

## Research Article

### Predicting Performance of Schools by Applying Data Mining Techniques on Public Examination Results

<sup>1</sup>J. Macklin Abraham Navamani and <sup>2</sup>A. Kannammal

<sup>1</sup>Department of Computer Applications, Karunya University,

<sup>2</sup>Department of Computer Applications, Coimbatore Institute of Technology, Coimbatore, India

**Abstract:** This study work presents a systematic analysis of various features of the higher grade school public examination results data in the state of Tamil Nadu, India through different data mining classification algorithms to predict the performance of Schools. Nowadays the parents always targets to select the right city, school and factors which contributes to the success of the results in schools of their children. There could be possible effects of factors such as Ethnic mix, Medium of study, geography could make a difference in results. The proposed work would focus on two fold factors namely Machine Learning algorithms to predict School performance with satisfying accuracy and to evaluate the data mining technique which would give better accuracy of the learning algorithms. It was found that there exist some apparent and some less noticeable attributes that demonstrate a strong correlation with student performance. Data were collected through the credible source data preparation and correlation analysis. The findings revealed that the public examinations results data was a very helpful predictor of performance of school in order to improve the result with maximum level and also improved the overall accuracy with the help of Adaboost technique.

**Keywords:** Educational data mining, prediction of school performance, public examination, random forest

## INTRODUCTION

Tamil Nadu Board of Secondary Education, established in 1910, is under the purview of the Department of Education, Government of Tamil Nadu, India. The Directorate of Government Examinations was formed as a separate directorate in Feb.1975. Dr. Lawrence planned and implemented the all India 10+1+2 pattern of education in 1978. The Higher Secondary Examinations were introduced in the year 1980 (Anon, 2014a, b). This Higher Secondary examinations play vital role in the career of any student who is completing their schools with talent. These results mostly determine their career aspirations and considered entry criteria to join any new college or universities. The scores from the Higher Secondary Board examinations are used by universities to determine eligibility and as a cut-off for admissions into their courses. Thus this examination and its results play a vital role in the context of educational systems in Tamil Nadu, India. Despite its importance and unique position in the educational systems, there is no or very limited predictive strength is existing on the Higher Secondary Examinations.

The transformation of examinations from a student selection and certification tool into an indicator of school effectiveness and an accountability instrument is

a core reform in educational policy making (Naidoo *et al.*, 2014). There is increasing research interests in using data mining in education. This new emerging field, called Educational Data Mining (Barnes *et al.*, 2009), concerns with developing methods that discover knowledge from data originating from educational environments. Databases are rich with hidden information, which can be used for intelligent decision making. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends (Micheline and Jiawei, 2008). School evaluation is part of the decision-making process in education; it involves judgments about the performance of schools through systematically collecting and analyzing information and relating this to explicit objectives, criteria and values. Ideally, school evaluation involves an (internal and external) assessment that covers all aspects of a school and their impact upon student learning. Such review and analysis covers a range of inputs, processes and outcomes reflected in such elements as staffing and physical resources, curriculum resources, the quality of leadership and management, learning and teaching activities and the standards achieved by students.

In our previous work (Macklin *et al.*, 2014), we provided data cubes to analyse the exam results. We

**Corresponding Author:** J. Macklin Abraham Navamani, Department of Computer Applications, Karunya University, Coimbatore, India

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

Table 1: Statistics of students in private examination

Attribute name	Type	Statistics	Range
Performance	Nominal	Mode = Average (8202), least = Bad (5019)	Excellent (7163), Average (8202), Good (7610), Bad (5019)
Result year	Integer	avg = 2011.032 +/- 0.816	(2010.000 ; 2012.000)
Dist	Nominal	Mode = CHENGALPATTU (1043), least = DUBAI (6)	CHENGALPATTU (1043), COIMBATORE (997), TIRUPATHUR (815), PONNERI (779), TIRUPPUR (758), SALEM (713), NAMAKKAL (661), VILLUPURAM (563), VELLORE (557), more...
sch_name	Nominal	Mode = (26), least = G H S S G MEENSURUTTI (1)	(26), GOVT HSS VEERAPANDI (20), GOVT HSS MANGALAM (18), GOVT HSS AGARAM (12), DHARAPURAM (10), ... and 6219 more ...
Sex	Nominal	Mode = F (14392), least = M (13602)	F (14392), M (13602)
Total students	Integer	Avg = 154.005 +/- 188.404	(2.000; 2406.000)
Mark1	Real	Avg = 142.583 +/- 19.734	(13.000; 194.000)
Mark2	Real	Avg = 126.300 +/- 28.259	(0.000; 189.000)
Mark3	Real	Avg = 119.763 +/- 23.074	(30.000; 196.000)
Mark4	Real	Avg = 122.879 +/- 22.785	(19.000; 196.000)
Mark5	Real	Avg = 132.660 +/- 24.157	(10.000; 200.000)
Mark6	Real	Avg = 119.876 +/- 26.890	(9.000; 200.000)
Average score	Real	Avg = 764.055 +/- 126.556	(226.000; 1163.000)
numPass	Real	Avg = 132.346 +/- 169.667	(0.000; 2282.000)
numFail	Real	Avg = 0.906 +/- 6.806	(0.000; 244.000)
Pass percentage	Real	Avg = 0.854 +/- 0.171	(0.000; 1.000)

test different classification algorithms to predict which schools performs best based on the historical result data. We selected various algorithms like Naïve Bayes, Random Forest and K-NN. After evaluating the outcomes of these classifiers we decided to increase the accuracy using Adaboost and considering Naive Bayes as the weak classifier of the 3 classifiers. Overall, our Adaboost algorithms performance was accurate after multiple iterations.

### LITERATURE REVIEW

In all of the countries, a major component of evaluation and school reform includes attempts to improve academic standards and quality through the use of tests or examinations. Many assessment systems have come into the picture includes the following: (a) national assessment, which includes public (external) examinations to select students for successive levels in the education system, system assessments to determine if children are acquiring certain knowledge, skills and values. Student results leads the school become a judgments on the school's performance. In a growing number of countries, 'league tables' of schools, especially at secondary level, are published in newspapers, as information to the public, to allow parents to choose a school (Naidoo *et al.*, 2014). Sonali *et al.* (2012) determined that data mining could be used to be used to improve the education system to enhance the efficacy and overall efficiency by optimizing the resource available. Brijesh and Saurabh (2011) with the help of variables such as Semester Marks, Attendance, etc., used in the classification techniques for predicting the end semester results. Sundar (2013) Prediction of student's performance based on the exam results of engineering college students where the comparison of

the classifiers has been done which had helped students to focus on their performance area. Kabakchieva (2013) 10330 instances of data from Bulgarian schools were taken as samples and classified with labels Excellent, Very Good, Good, Average and Bad. These were used to predict the target label (Adeyemi, 2008). This study focus on reviewing the strategy by looking at the performance of the students at Junior Secondary Certificate examinations in the Ondo State, Nigeria. In one of the experiment done for evaluating performance of various classification techniques for distance education student's education dataset, it has been identified that Naive Bayes performs adequately with accuracy of 80.97% (García-Saiz and Zorrilla, 2011).

**Data set:** The data set used in this proposed work contains students public examination results information collected from the Directorate of Higher Secondary Education, Tamil Nadu. This was done based in reference to building data warehouse/data mart to store and analyze the public examination results of higher grade students by Directorate of Government Examinations belonging to Tamil Nadu, India which was obtained based on Microsoft SSAS (Macklin *et al.*, 2014). The dataset has around 27994 rows which has data segregated by District, School, Sex, Average Marks in individual subjects and the overall pass percentage among them. There were 6269 schools covering around 71 districts. Totally there are 2305726 (53%) female students and 2005502 (47%) male students. Since the volume of data we handle were huge, we intend to use MySQL with Rapid Miner for the purpose of loading data and training. The data was available in the MS Access 2007 format, further to that we exported the same to CSV Format and then did loaded to the MySql database. Table 1 shows that the data was having details of the students who have attended the

exam in private. Those students have been removed to have refined dataset.

**METHODOLOGY**

**Data mining:** Data mining refers to extracting or "mining" knowledge from large amounts of data. Educational Data Mining is an emerging interdisciplinary research area that deals with the development of methods to explore data originating in an educational context. Data mining helps to discover underlying structures in the data, to turn data into information and information into knowledge. It can be defined as the process involved in extracting interesting, interpretable, useful and novel information from data from the Educational Domain such as Schools, Colleges, E-learning platforms, Intelligent Tutoring, Learning Management Systems (Romero and Ventura, 2010). Data mining consists of a set of techniques that can be used to extract relevant and

interesting knowledge from data. Data mining has several tasks such as association rule mining, classification and prediction and clustering. Classification techniques are supervised learning techniques that classify data item into predefined class label. It is one of the most useful techniques in data mining to build classification models from an input data set. The used classification techniques commonly build models that are used to predict future data trends. There are several algorithms for data classification such as decision tree and Naïve Bayes classifiers. With classification, the generated model will be able to predict a class for given data depending on previously learned information from historical data. Figure 1 depicts that the overall process of Data Mining.

**Classification:** Classification refers to the task of predicting a class label for a given unlabeled point (Zaki and Meira Jr., 2013). Based on the above labelling approach each training point belongs to one of the 4 different classes namely "Excellent", "Good", "Average"

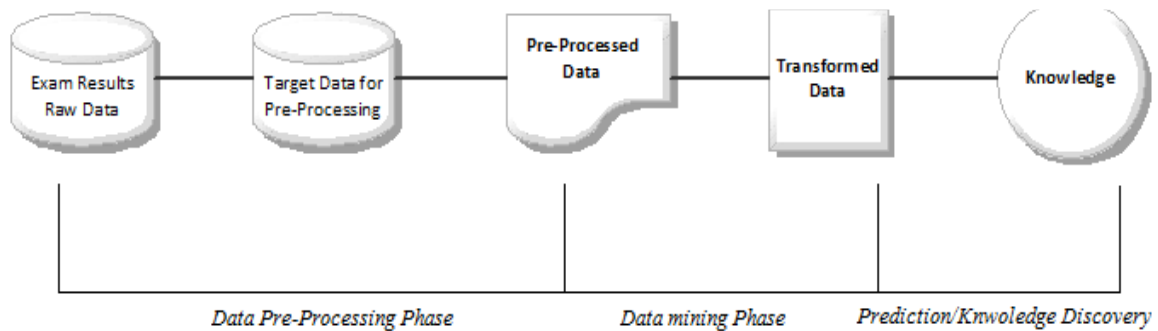


Fig. 1: Data mining overall process diagram

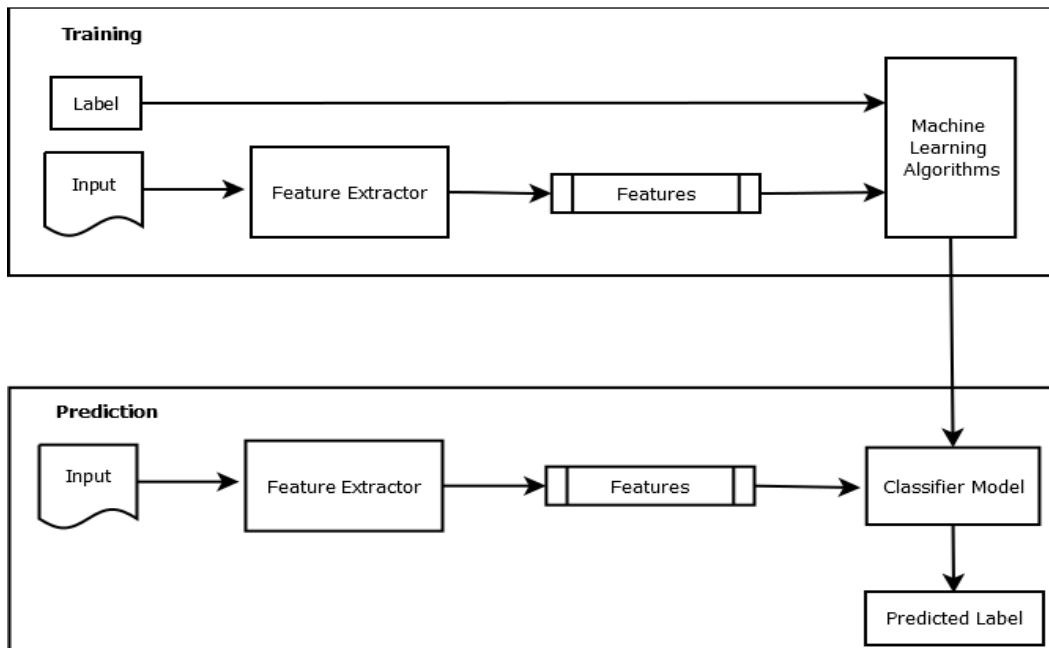


Fig. 2: Pictorial representation of classification

and "Bad". In a multiclass prediction, the result on a test set is often displayed as a two-dimensional confusion matrix with a row and column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Figure 2 shows that the pictorial representation of classification.

As our intention is to choose the best tool and classification algorithms for handling educational datasets which can be integrated in our Java application tool, we have to search among those that can support categorical and numeric data, handle large set of data and be accurate. Given that the confusion matrix of the Classifier is oriented toward True Positive and True Negative using One Vs All Methodology. i.e., that a given row of the matrix corresponds to specific value for the "truth", we have:

$$\text{Precision } i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$\text{Recall } i = \frac{M_{ii}}{\sum_j M_{ij}}$$

That is, precision is the fraction of events where we correctly declared  $i$  out of all instances where the algorithm declared  $i$ . Conversely, recall is the fraction of events where we correctly declared  $i$  out of all of the cases where the true of state of the world is  $i$ .

**Rapid miner:** While technology enables us to capture and store ever larger quantities of data, finding relevant information like underlying patterns, trends, anomalies

and outliers in the data and summarizing them with simple understandable and robust quantitative and qualitative models is a grand challenge. RapidMiner is a system which supports the design and documentation of an overall data mining process. It offers not only an almost comprehensive set of operators, but also structures that express the control flow of the process. RapidMiner and RapidAnalytics provide an integrated environment for all steps of the data mining process, an easy-to-use Graphical User Interface (GUI) for the interactive data mining process design, data and results visualization, validation and optimization of these processes and for their automated deployment and possible integration into more complex systems. RapidMiner enables one to design data mining processes by simple drag and drop of boxes representing functional modules called operators into the process, to define data flows by simply connecting these boxes, to define even complex and nested control flows and all without programming (Markus and Ralf, 2013). The below given figure refers to the process block typically used in rapid miner to setup the Data mining process with different machine learning algorithms. As we can see in this example first the database is read and attributes/features are selected for the process, the set role operator defines the feature which is to be considered for learning and next goes to the validation step. Figure 3 describes that the process block of Rapid Miner.

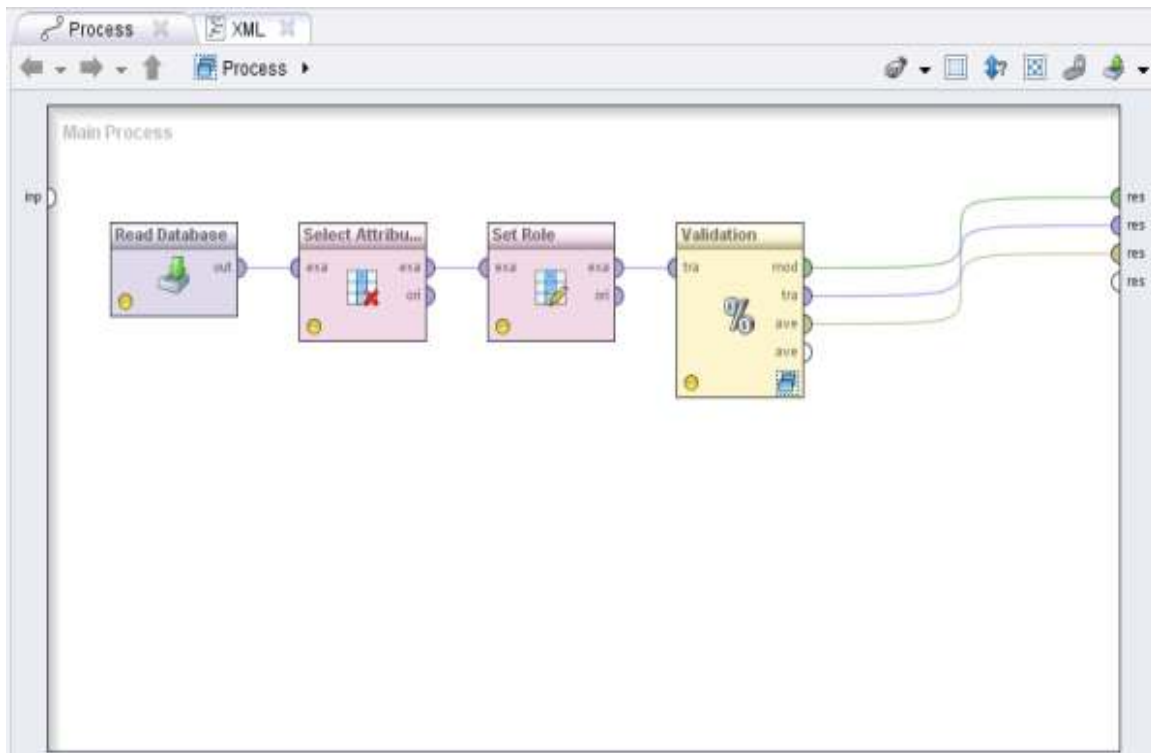


Fig. 3: Rapid miner process block

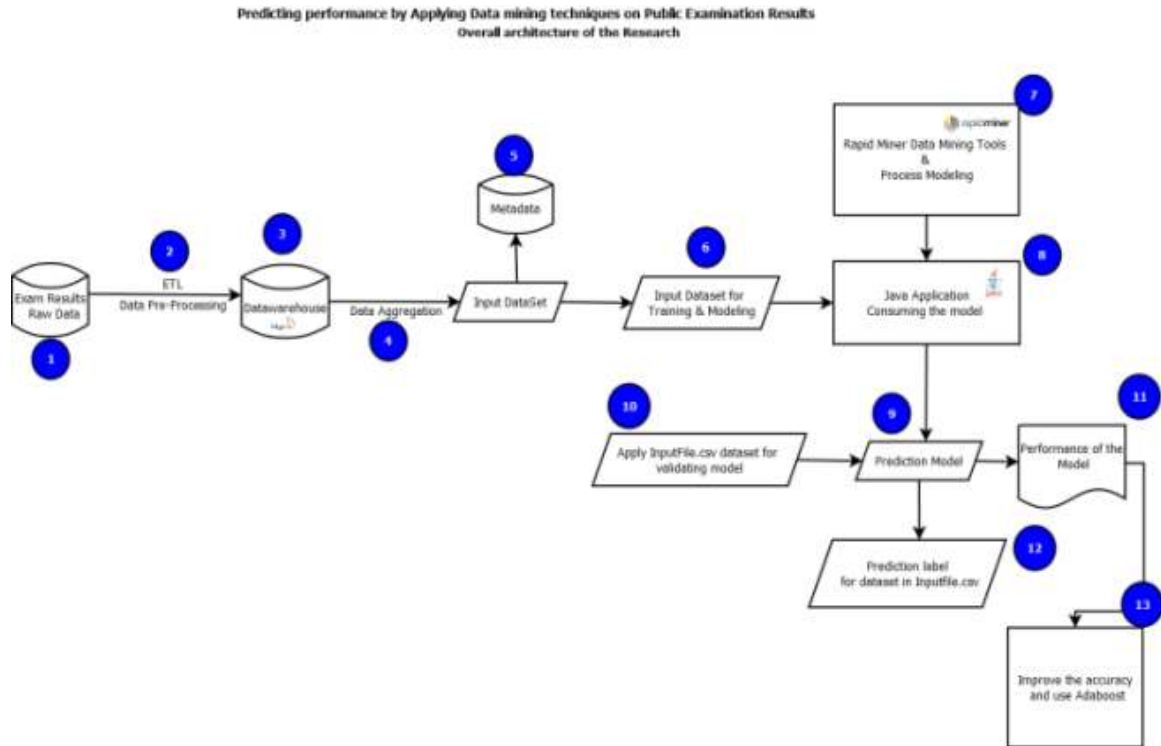


Fig. 4: Research architecture diagram

**Research architecture and implementation:** Figure 4 shows that the Research Architecture Diagram of the proposed work and the steps are described as follows:

- Raw data is obtained in the CSV format which contains the Public Examinations results.
- Data pre-processing such as eliminating records such as results of candidates who has written exam in private, etc.,
- These pre-processed data are loaded to a MySQL based Datawarehouse which we use it to provide as an Input to the Rapid Miner processes along with labels such as (Excellent, Average, Good and Bad)
- From the data available in the Datawarehouse we use Cubes to aggregate the data and generate a data set which is given as input to the Rapid Miner Process model.
- The aggregated data will form the input dataset with the data from the year 2010, 2011 and 2012 data which would be up to 27994 rows of data.
- The Training dataset is provided as input to the Rapid Miner process model and we are handling through X-Cross validation with 10 Cross validation. Tenfold cross-validation is the standard way of measuring the error rate of a learning scheme on a particular dataset; for reliable results, 10 times 10-fold cross-validation.
- The input data set is obtained and Label Role is setup for the Machine learning to identify during the learning process

- Java application has been developed which will use the models and generate the appropriate performance and prediction which uses the core library of the Rapid Miner.
- Its generates the appropriate Prediction Model based on the given selection.
- The inputfile.csv have the input data for validating the Prediction model.
- The performance will report the confusion matrix and overall accuracy of the prediction model
- We can also get the Predicted Label for the data available in the inputfile.csv. This will help us to validate the performance of the prediction model.
- Continuously iterate the weaker learning algorithm for better performance using Adaboost algorithm.

In the context of the problem of estimating classes for a test set containing instances. The true classes are noted, whereas the estimated classes, as defined by the considered classifier, are noted as Excellent, Good, Average and Bad (Cherif *et al.*, 2011). Most measures are not processed directly from the raw classifier outputs, but from the confusion matrix built from these results. This matrix represents how the instances are distributed over estimated (rows) and true (columns) classes shown in Table 2.

The terms  $n_{ij}(1 \leq i, j \leq k)$  correspond to the number of instances put in class number  $i$  by the classifier (i.e.,  $C_i$ ), when they actually belong to class number  $j$  (i.e.,  $\hat{C}_j$ ). The rules for labeling classification is exists in Table 3.

Table 2: Matrix representation

	C <sub>1</sub>	C <sub>2</sub>	C <sub>...</sub>	C <sub>n</sub>
C <sub>1</sub>	n <sub>11</sub>	..	..	n <sub>1k</sub>
C <sub>2</sub>	..	..	..	..
C <sub>...</sub>	..	..	..	..
C <sub>n</sub>	nk <sub>1</sub>	..	..	n <sub>kk</sub>

Table 3: Experimental results and discussion

Class EXCELLENT IF PASSPERCENTAGE>1
Class GOOD IF PASSPERCENTAGE>0.9 AND PASSPERCENTAGE<1
Class AVERAGE IF PASSPERCENTAGE>0.7 AND PASSPERCENTAGE<0.9
Class BAD IF PASSPERCENTAGE>0.5 AND PASSPERCENTAGE<0.7

### RESULTS AND DISCUSSION

**Naive Bayes process:** A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. The Naive Bayes classifier assumes that attributes are independent, but it is still surprisingly powerful for many applications (Zaki and Meira Jr., 2013). In naive Bayes classifiers, every feature gets a say in determining which label should be assigned to a given input value. To choose a label for an input value, the naive Bayes classifier begins by calculating the prior probability of each label, which is determined by checking frequency of each label in the training set (Steven, 2009). The contribution from each feature is then combined with this prior probability, to arrive at a likelihood estimate for each label. The label whose likelihood estimate is the highest is then assigned to the input value. The independence assumption immediately implies that the likelihood can be decomposed into a product of dimension-wise probabilities:

$$V_{nb} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod P(a_i | v_j)$$

We generally estimate P(ai | Vj) using m-estimates:

$$P(a_i | v_j) = \frac{n_c + mp}{n + m}$$

where,

n = The number of training examples for which v = v<sub>j</sub>

nc = Number of examples for which v = v<sub>j</sub> and a = a<sub>j</sub>

p = A priori estimate for P (a<sub>j</sub> | v<sub>j</sub>)

m = The equivalent sample size

Implementation of the Naive bayes algorithm in the Rapid Miner with the dataset provides accuracy of 83.96% which is depicted in the Fig. 5.

**Ensemble learning methods:** Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions. Ensembles are well-established as a method for obtaining highly accurate classifiers by combining less accurate ones (Dietterich, 2014). Since we have large volume of data to be used for training, we envisaged the use of ensemble based systems. Ensemble based systems can be useful when dealing with large volumes of data or lack of adequate data. When the amount of training data is too large to make a single classifier training difficult, the data can be strategically partitioned into smaller subsets. Each partition can then be used to train a separate classifier which can then be combined using an appropriate combination rule (see below for different combination rules).

**Random forest:** Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest (Breiman, 2001). Random forest (or random forests) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees (Anon, 2013). To classify a new object from an input vector, put the input vector down each of the CARTs in the forest. Each CART gives a classification and Random Forest asks the trees "votes" for that class. The forest chooses the

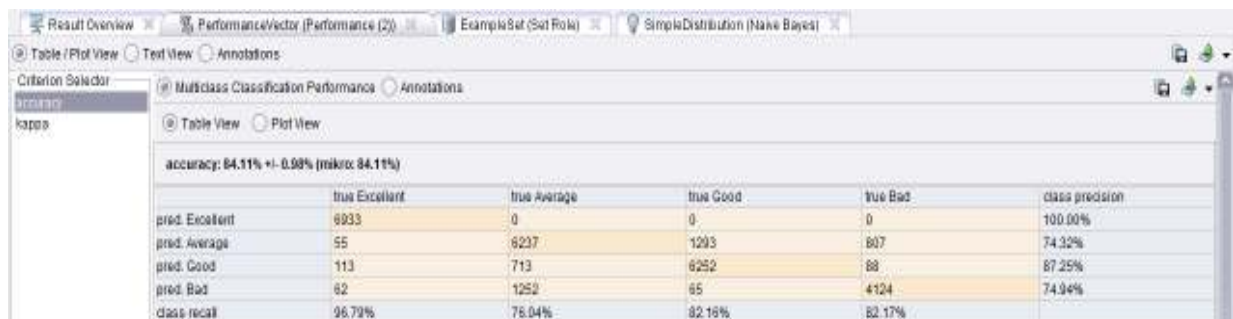


Fig. 5: Naive Bayes rapid miner implementation results



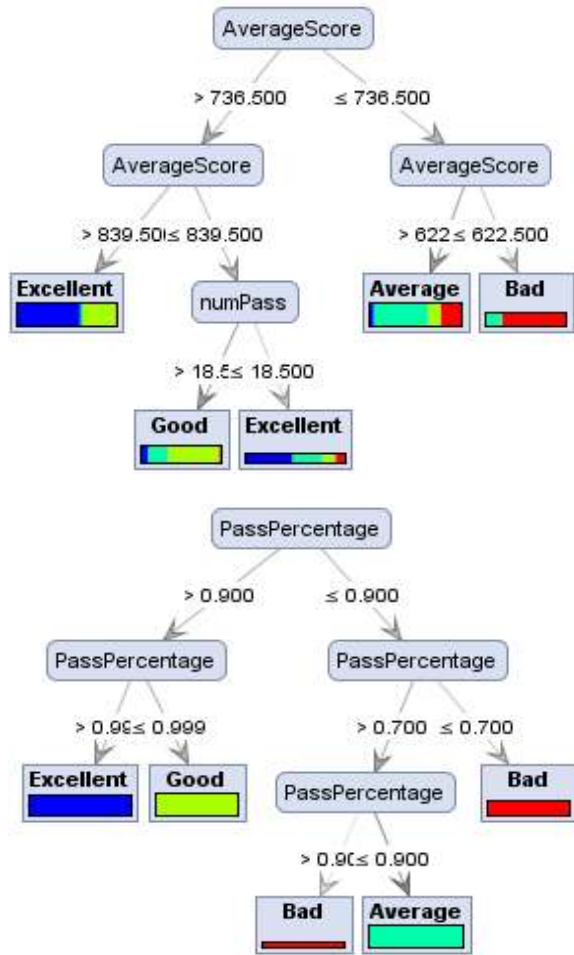


Fig. 6: Decision trees from random forest

classification having the majority votes. Random forest was attempted with Gain Ratio which resulted with the accuracy of 67.96% within 43 sec, to further improve the performance attempt was made to evaluate Information gain which resulted in 71.21% within 1 min 10 sec. Figure 6 are the some of the decisions trees which are getting generated out of Random Forest.

**K-NN:** KNN, originally proposed by Fix and Hodges is a very simple 'instance-based' learning algorithm. The principle of this method is based on the intuitive concept that data instances of the same class should be closer in the feature space. While a training dataset is required, it is used solely to populate a sample of the search space with instances whose class is known. No actual model or learning is performed during this phase; for this reason, these algorithms are also known as lazy learning algorithms. Different distance metrics can be used, depending on the nature of the data. Euclidean distance is typical for continuous variables, but other metrics can be used for categorical data. Specialized metrics are often useful for specific problems, such as text classification. When an instance whose class is

unknown is presented for evaluation, the algorithm computes its k closest neighbors and the class is assigned by voting among those neighbors. To prevent ties, one typically uses an odd choice of k for binary classification. For multiple classes, one can use plurality voting or majority voting (Anon, 2013). We shall define KNNC in a more rigorous manner. Suppose that we are given a training dataset of n points with their desired class, as shown below:

$$\{(x_i, y_i), (x_2, y_2), \dots, (x_n, y_n)\}$$

where,  $(x_i, y_i)$  represent data pair i, with  $x_i$  as the feature vector and  $y_i$  as the corresponding target class. Then for a new data point x, the most likely class should be determined by KNNC (k = 1 in this case), as follows:

$$nnc(x, 1) = y_p, p = \operatorname{argmin}_i \|x - x_i\|_2$$

The preceding equation uses the nearest neighbor to determine the class. Alternatively, we can have K nearest neighbors to determine the class by voting. As the extension to KNN is straightforward, we shall not formulate it separately. Using Mixed Euclidean distance as the methodology we have implemented this algorithm. In our case by implementing K-NN we were able to arrive at the accuracy of 68.49% within the time frame of 1 min 55 sec.

**Comparison of Classifier performance:** To gauge and investigate the performance on the selected classification methods or algorithms namely Naive Bayes, Random Forest and K-NN we implemented the methodology outlined in the preceding section with the help of Rapid Miner. All the implementation are done with 10-Fold cross validation and final results are provided to compare in the following Table 4.

Since the information gain has better accuracy rate when compared to Gain Ratio with respect to Random Forest we use Information Gain for benchmarking them.

Figure 7 shows that the comparison of various classification of instances. In this above chart we can we can realize that Naive Bayes has classified with better accuracy with 23504 instances are predicted correctly. Subsequently we have taken the following aspects to do comparison between different classifiers as we use multiple classes in our research. The following are the factors which we use to measure and compare their performance across their respective classes: Sensitivity (specificity) approximates the probability of the positive (negative) label being true; in other words, it assesses the effectiveness of the algorithm on a single class; F-score is a composite measure which benefits algorithms with higher sensitivity and challenges algorithms with higher specificity.

Table 4: Comparison of performance of different classifiers

Classification methods factors	Random forest			
	Information gain	Gain ratio	K-NN	Naïve Bayes
Overall time taken classification	1:10	43 Sec	2:03	11 Sec
Accuracy	71.21%	67.96%	68.49%	83.96%
Kappa	0.605	0.555	0.575	0.786
Classification error	28.79%	32.04%	31.51%	16.04%
Mean squared error	0.448	0.265	0.315	0.143
Mean absolute errors	0.369	0.417	0.315	0.164
Root mean squared error	0.448	0.51	0.561	0.378
Root relative squared error	1.195	1.362	1.499	1.008

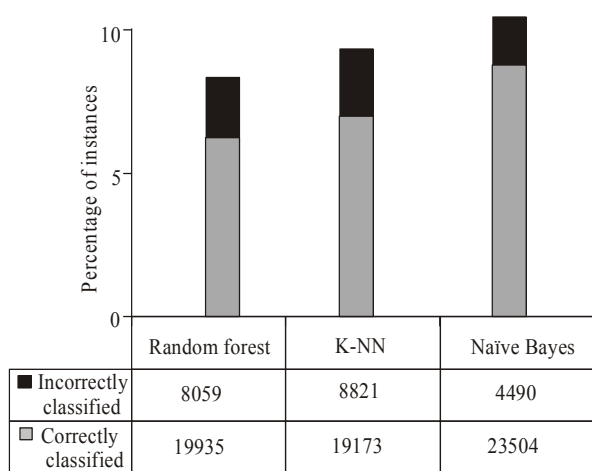


Fig. 7: Performance of classifiers based on classification of instances

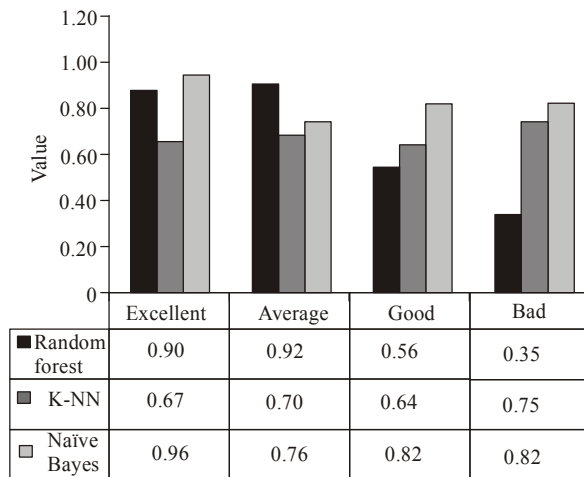


Fig. 8: Sensitivity

The sensitivity across multiple classes are shown in Fig. 8 and Specificity across different classes are depicts in Fig. 9. The F-Score Comparison is describes in Fig. 10.

**Validation of the performance of the classifier:** To validate the accuracy of the classifier we decided to predict the performance of the school by providing inputs through an external source to the Java

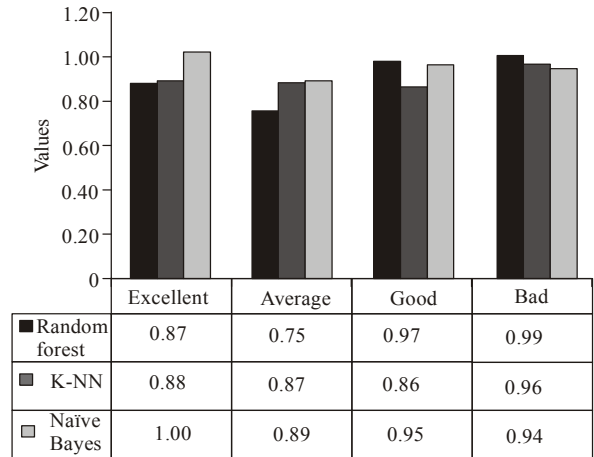


Fig. 9: Specificity

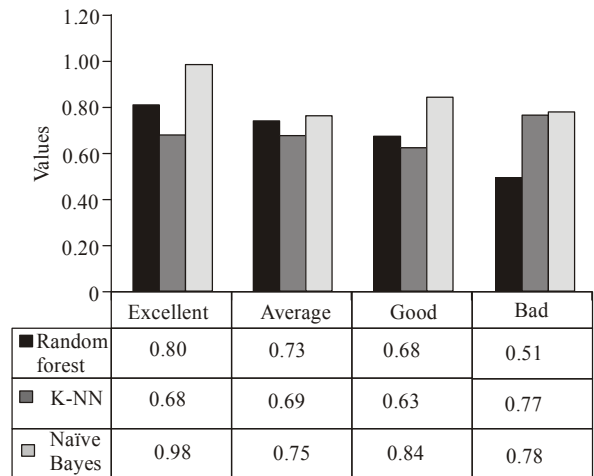


Fig. 10: F-score

application is shown in Table 5. In the context of validation, Random Forest gives the better accuracy for the data selected from the year 2013.

**Implementation of meta-algorithm: AdaBoost:** AdaBoost (Abu Afza *et al.*, 2011), short for Adaptive Boosting, is a machine learning algorithm, formulated by Freund and Schapire (2007) and steps to iterate better performance was taken. This Boosting approach is being taken with the idea of creating a highly



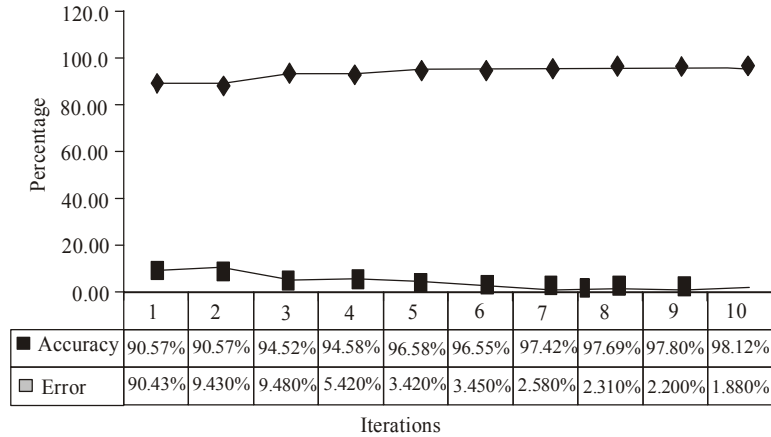


Fig. 11: Adaboost accuracy increase over iterations

Table 5: Validation on the classifier with actual data

Result year	Dist	Sch Name	Sex	Total students	Actual label	KNN-Prediction	Naïve Bayes -Prediction	Random fores prediction
2013	COIMBATORE	CORPN HSS PEELAMEDU CBE	F	140	Good	Good	Excellent	Good
2013	COIMBATORE	CORPN HSS PEELAMEDU CBE	M	92	Average	Average	Excellent	Average
2013	COIMBATORE	C M S MAT HSS GANAPATHY CBE	F	226	Excellent	Good	Excellent	Excellent
2013	COIMBATORE	C M S MAT HSS GANAPATHY CBE	M	136	Excellent	Good	Excellent	Excellent

Table 6: Iterations use in Meta-algorithm

Iteration #	Accuracy (%)	Error (%)	Time (Min: sec)
1	90.57	9.43	4:08
2	90.57	9.43	3:30
3	94.52	5.48	3:52
4	94.58	5.42	3:27
5	96.58	3.42	3:22
6	96.55	3.45	3:50
7	97.42	2.58	4:08
8	97.69	2.31	2:03
9	97.80	2.20	1:56
10	98.12	1.88	2:02

accurate prediction rule by combining many relatively weak and inaccurate rules on the existing Naive bayes application which has fair accuracy in the current context. Since this algorithm is the first practical boosting algorithm and remains one of the most widely used and studied, with applications in numerous fields. The weak learner naive bayes which is providing a accuracy is 83.96% is further being improved by iterations with the help of this Meta-algorithm and the iterations are shown in Table 6:

**Pseudo code:**

Given:  $(x_i, y_i) \dots (x_m, y_m), x_i \in X, y_i \in \{-1, +1\}$   
 Initialize weights  $D_1(i) = 1/m$   
 For  $t = 1 \dots T$ :  
 Find  $h_t = \arg \max_{h_j \in H} \epsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)]$   
 If  $\epsilon_j \geq \frac{1}{2}$  then stop

$$\text{Set } \alpha_t = \frac{1}{2} \log \left( \frac{1-\epsilon_j}{\epsilon_j} \right)$$

Update

$$D_{t+1}(i) = \frac{D_t(i) \exp(\alpha_t y_i h_t(x_i))}{Z_t}$$

Output the final classifier:

$$H(x) = \text{Sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right)$$

Subsequent to the evaluation and improvement of the Naive Bayes performance with the boosting algorithm the performance has reached up to 98.12% from 83.96% which is 14% higher than the original performance of Naive Bayes without adaboost implementation (Fig. 11).

**CONCLUSION**

Though there are different benchmarks comparing the performance and accuracy of different classification algorithms, there are still very few experiments carried out on Educational datasets such as this one we have done in this experiment. We have compared the performance and the interpretation level of the output of different classification techniques applied on educational datasets with multiple classes in order to determine which one is more suitable for integrating with Java application and use widely. As a conclusion, we have met our objective which is to evaluate and investigate the three selected classification algorithms which would help which to predict the performance of the school. Our experimentation shows that there is not one algorithm that obtains significantly better

classification accuracy. Though in our case Naive Bayes has better accuracy rate of 83.96% when compared with K-NN having 68.49% and Random Forest having 71.21%, respectively. The accuracy of the weak classifier Naive Bayes is also increased upto 98.12% with the help of Adaboost Algorithm. Added to that Naive Bayes can generate this performance within 11 sec. From the above results it is clear that Naive Bayes classification techniques can be applied on educational data for predicting the School's outcome and improve their results. Our near future work is to extend this experimentation by building a novel self-constructing cascading classifier algorithm for analyzing the public Examination results.

### REFERENCES

- Abu Afza, A.J.M., F. Dewan and M.R. Chowdhury, 2011. A hybrid classifier using boosting, clustering and naïve Bayesian classifier. *World Comput. Sci. Inform. Technol. J.*, 1(3): 105-109.
- Adeyemi, T.O., 2008. Predicting students' performance in senior secondary certificate examinations from performance in junior secondary certificate examinations in Ondo State, Nigeria. *Hum. Soc. Sci. J.*, 3(1): 26-36.
- Anon, 2013. Data Mining Algorithms in R/Classification/kNN. [Online]. Retrieved form: [http://en.wikibooks.org/wiki/Data\\_Mining\\_Algorithms\\_In\\_R/Classification/kNN](http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Classification/kNN). (Accessed on: November 12, 2013)
- Anon, 2014a. Wikipedia. Retrieved form: [http://en.wikipedia.org/wiki/Tamil\\_Nadu\\_Board\\_of\\_Secondary\\_Education](http://en.wikipedia.org/wiki/Tamil_Nadu_Board_of_Secondary_Education). (Accessed on: May 12, 2014)
- Anon, 2014b. Random Forest. Retrieved form: [https://en.wikipedia.org/wiki/Random\\_forest](https://en.wikipedia.org/wiki/Random_forest). (Accessed on: May 2, 2014)
- Barnes, T., M. Desmarais, C. Romero and S. Ventura, 2009. Educational data mining. *Proceeding of 2nd International Conference on Educational Data Mining*. Cordoba, Spain.
- Breiman, L., 2001. Random Forest. *Statistics Department University of California, Berkeley, CA 94720*.
- Brijesh, K.B. and P. Saurabh, 2011. Mining educational data to analyze students' performance. *Int. J. Adv. Comput. Sci. Appl.*, 2(6): 63-69.
- Cherif, H., L. Vincent and Hocine, 2011. Accuracy Measures for the Comparison of Classifiers. Retrieved form: <http://web.engr.oregonstate.edu/~tgd/classes/534/slides/part13.pdf>. (Accessed on: May 2, 2014)
- Dietterich, T.G., 2014. Ensemble Methods in Machine Learning. Important Milestones of Directorate of Government Examinations. Retrieved form: <http://dge.tn.gov.in/milestone.htm>. (Accessed on: May 5, 2014)
- Freund, Y. and R.E. Schapire, 2007. Experiments with a new boosting algorithm. *Proceeding of the 13th International Conference on Machine Learning*, pp: 148-156.
- García-Saiz, D. and M. Zorrilla, 2011. Comparing classification methods for predicting distance students' performance. *Proceeding of the 2nd Workshop on Applications of Pattern Analysis Workshop and Conference*, pp: 26-32.
- Kabakchieva, D., 2013. Predicting student performance by using data mining methods of classification. *Cybern. Inform. Technol.*, 13(1): 61-72.
- Macklin, A.N.J., A. Kannammal and J.T.P. Ranjit, 2014. Building multi-dimensional cube to analyze public examination results: A business intelligence approach. *Appl. Mech. Mater.*, 622: 11-22.
- Markus, H. and K. Ralf, 2013. RapidMiner: Data Mining Use Cases and Business Analytics Applications. Chapman and Hall/CRC. Retrieved form: <http://www.amazon.in/RapidMiner-Analytics-Applications-Knowledge-Discovery/dp/1482205491>. (Accessed on: May 15, 2014)
- Micheline, K. and H. Jiawei, 2008. *Data Mining-concepts and Techniques*. 2nd Edn., Morgan Kaufmann-Elsevier Publishers. Retrieved form: <http://web.engr.illinois.edu/~hanj/bk2/toc.pdf>. (Accessed on: May 25, 2014)
- Naidoo, P.J., D.G. Anton and P. Jordan, 2014. School Evaluation for Quality Improvement. *International Institute for Educational Planning*. Retrieved form: <http://unesdoc.unesco.org/images/0013/001398/139804e.pdf>. (Accessed on: May 30, 2014)
- Romero, C. and S. Ventura, 2010. Educational data mining: A review of the state-of-the-art. *IEEE T. Syst. Man Cy. C*, 40: 601-618.
- Sonali, A., G.N. Pandey and M.D. Tiwari, 2012. Data mining in education: Data classification and decision tree approach. *Int. J. e-Educ. e-Bus. e-Learn.*, 2(2): 140-144.
- Steven, B., 2009. *Natural Language Processing with Python*. O'Reilly Media. Retrieved form: <http://shop.oreilly.com/product/9780596516499.doc>. (Accessed on: June 24, 2014)
- Sundar, P.V., 2013. A comparative study for predicting student's academic performance using Bayesian network classifiers. *IOSR J. Eng.*, 3(2): 37-42.
- Zaki, M.J. and W. Meira Jr., 2013. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Retrieved form: <http://www2.dcc.ufmg.br/livros/miningalgorithms/files/pdf/dmafca.pdf>. (Accessed on: May 30, 2014)