a working web application for staff members to use to get the predicted results of admitted students.

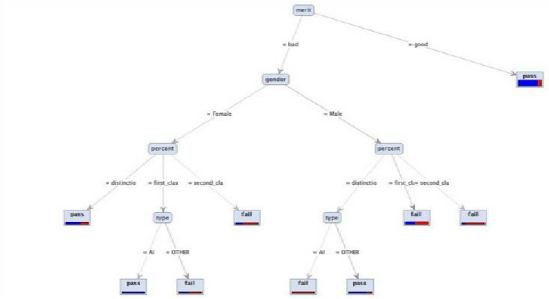

Fig. 2. Decision tree for C4.5

3) *Code igniter*

The web application was developed using a popular PHP framework named Code Igniter. The application has provisions for multiple simultaneous staff registrations and staff logins. This ensures that the work of no two staff members is interrupted during performance evaluation. Figure 3 and Figure 4 depict the staff registration and staff login pages respectively.

4) *Mapping decision trees to PHP*

The essence of the web application was to map the results achieved after data processing to code. This was done in form of class methods in PHP. The result of the improved C4.5 algorithms were in the form of trees and these were translated to code in the form of if-else ladders. We then placed these ladders into PHP class methods that accept only the splitting attributes - PCM percentage, merit marks, admission type and gender as method parameters. The class methods return the final result of that particular evaluation, indicating whether that student would pass or fail in the first semester examination. Fig. 7 shows a class method with the if else ladder.

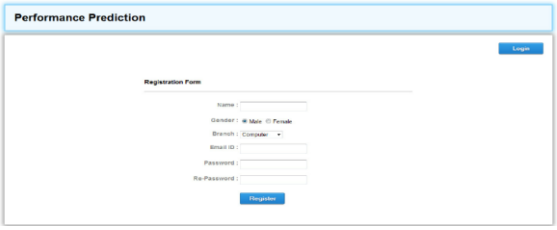

Fig. 3. Registration page for staff members

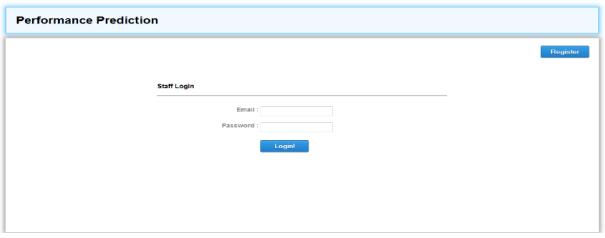

Fig. 4. Login page for staff members

5) Singular evaluation

Once the decision trees were mapped as class methods, we built a web page for staff members to feed values for the name, application ID and splitting attributes of a student, as can be seen in Figure 8. These values were then used to predict the result of that student as either “Pass” or “Fail”.

6) Upload excel sheet singular

Evaluation is beneficial when the results of a small number of students are to be predicted, one at a time. But in case of large testing datasets, it is feasible to upload a data file in a format such as that of a Microsoft Excel spreadsheet, and evaluate each student's record. For this, staff members can upload a spreadsheet containing records of students with attributes in a predetermined order. Fig. 5 shows the upload page for Excel spreadsheets.

```php
public function dtalgo3($percent, $merit, $ad_type, $gender){
    if( $percent == "distinction" )
        return "pass";
    else{
        if( $percent == "first_class" ){
            if( $merit == "bad" ){
                if( $ad_type == "AI" )
                    return "pass";
                else
                    return "fail";
            }
            else
                return "pass";
        }
        else
            return "fail";
    }
}
```

Fig. 5. PHP class method mapping a decision tree

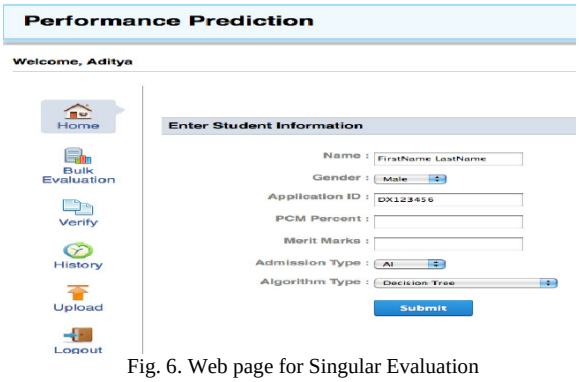

Fig. 6. Web page for Singular Evaluation

D. Bulk evaluation

Under the Bulk Evaluation tab, a staff member can choose an uploaded dataset to evaluate the results, along with the algorithm to be applied over it. After submitting the dataset and algorithm, the predicted result of each student is displayed in a table as the value of the attribute "class". A sample result of Bulk Evaluation can be seen in Fig. 8.

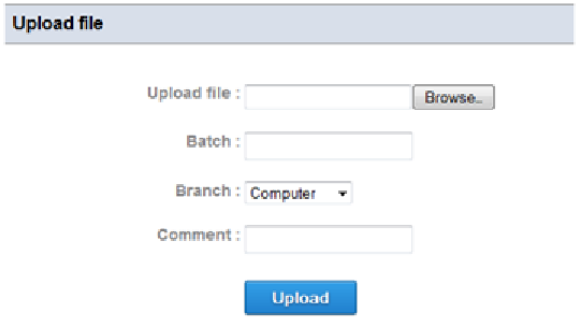

Fig. 7. Page to upload Excel spreadsheet

merit_marks	app_id	name	gender	caste	location	percent	type	class
153	DX10205034	AKSHAY DEBNATH	Male	Open	Mumbai	95.66	AI	PASS
152	DX10279070	YEMPALLE SUSHMA BASWARAJ	Female	Open	Mumbai	86.66	AI	PASS
143	DX10288911	KIRAN SUSHIL GRIFFITHS	Male	Open	Mumbai	90	AI	PASS
136	DX10167854	WALCHALE ABHIJEET SUHAS	Male	Open	Mumbai	82	AI	PASS
132	DX10255786	KUNAL JADHAV	Male	Open	Mumbai	80.33	AI	PASS
109	DX10230782	KARKHELE RAVINDRAKUMAR VITTHAL	Male	NT 3 (NT-D)	Mumbai	83.66	GNTSH	PASS
156	DX10172564	TALAWADEKAR ADITYA SHYAM	Male	OBC	Mumbai	89.33	GOBCH	PASS

Fig. 8. Page showing results after Bulk Evaluation

1) Verifying accuracy of predicted results

The accuracy of the algorithm results can be tested under the Verify tab. A staff member has to select the uploaded verification file which already has the actual results and the algorithm that has to be tested for accuracy. After submission the predicted result of evaluation is compared with actual results obtained and the accuracy is calculated. Figure 9 shows that the accuracy achieved is 75.145% for C4.5 algorithms. Fig. 10 shows the mismatched tuples, i.e. the tuples which were predicted wrongly by the application for the current test data.

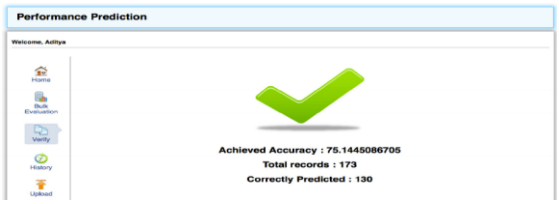

Fig. 9. Accuracy achieved after evaluation

merit_marks	app_id	name	gender	caste	location	percent	type	class	Predicted
140	DX10196064	PATIL SUMEET BHAGWAN	Male	OBC	Mumbai	88.33	GOBCH	fail	PASS
124	DX10297565	MAHAJAN NISHANT VIJAY	Male	OBC	North Maharashtra	58	GOBCO	fail	PASS
118	DX10356072	NARKHEDE JUHI RAJEEV	Female	Open	North Maharashtra	76.33	LOPENO	pass	FAIL
108	DX10149595	WAGHAMARE LAXMAN PANDURANG	Male	OBC	Shivaji + Solapur	74	GOBCO	pass	FAIL
153	DX10182982	JAISWAL ABHAY SHAILESH	Male	Open	Mumbai	75.66	GOPENH	fail	PASS
150	DX10193225	RAJPUT ABHISHEK DANSINGH	Male	Open	Mumbai	82	GOPENH	fail	PASS
93	DX10260441	RAMYA MACHERI	Female	Open	Mumbai	73	AI	pass	FAIL

Fig. 10. Mismatched tuples shown during verification

2) Singular evaluation history

Using the web interface, staff members can view all Singular Evaluations they had conducted in the past. This is displayed in the form of a table, containing attributes of the student and the predicted result. If required, a record from this table may be

Table 1
Results of bulk evaluation

Algorithm	Total Students	Students whose results are correctly predicted	Accuracy (%)	Execution Time (in milliseconds)
C4.5	173	130	75.145	39.1

Table 2
Results of Singular Evaluation

Algorithm	Total Students	Students whose results are correctly predicted	Accuracy (%)
C4.5	9	7	77.778

deleted by a staff member. A snapshot of this table is shown in Fig. 11.

Application ID	Name	Gender	Percentage	Merit marks	Admission Type	Algorithm	Class	
DX123456	Aditya Gaykar	Male	89.17	157	OTHER	C4.5	pass	Delete
DX123456	Rahul	Male	123	89	OTHER	Decision Tree	pass	Delete
DX121312	Aditya Gaykar	Male	90.33	157	OTHER	Decision Tree	pass	Delete

Fig. 11. History of Singular Evaluations performed by staff members

5. Future work

In this project, prediction parameters such as the decision trees generated using RapidMiner are not updated dynamically within the source code. In the future, we plan to make the entire implementation dynamic to train the prediction parameters itself when new training sets are fed into the web application. Also, in the current implementation, we have not considered extracurricular activities and other vocational courses completed by students, which we believe may have a significant impact on the overall performance of the students. Considering such parameters would result in better accuracy of prediction.

6. Conclusion

In this paper, we have explained the system we have used to predict the results of students currently in the first year of engineering, based on the results obtained by students currently in the second year of engineering during their first year. The results of Bulk Evaluation are shown in Table 1. Random test cases considered during individual testing resulted in approximately equal accuracy, as indicated in Table 2.

Thus, for a total of 182 students, the average percentage of accuracy achieved in Bulk and Singular Evaluations is approximately 75.275.

Acknowledgements

We express sincere gratitude to our project guide Ms. Smita Dandge, for their guidance and support.

References

[1] Han, J. and Kamber, M., (2006) Data Mining: Concepts and Techniques, Elsevier.
[2] Dunham, M.H., (2003) Data Mining: Introductory and Advanced Topics, Pearson Education Inc.
[3] Kantardzic, M., (2011) Data Mining: Concepts, Models, Methods and Algorithms, Wiley-IEEE Press.
[4] Xiaoliang, Z., Jian, W., Hongcan Y., and Shangzhuo, W., (2009) "Research and Application of the improved Algorithm C4.5 on Decision Tree", International Conference on Test and Measurement (ICTM), Vol. 2, pp. 184-187.
[5] CodeIgnitor User Guide Version 2.14, http://ellislab.com/codeigniter/user-guide/toc.html
[6] RapidMiner, http://rapid-i.com/content/view/181/190/
[7] MySQL-The world's most popular open source database, http://www.mysql.com/