Jishan *et al. Decision Analytics* (2015) 2:1
DOI 10.1186/s40165-014-0010-2

Decision Analytics
a SpringerOpen Journal

RESEARCH Open Access

Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique

Syed Tanveer Jishan, Raisul Islam Rashu, Naheena Haque and Rashedur M Rahman*

* Correspondence:
rashedur@northsouth.edu
Department of Electrical and Computer Engineering, North South University, Plot-15, Block-B, Bashundhara, Dhaka 1229, Bangladesh

Abstract

There is a perpetual elevation in demand for higher education in the last decade all over the world; therefore, the need for improving the education system is imminent. Educational data mining is a newly-visible area in the field of data mining and it can be applied to better understanding the educational systems in Bangladesh. In this research, we present how data can be preprocessed using a discretization method called the Optimal Equal Width Binning and an over-sampling technique known as the Synthetic Minority Over-Sampling (SMOTE) to improve the accuracy of the students' final grade prediction model for a particular course. In order to validate our method we have used data from a course offered at North South University, Bangladesh. The result obtained from the experiment gives a clear indication that the accuracy of the prediction model improves significantly when the discretization and over-sampling methods are applied.

Keywords: Educational data mining (EDM); Classification; Naive Bayes; Decision tree; Neural network; Discretization; Equal width binning; Over-sampling; SMOTE; Class imbalance

Background

Educational Data Mining (EDM) is an interdisciplinary research area that fixates on the utilization of data mining in the educational field. Educational data can be from different sources, but generally from academic institutions, but nowadays, online learning systems are also the incipient environment for acquiring educational data which can be habituated to analyze and extract utilizable information (Romero & Ventura 2010). The goal of the research is to predict the students' performance using attributes such as Cumulative Grade Point Average, Quiz, Laboratory, Midterm and Attendance marks. However, in order to improve the prediction model we introduced some preprocessing techniques so that the prediction model provides with more precise results which could be used to alert students before the final examination regarding their final outcome.

We received the course data and student information from the North South University. After acquiring the data we preprocessed it and then applied three classification algorithms, e.g., Naïve Bayes, Decision Tree and Neural Network. In order to improve the model we looked into the techniques at the data preprocessing level. At first we

© 2015 Jishan; licensee Springer. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

discretized the continuous attributes using optimal equal width binning as proposed by Kayah (2008) and then used Synthetic Minority Over-Sampling (SMOTE) technique (Chawla et al. 2002) to increase the volume of the data, provided that there were limited instances in the acquired data. There are four forms of the preprocessed data: normal acquired data, data with discretization technique applied, class balanced data using oversampling and the data where both the discretization and oversampling methods were used. We build twelve models by preprocessing the data in four different ways mentioned and using three classification techniques mentioned earlier. After all the models were built we compared their accuracy, precision, recall and F-measure of the class labels for those models. ROC Curves for each of the models are generated and Area Under the Curves (AUC) are also calculated and compared.

Related works

Educational Data Mining is a vast domain which consists of different applications. Using data mining techniques it is possible to build course planning system, detecting what type of learner a student is, making group of similar types of students, predicting the performance of the students as well as helping instructors to get insight on how to commence the classes (Romero & Ventura 2010). Pal and Pal (2013) conducted studies at the VBS Purvanchal University, Jaunpur, India and used classification algorithms to identify the students who need special advising or counseling from the teachers.

Ayers et al. (2009) used several clustering algorithms such as hierarchical agglomerative clustering, K-means and model based clustering in order to understand skill levels of the students and group them based on their skill sets. Bharadwaj and Pal (2012) found that students' grade in the senior secondary exam, living location, medium of teaching, mother's qualification, family annual income, and student's family status are correlated strongly and help to predict how the students perform academically. In another study Bharadwaj and Pal (2011) used students' previous semester marks, class test grade, seminar performance, assignment performance, general proficiency, attendance in class and lab work to predict the end of the semester marks.

A comparison of machine learning methods has been carried out to predict success in a course (either passed or failed) in Intelligent Tutoring Systems (Hämäläinen & Vinni 2006). Nebot et al. (2006) used different types of rule-based systems have been applied to predict student performance such as mark prediction in an e-learning environment using fuzzy association rules. Several classification algorithms have been applied in order to group students, such as: discriminant analysis, neural networks, random forests and decision trees for classifying university students into three groups such as low-risk, medium-risk and high-risk of failing (Superby et al. 2006).

Zhu et al. (2007) explains how making a personalized learning recommendation system which will help the learner beforehand what he or she should learn before moving to the next step. Yadav et al. (2012) used students' attendance, class test grade, seminar and assignment marks, lab works to predict students' performance at the end of the semester. They used the decision tree algorithms such as ID3, CART and C4.5 and made a comparative analysis. In their study, they achieved 52.08%, 56.25% and 45.83% accuracy of each of these classification techniques respectively.

Prati et al. (2004) discussed about recent works in the field of data mining to overcome the imbalanced dataset problem. They mainly focused in concepts and methods to deal with imbalanced datasets. Chawla et al. (2002) found that majority class and minority class both have to equally represent in classification category for balanced dataset. They used combination of the method of over sampling the minority class and under sampling the majority class to accomplish the better classifier performance in ROC space. They mainly introduced the Synthetic Minority Over-sampling approach which provides the new technique in over sampling and intercourse with the under sampling makes the better result.

Chen (2009) used several re-sampling techniques for finding the maximum accuracy of classification from fully labeled imbalanced training data set. SMOTE, Oversampling by duplicating minority examples, random under sampling, is mainly used to create new training data set. Standard classifiers like Decision Tree, Naive Bayes, Neural Network are trained in this data set and all the techniques show improved accuracy except Naive Bayes. Rahman and Davis et al. (2013) tried to address class imbalance issue in medical datasets. They used undersampling techniques as well as oversampling techniques like SMOTE to balance the classes.

There are some works done using Neural Network to predict students' grade. Gedeon and Turner (1993) compared different types of neural network models which have been used to predict final student grades primarily; they mainly used backpropagation and feedforward neural networks. Want and Mitrovic (2002) used feedforward and backpropagation to predict the number of errors a student will make. Oladokun et al. (2008) used multilayer perceptron topology for predicting the likely performance of a candidate being considered for admission into the university.

We can notice that there is handful of works on grade prediction models, however, our focus was to address the issue of class imbalance and discretizing the continuous attributes effectively instead of taking an assumption such as, normal distribution. The primary goal was to observe whether synthetic minority oversampling method and optimum equal width binning together will result in better performance of the grade prediction models provided that most of the attributes in course mark sheets or data sets are continuous in nature and the number of instances were low.

Methods

Data selection

The dataset we are using contains 181 instances which is the number of students enrolled in the course during the prior 18 months. This dataset is from a course titled "Numerical Analysis" which is a core course in EEE disciple in North South University, Dhaka, Bangladesh. Originally the dataset had student ID, student name, five quiz marks, midterm marks, attendance, laboratory marks, final marks and final grade as attributes. We have selected the attribute which contains the percentage of marks obtained by the students in quizzes rather than taking all the quizzes into account. Final grade is considered as the class label. The same dataset is used for creating the over-sampled dataset where the number of instances is 360.

Data preparation

At first we discarded the Students' ID in the dataset provided that it is not directly required for classification. Students' CGPA, which was not initially a part of the dataset, it was retrieved and added as an attribute. All the attributes which are used for classification are listed in the Table 1.

Balancing the dataset using synthetic minority over-sampling

SMOTE (Synthetic Minority Oversampling Technique) is an over-sampling technique which is used to overcome the problem of imbalanced dataset. SMOTE modifies an imbalanced dataset and generates a balanced dataset from the imbalanced dataset. SMOTE distributes the instances of the majority class and the minority class equally. SMOTE technique increases the predictive accuracy over the minority class by creating synthetic instances of that minority class. SMOTE does not overfit largely because it uses synthetic sampling technique. There are opportunities for inductive learners like decision tree or rule-learner to extend their decision regions for the minority class. As a consequence, in the field of imbalance data classification problem, a better performance can be easily achieved (Chawla et al. 2002). In SMOTE, the minority class is over-sampled by introducing synthetic instances where each minority class sample is taken. The instances are inserted along the line segments joining any or all of the k-nearest neighbors of the minority class. Neighbors are randomly chosen from k-nearest neighbors depending upon the amount of over-sampling that is required. Five nearest neighbors are currently used in the implementation of SMOTE. For instances, if the size of over-sampling is 200%, then only two neighbors from the five nearest neighbors are chosen and one synthetic sample is generated in each direction (Chen 2009). In short the SMOTE algorithm can be stated in the steps as, taking the difference between the feature vector (minority class example) under consideration and its nearest neighbor (minority class examples) and then multiplying this difference by a random number between 0 and 1. Furthermore, adding the difference calculated in previous step to the feature vector as a result creating a new feature vector, the idea is represented as equation in (1).

$$x_{new} = x_i + \left(\hat{x}_i - x_i\right) \times \delta \qquad (1)$$

In the equation,

Table 1 Attributes of the dataset

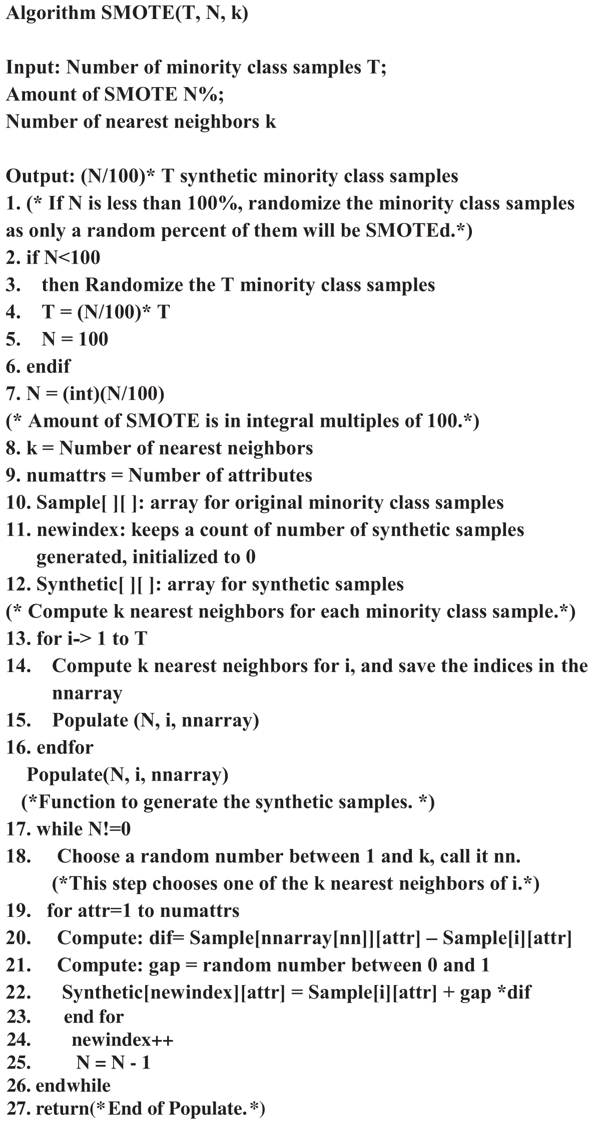
Attributes	Remarks
CGPA	Cumulative Grade Point Average. It ranges from 0.00 to 4.00. This is a measure to evaluate students' past record
Quiz marks	Best 4 out of 5 quizzes are counted as per the course policy which was intact throughout the five semesters. The average is taken and is normalized between 0 to 100.
Midterm marks	Number of midterm examination differed between 1–2 among all the semesters taken into consideration. For the semesters where two midterms were held, average of them is taken. The data is then normalized between 0 to 100.
Laboratory mark	Weight of the laboratory marks varied from semester to semester, therefore, the marks are normalized between 0 to 100.
Attendance marks	Ranges from 0 to 100
Final grade	This is label our classification models will try to predict. final grade consist of five classes: A,B,C,D,F.

$$x_i \ = \ \text{the feature vector (minority class example) under consideration}$$

$$\hat{x}_i \ = \ \text{oneof the k-nearest neighbors for } \mathrm{x_i}$$

$$\delta = \ \textit{random number between } [0, 1]$$

More details of the SMOTE algorithm could be found in Rahman and Davis et al. (2013), a short description is given below:

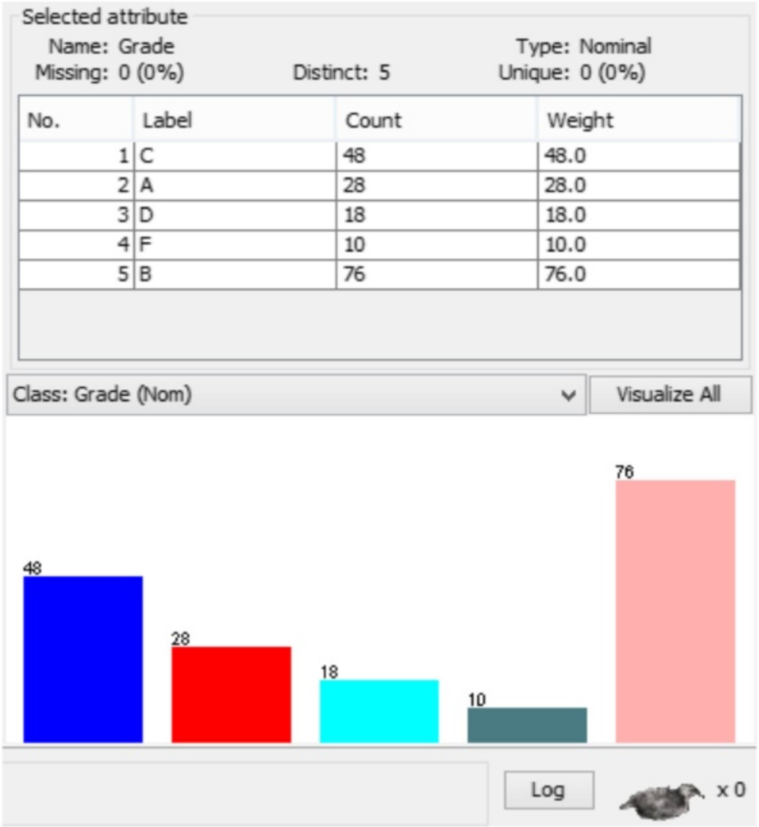


```
Algorithm SMOTE(T, N, k)

Input: Number of minority class samples T;
Amount of SMOTE N%;
Number of nearest neighbors k

Output: (N/100)* T synthetic minority class samples
(* If N is less than 100%, randomize the minority class samples
as only a random percent of them will be SMOTEd.*)
if N<100
    then Randomize the T minority class samples
    T = (N/100)* T
    N = 100
endif
N = (int)(N/100)
(* Amount of SMOTE is in integral multiples of 100.*)
k = Number of nearest neighbors
numattrs = Number of attributes
Sample[ ][ ]: array for original minority class samples
newindex: keeps a count of number of synthetic samples
    generated, initialized to 0
Synthetic[ ][ ]: array for synthetic samples
(* Compute k nearest neighbors for each minority class sample.*)
for i-> 1 to T
    Compute k nearest neighbors for i, and save the indices in the
    nnarray
    Populate (N, i, nnarray)
endfor
  Populate(N, i, nnarray)
 (*Function to generate the synthetic samples. *)
while N!=0
    Choose a random number between 1 and k, call it nn.
    (*This step chooses one of the k nearest neighbors of i.*)
  for attr=1 to numattrs
    Compute: dif= Sample[nnarray[nn]][attr] – Sample[i][attr]
    Compute: gap = random number between 0 and 1
     Synthetic[newindex][attr] = Sample[i][attr] + gap *dif
     end for
      newindex++
       N = N - 1
endwhile
return(*End of Populate.*)
```

The data mining software Weka was used for implementing the SMOTE over-sampling technique. The over-sampled data is then randomized twice for class balancing.

In Figure 1, the original count for each of the samples are provided. We can observe imbalance among the class labels. For example, class B contains 76 instances but class F contains 10 instances.

Figure 2 represents the number of counts for each of the class labels once the class imbalance issue was solved using SMOTE. For example, we can observe that class B



Selected attribute

Name: Grade Type: Nominal

Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

No.	Label	Count	Weight
1	C	48	48.0
2	A	28	28.0
3	D	18	18.0
4	F	10	10.0
5	B	76	76.0

Class: Grade (Nom)

Figure 1 Class distribution of data before re-sampling with SMOTE in Weka.

contain 76 instances but class F now contains 70 instances. This can be considered as a significant improvement between each class labels.

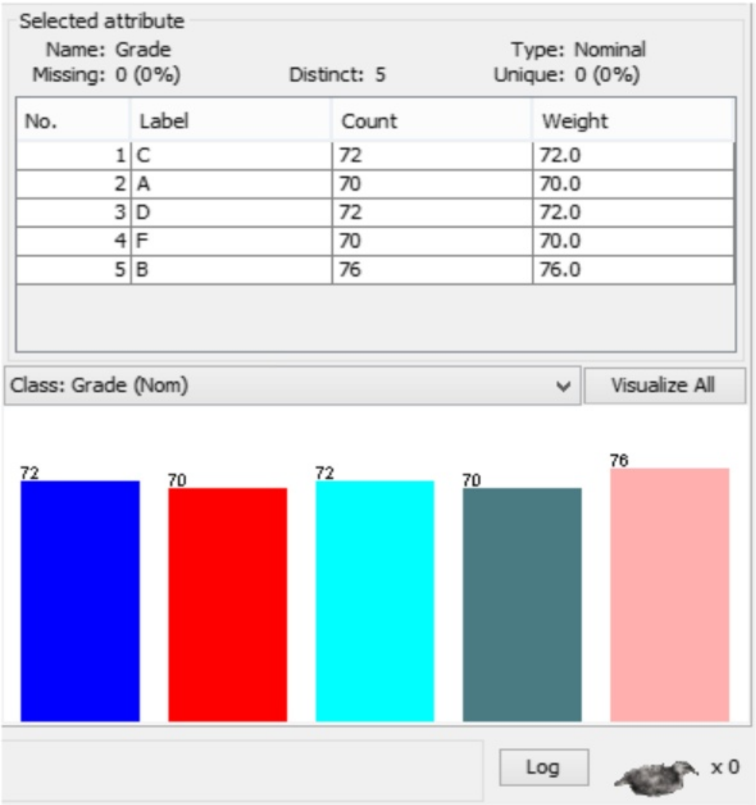
Weka (Holmes et al. 1994) is a open source software designed to carry out data analysis. It is widely used for machine learning and data mining purposes.

Handling continuous data using probability distribution function

The most common type of continuous data we usually come across fits the Gaussian Distribution which is stated in (2). The Gaussian Distribution Function (Tan et al 2006) is a bell shaped density function having the center representing the mean value. One of the disadvantages of using such estimation is that the data distribution density may not coincide at all with the Gaussian Distribution Function, as a result, the accuracy of the model can be poor. In this equation $\boldsymbol{A_i}$ is the i^{th} instance of the attribute A and c_j is the j^{th} class label. The symbol, μ, stands for the population mean and the symbol, σ^2, stands for variance of the given population.

$$P(A_i|c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{\left(A_i-\mu_{ij}\right)^2}{2\sigma_{ij}^2}} \quad (2)$$

There are uses of probability distribution function on the continuous attributes in the dataset in the model built using Naive Bayes classification.



Selected attribute

Name: Grade Type: Nominal
Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

No.	Label	Count	Weight
1	C	72	72.0
2	A	70	70.0
3	D	72	72.0
4	F	70	70.0
5	B	76	76.0

Class: Grade (Nom)

Figure 2 Class distribution of data after re-sampling with SMOTE in Weka.

Handling continuous data using optimal equal width binning

Data binning is a method of splitting continuous data into small intervals. There are several methods of creating bins. The method we are using is called Equal Width Binning where each interval has the same length. However, randomly selecting the bin width value may not provide us with better accuracy. Therefore, we have implemented a discretization technique proposed by Kayah (2008) which is based on equal width binning and error minimization. According to that paper, for a continuous

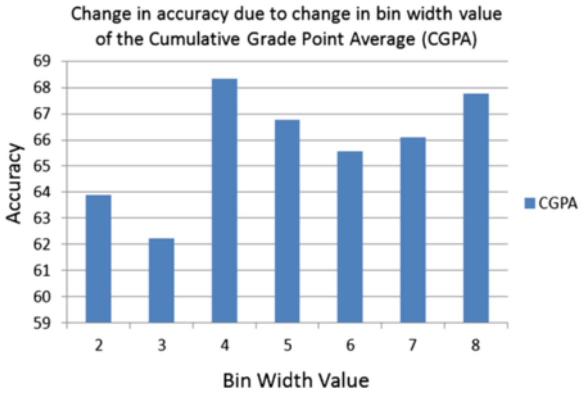


Figure 3 Accuracy fluctuation due to bin width values.

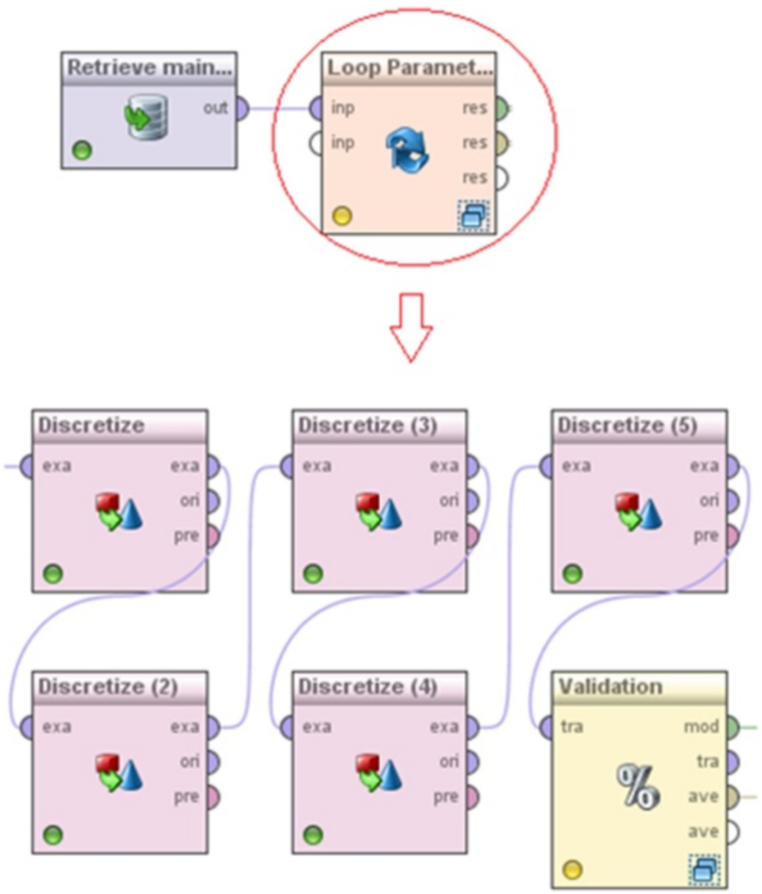
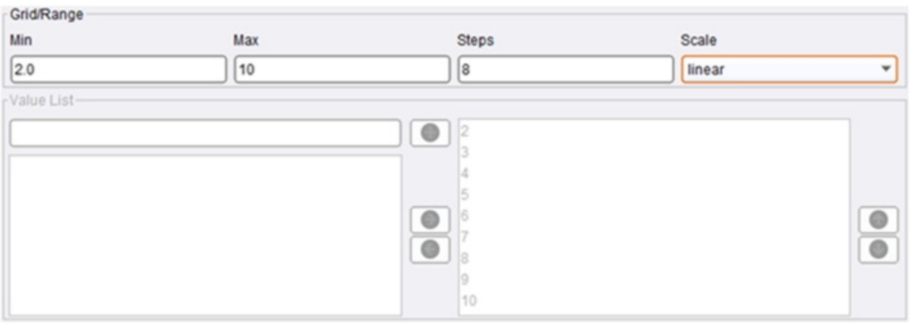


Figure 4 Iteration through the attribute bin values for each discretization blocks.

attribute we will dynamically search for the bin width value until we find the optimal one. The dynamic searching indicates that we need to use iteration in order find the optimal bin width value. Moreover, data sets can have more than one continuous attribute, if the attributes are independent of each other, then finding optimal bin width value for all the continuous attributes in the data set will result in better overall performance. In Figure 3 a bar graph is shown which represents how different bin width values for the attribute Cumulative Grade Point Average affects



Grid/Range

Min: 2.0

Max: 10

Steps: 8

Scale: linear

Value List

2
3
4
5
6
7
8
9
10

Figure 5 Setting the iteration steps for each discretization blocks.

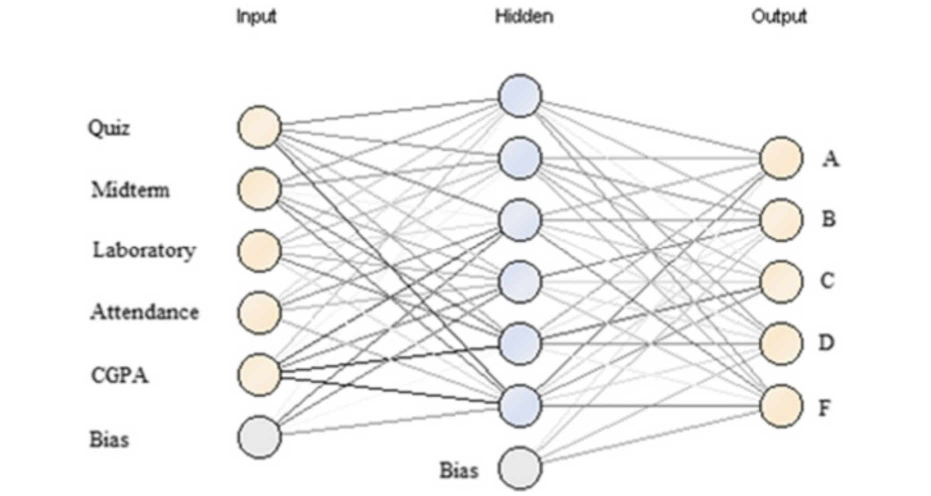


Figure 6 Design of the neural network.

the accuracy of Naive Bayes Classification model when optimal equal width binning is used on normal data. On the *x*-axis, bin width values are provided and on the *y*-axis, accuracy of the classifier is provided. We can observe that when the bin width value is set to 4 we get the highest accuracy thus it is the optimal bin width value.

In order to find the optimal bin values for each of the attributes for a particular model we used loop operation in RapidMiner 5. We iterate through each bin width value for each attribute discretization block shown in Figure 4.

Each discretization block is assigned for one attribute, therefore we had five blocks. The condition for the loop is shown in the Figure 5. The min value represents the

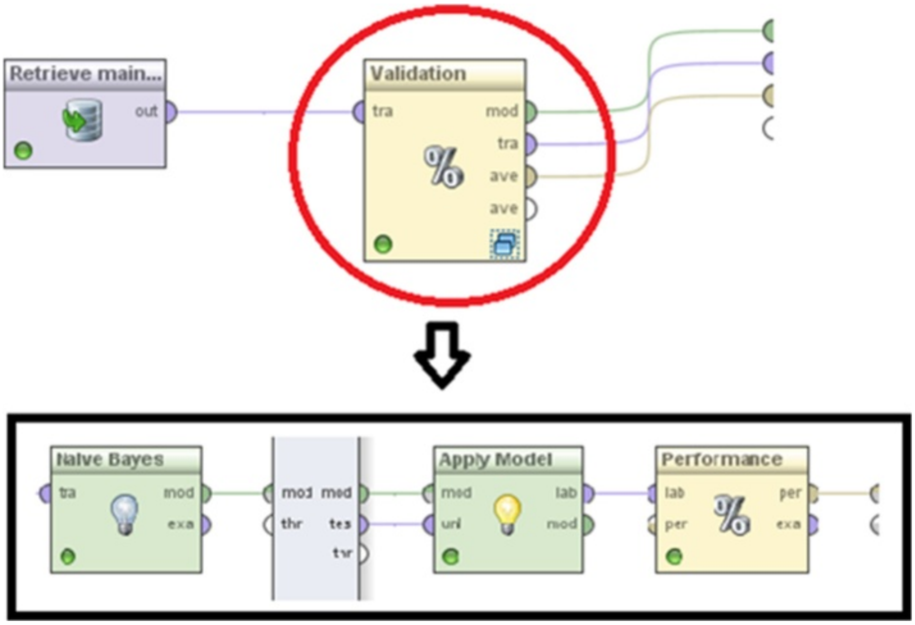


Figure 7 Building a classification model in RapidMiner 5.

starting iteration value and the max value represents the last iteration value. The step value indicates the number of steps that should be taken between min and max value.

Naive Bayes for classification

Naive Bayes classifier (Tan et al. 2006) is a probabilistic classifier based on applying Bayes' theorem. Naive Bayes assumes that all the attributes which will be used for classification are independent of each other. We used Naïve Bayes Classification to create four different models. In the first model we estimated the class labels for continuous attributes using probability distribution function (PDF), in the second model we used optimal equal binning width value, for the third model we over-sampled the data using SMOTE and for the fourth model we used both optimal equal width binning and SMOTE.

C4.5 algorithm for classification

C4.5 is an extended version of Iterative Dichotomiser 3 decision tree algorithm. In this algorithm, we need to calculate entropy of every attribute of the dataset and then we have to split the data set into subsets using the attributes of minimum entropy or maximum information gain. Some of the major extensions of C4.5 from ID3 is that it accepts both continuous and discrete features, handles incomplete data points and different weights can be applied on the features that comprise the training data (Quinlan 1993). We split the data using gain ratio and minimal size for the split was set to 4. Therefore, nodes where the number of subsets is greater than or equal to 4 will be split.

Backpropagation algorithm for classification

Backpropagation is a method of artificial neural network. It is used along with an optimization method called gradient descent. The Backpropagation algorithm is divided into two phases: propagation and weight update (Haykin 2008).

In our model, which is shown in Figure 2, we used 3 layers of neurons: input layer, hidden layer and output layer. In input layer, the numbers of neurons are 6. They are basically attributes of the data set, such as Quiz, Midterm, Laboratory, Attendance, CGPA and one extra bias. In output layers, the number of neurons are 5 and they represent the class label of the course grade. In hidden layer, the numbers of neurons are 6 with one extra bias which makes the total number of neurons to 7. The number of neurons for hidden layer is calculated using the equation (3). The training of the Neural Network was done for 350 cycles with a learning rate of 0.1 and momentum 0.17. The Neural Network model for the grade prediction system is shown in the Figure 6.

$$No.ofNeurons = \frac{No.ofAttributes + No.ofClasses}{2} + 1 \tag{3}$$

Implementation of the models

All the models are building using RapidMiner 5 which is a data mining tool. For the original data we import the Microsoft Excel file into RapidMiner and then preprocessed it. During this process, we selected the grade attributes as the response variable and considered it as nominal type whereas the rest of the attributes are considered as numeric type. For the over-sampled data we used Weka for SMOTE before importing the Microsoft Excel file into RapidMiner. After importing data is further preprocessed in RapidMiner, then we used the validation system in which the classification model is build. The process blocks are presented in the Figure 7. At the top left corner we have the data block which is connected to validation block. Inside the validation block the data from the data block is connected to the classifier. For this example the classification block is Naive Bayes. The Apply Model block applies an already learnt trained model on the testing data set. This learnt model is actually built by the classification block. After the model is applied we use the Performance block which calculates the accuracy of the trained model. For each block "tra" stands for training dataset, "mod" stands for model, "lab" means labeled data, likewise performance of the classifier is denoted as "per".

Results

We have split the data into two portions, the training data and the testing data. The training data consists of about 80% of the original data and the testing data consist of about 20% of the original data. Table 2 represents the optimal equal bin width value of the attributes for each of the classifiers on the original data and Table 3 represents the optimal equal bin width value of the attributes for each of the classifiers on the over-sampled data. A loop operation has been used each attributes to look for the optimum bin width value as mentioned beforehand. We made two observations from the iteration. Firstly, as the size of the data increases the optimal equal bin width value also increases. Secondly, it is not true for two attributes which are: Attendance Marks and Cumulative Grade Point Average (CGPA) where the bin width value stays almost the same for all the classifiers. Stratified Sampling which is a method of sampling is used because it gives better coverage of the whole population. The final accuracy of the model is measured by taking the average of the five iterations.

Naive Bayes classification

Tables 4-7 represent the models build using Naive Bayes Classification. When probability distribution function is used to handle the continuous data the accuracy of the model is about 61.11%. However, when we introduce optimal equal width binning for discretizing the continuous data the accuracy increases about 7%. When we balance the classes using SMOTE oversampling method and use probability distribution function on the continuous data the accuracy is almost 67%. When we use optimal equal width binning on the over-sampled data then the accuracy rises up to 75%. From the Table 4 we can see that the class D and F have very low precision and recall which is a indication of

Table 2 Bin width values for the classification methods

	Decision tree	Naïve Bayes	Neural network
Quiz	3	7	6
Midterm	7	6	8
Laboratory	5	6	4
Attendance	2	2	2
CGPA	4	4	4

Table 3 Bin width values for the classification methods on SMOTE over-sampled data

	Decision tree	Naïve Bayes	Neural network
Quiz	8	8	8
Midterm	3	9	8
Laboratory	5	6	6
Attendance	2	2	2
CGPA	6	4	4

Table 4 Detailed analysis of the naive Bayes model

	True C	True A	True D	True F	True B	Class precision
Pred. C	33	0	8	2	16	55.93%
Pred. A	0	22	0	0	15	59.46%
Pred. D	6	0	9	6	0	42.86%
Pred. F	0	0	1	1	0	50.00%
Pred. B	9	6	0	1	45	73.77%
Class recall	68.75%	78.57%	50.00%	10.00%	59.21%	
F-measure	61.68%	67.69%	46.15%	16.66%	65.69%	

Table 5 Detailed analysis of the naive Bayes model with optimal equal width binning

	True C	True A	True D	True F	True B	Class precision
Pred. C	34	0	6	0	13	64.15%
Pred. A	0	22	0	0	7	75.86%
Pred. D	6	0	9	5	1	42.86%
Pred. F	1	0	3	3	0	42.86%
Pred. B	7	6	0	2	55	78.57%
Class recall	70.83%	78.57%	50.00%	30.00%	72.37%	
F-measure	67.32%	77.19%	46.15%	35.29%	75.34%	

Table 6 Detailed analysis of the naive Bayes model with SMOTE oversampling

	True D	True C	True F	True A	True B	Class precision
Pred. D	64	15	35	0	1	55.65%
Pred. C	8	47	14	0	13	57.32%
Pred. F	0	0	19	0	0	100.00%
Pred. A	0	0	0	65	17	79.27%
Pred. B	0	10	2	5	45	72.58%
Class recall	88.89%	65.28%	27.14%	92.86%	59.21%	
F-measure	68.44%	61.04%	42.69%	85.52%	65.21%	

Table 7 Detailed analysis of the naive Bayes model with optimal equal width binning and SMOTE oversampling

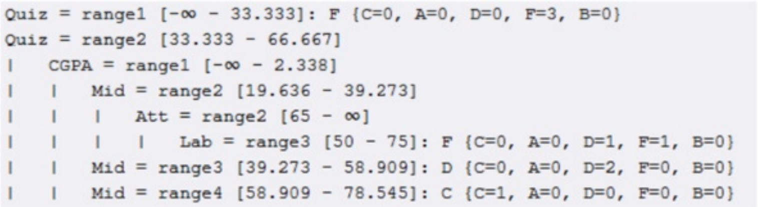
	True D	True C	True F	True A	True B	Class precision
Pred. D	56	12	9	0	2	70.89%
Pred. C	7	49	3	0	14	67.12%
Pred. F	9	4	56	0	2	78.87%
Pred. A	0	0	0	65	13	83.33%
Pred. B	0	7	2	5	45	76.27%
Class recall	77.78%	68.06%	80.00%	92.86%	59.21%	
F-measure	74.17%	67.58%	79.43%	87.83%	66.66%	

misclassification. The problem arises as there are less number of instances of Class D and F. Once we address the class imbalance issue the precision and recall of these classes improves significantly as we can notice in the Table 6. From the Tables 6 and 7, we can observe that when we introduce optimal equal width binning on the over-sampled data the number of instances of class F accurate predicted increase from 19 to 56.

Decision tree classification

Figure 8 represents a portion of the decision tree model build after the data has been discretized using optimal equal width binning. From the figure we can derive the rules required to determine the students' grades. For Example, if a student having CGPA below 2.3, ends up getting between 33.3% to 66.7% in Quiz and less than 40% in Midterm and if the student gets between 50% to 75% in Laboratory he or she is mostly likely to get D as overall grade. Another rule that we can observe is that if the Quiz marks are below 33%, no matter what is the CGPA of the student, he or she will fail in the course. From the decision tree in Figure 4 we can understand that the attribute Quiz had the highest information gain which is then followed by CGPA.

Tables 8-11 show the detailed analysis of the models generated using Decision Tree. Decision Tree models are comparatively less accurate than the models build using Naive Bayes classification and Neural Network. Just like the Naive Bayes models there are issues related to precision and recall for Decision Tree models which is solved when SMOTE oversampling technique was used. However, when Decision Tree classification was used on over-sampled data, the outcome of the Class D and F is better than the outcome of these classes when Naive Bayes Model was used. The F-measure for the Class F is 84.67% as we can see in the Table 11 whereas the F-



```
Quiz = range1 [-∞ - 33.333]: F {C=0, A=0, D=0, F=3, B=0}
Quiz = range2 [33.333 - 66.667]
|   CGPA = range1 [-∞ - 2.338]
|   |   Mid = range2 [19.636 - 39.273]
|   |   |   Att = range2 [65 - ∞]
|   |   |   |   Lab = range3 [50 - 75]: F {C=0, A=0, D=1, F=1, B=0}
|   |   Mid = range3 [39.273 - 58.909]: D {C=0, A=0, D=2, F=0, B=0}
|   |   Mid = range4 [58.909 - 78.545]: C {C=1, A=0, D=0, F=0, B=0}
```

Figure 8 Decision tree model when using optimal equal width binning.

Table 8 Detailed analysis of the decision tree model

	True C	True A	True D	True F	True B	Class precision
Pred. C	29	0	14	7	11	47.54%
Pred. A	0	25	0	0	20	55.56%
Pred. D	2	0	1	1	0	25.00%
Pred. F	0	0	1	2	0	66.67%
Pred. B	17	3	2	0	45	67.16%
Class recall	60.42%	89.29%	5.56%	20.00%	59.21%	
F-measure	53.21%	68.49%	9.09%	30.76%	62.93%	

Table 9 Detailed analysis of the decision tree model with optimal equal width binning

	True C	True A	True D	True F	True B	Class precision
Pred. C	33	1	11	3	17	50.77%
Pred. A	0	19	0	0	4	82.61%
Pred. D	9	0	5	3	1	27.78%
Pred. F	1	0	1	4	0	66.67%
Pred. B	5	8	1	0	54	79.41%
Class recall	68.75%	67.86%	27.78%	40.00%	71.05%	
F-measure	58.40%	74.51%	27.78%	50.00%	74.99%	

Table 10 Detailed analysis of the decision tree model with SMOTE oversampling

	True D	True C	True F	True A	True B	Class precision
Pred. D	63	23	16	0	6	58.33%
Pred. C	4	33	6	1	8	63.46%
Pred. F	5	4	47	0	0	83.93%
Pred. A	0	1	0	58	8	86.57%
Pred. B	0	11	1	11	54	70.13%
Class recall	87.50%	45.83%	67.14%	82.86%	71.05%	
F-measure	69.99%	53.22%	74.60%	84.67%	70.58%	

Table 11 Detailed analysis of the decision tree model with optimal equal width binning and SMOTE oversampling

	True D	True C	True F	True A	True B	Class precision
Pred. D	59	23	8	0	7	60.82%
Pred. C	5	39	2	1	14	63.93%
Pred. F	7	2	58	0	0	86.57%
Pred. A	0	0	0	63	20	75.90%
Pred. B	1	8	2	6	35	67.31%
Class recall	81.94%	54.17%	82.86%	90.00%	46.05%	
F-measure	69.81%	58.64%	84.67%	82.35%	54.68%	

measure for the Class F when Naive Bayes Model was used is 79.43% presented in Table 7.

Classification using neural network

Table 12-15 represent the detailed analysis of the models generated using the Neural Network. Although building Neural Network models are computationally slow compared to other models but it provided better results compared to other models. Before balancing the class using SMOTE Neural Network models failed to predict any instance of Class F. However, after balancing the class, precision and recall of Class F increased significantly. The accuracy gained when Backpropagation algorithm is used on balanced data is slightly greater than 75%.

Receiver operating characteristic (ROC) curve comparisons

ROC curve which stands for receiver operating characteristic curve is the graphical representation of the performance of the binary classifier system for varying discrimination threshold (Tan et al. 2006). The horizontal axis represents the fraction of false positives out of total actual negatives (FPR = False positive rate) and the vertical axis represents the fraction of true positives out of total actual positives (TPR = True positive rate).

Since ROC curve is a binary classifier system but we have five class labels for the grade so we are presenting five ROC curves. For each ROC curve one class is considered as True class and the rest of the classes are considered as False class. ROC curves change when over-sampled data was used for classification which are discussed in the Section 4.5.

In Figure 9, the ROC curve of the Neural Network Model where the continuous data are discretized using optimal equal width binning is represented along with other the curves of the other models. The area under the ROC curve is 0.985 for this model whereas the area uder the ROC curve for Naive Bayes using probability distribution function is about 0.93. This indicates the first model mentioned has better True Positive coverage for class A.

For Class B largest area under the curve is of the Naive Bayes Classification Model have the area under the curve which is 0.8335. Optimal Neural Network Classification Model is sandwiched between the Neural Network Classification model and the optimized Naive Bayes Classification Mode. The ROC curves are shown in Figure 10.

For the class C, Naive Bayes Classification Model where the continuous attributes are discretized using optimal equal width binning has the under the curve of around 81% coverage of the total area. Area under the ROC curves for other models are roughly 25-30% less compare to this model However AUC (area under the curve) for optimized ID3 is about 52% whereas for the ID3 model it is exactly 47%. The ROC curves are shown in the Figure 11.

Figure 12, represents the ROC curve of Neural Network Classification Model having area under the curve of about 0.93. This ROC curve is representing the class D and the ROC curve of Neural Network Classification Model where the continuous data are discretized using optimal equal width binning is the second best to that model for this class.

Table 12 Detailed analysis of the neural network model

	True C	True A	True D	True F	True B	Class precision
Pred. C	33	0	10	2	14	55.93%
Pred. A	0	18	0	0	6	75.00%
Pred. D	2	0	8	6	0	50.00%
Pred. F	0	0	0	0	0	0.00%
Pred. B	13	10	0	2	56	69.14%
Class recall	68.75%	64.29%	44.44%	0.00%	73.68%	
F-measure	61.68%	69.23%	47.05%	0.00%	71.33%	

Table 13 Detailed analysis of the neural network model with optimal equal width binning

	True C	True A	True D	True F	True B	Class precision
Pred. C	38	0	7	1	13	64.41%
Pred. A	0	18	0	0	5	78.26%
Pred. D	4	0	11	7	1	47.83%
Pred. F	0	0	0	0	0	0.00%
Pred. B	6	10	0	2	57	76.00%
Class recall	79.17%	64.29%	61.11%	0.00%	75.00%	
F-measure	71.03%	70.59%	53.66%	0.00%	75.49%	

Table 14 Detailed analysis of the neural network model with SMOTE oversampling

	True D	True C	True F	True A	True B	Class precision
Pred. D	53	12	6	0	3	71.62%
Pred. C	7	44	4	0	10	67.69%
Pred. F	12	6	59	0	0	76.62%
Pred. A	0	0	0	58	12	82.86%
Pred. B	0	10	1	12	51	68.92%
Class recall	73.61%	61.11%	84.29%	82.86%	67.11%	
F-measure	72.60%	64.23%	80.27%	82.86%	68.00%	

Table 15 Detailed analysis of the neural network model with optimal equal width binning and SMOTE oversampling

	True D	True C	True F	True A	True B	Class precision
Pred. D	52	10	3	0	4	75.36%
Pred. C	10	50	6	0	11	64.94%
Pred. F	10	4	59	0	0	80.82%
Pred. A	0	0	1	57	8	86.36%
Pred. B	0	8	1	13	53	70.67%
Class recall	72.22%	69.44%	84.29%	81.43%	69.74%	
F-measure	73.75%	67.11%	82.51%	83.82%	70.20%	

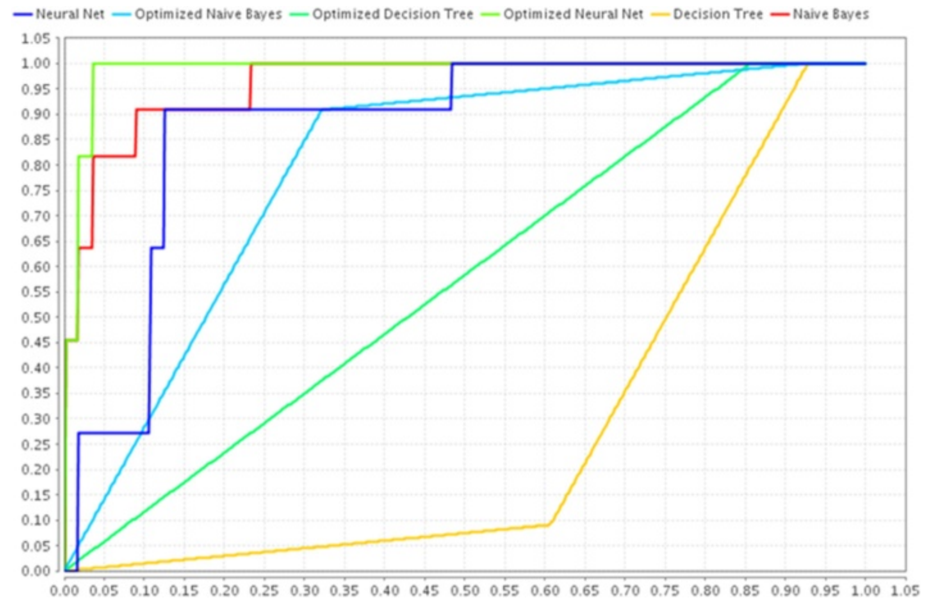


Figure 9 ROC curves for the class A.

ROC Curve for the class label F is represented in the Figure 13. Optimized Naive Bayes Classification Model is having area under the curve of about 85% of the whole area. Naive Bayes Classification Model is having an area of about 70% which is very close to Neural Network Classification Model having AUC of about 69%.

ROC curve comparisons after oversampling using SMOTE

Figure 14 shows the ROC Curve for the Class A for the models build with oversampled data. When optimal equal width binning is not used the area under the ROC

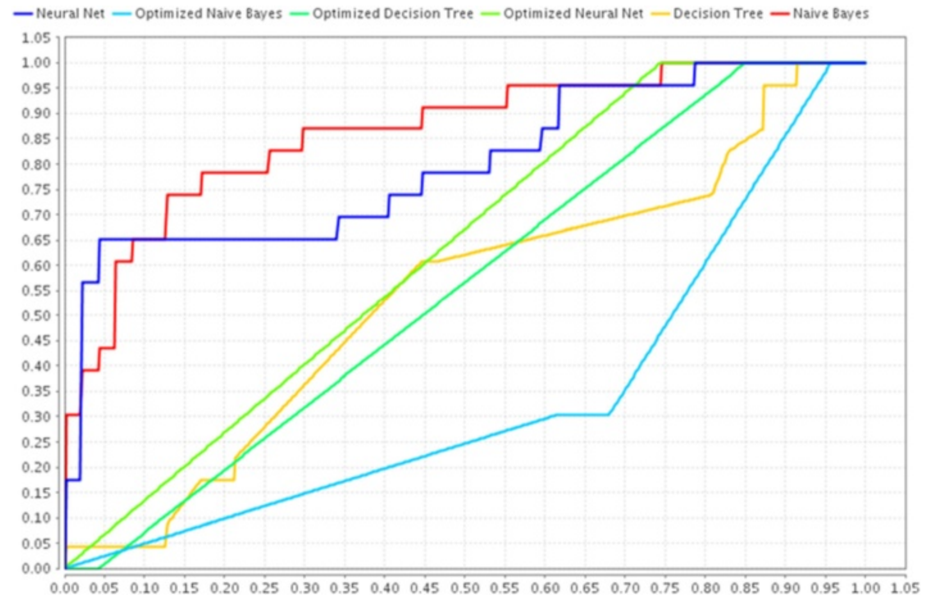


Figure 10 ROC curves for the class B.

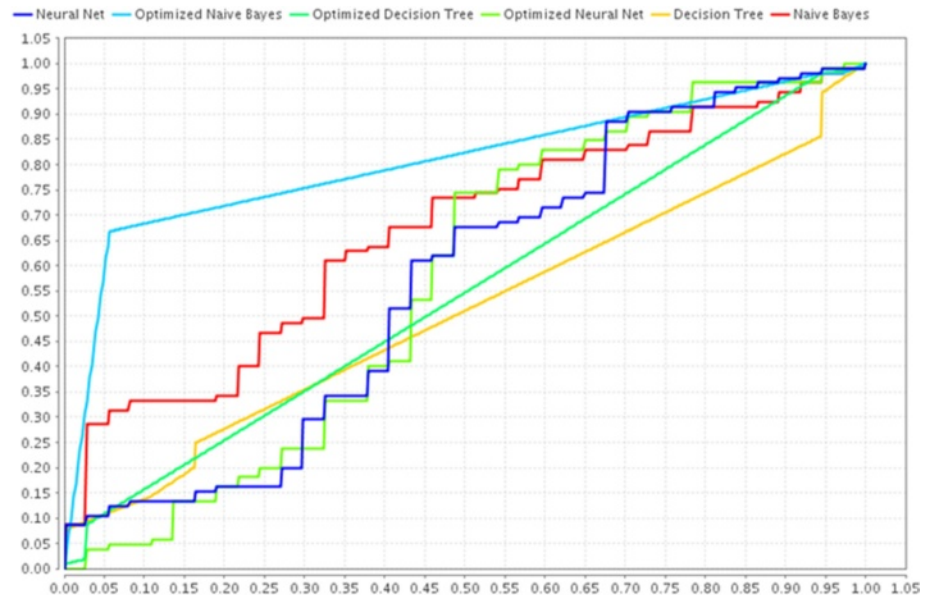


Figure 11 ROC curves for the class C.

curve is more than 85% for both the Naive Bayes and Neural Network classifiers. This also holds for the Decision Tree Models, however, Decision Tree Models have significantly low AUC compares to other models mentioned earlier.

While comparing Figure 10 and Figure 15 we can notice that there is noticeable gain in terms of area under the ROC curves of the models for class B. However, for the Decision Tree models area under the ROC curve stays in between 60-65% even after using class balanced data. Neural Network models having the highest area under the curve compared to other models for the Class B as shown in Figure 15. This indicates

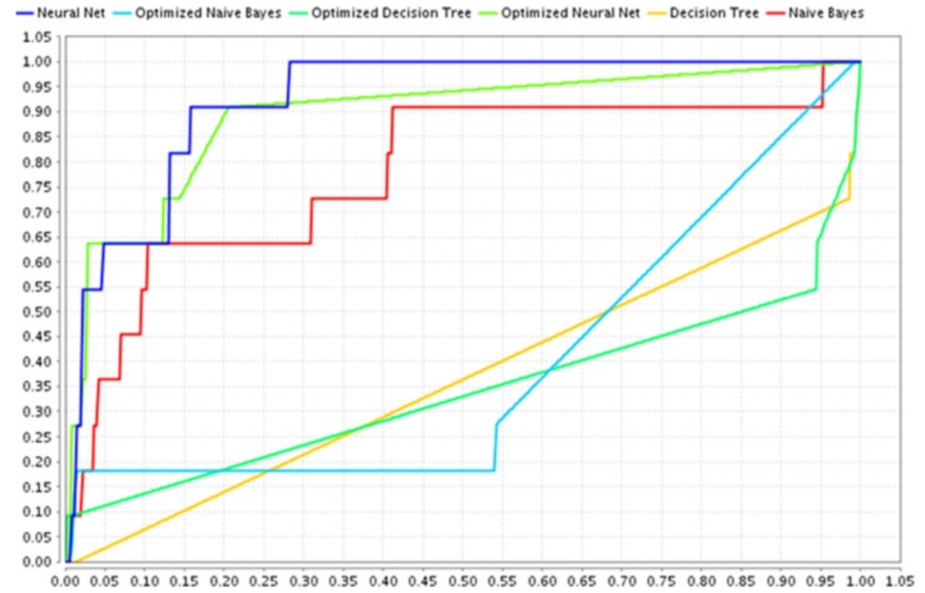


Figure 12 ROC curves for the class D.

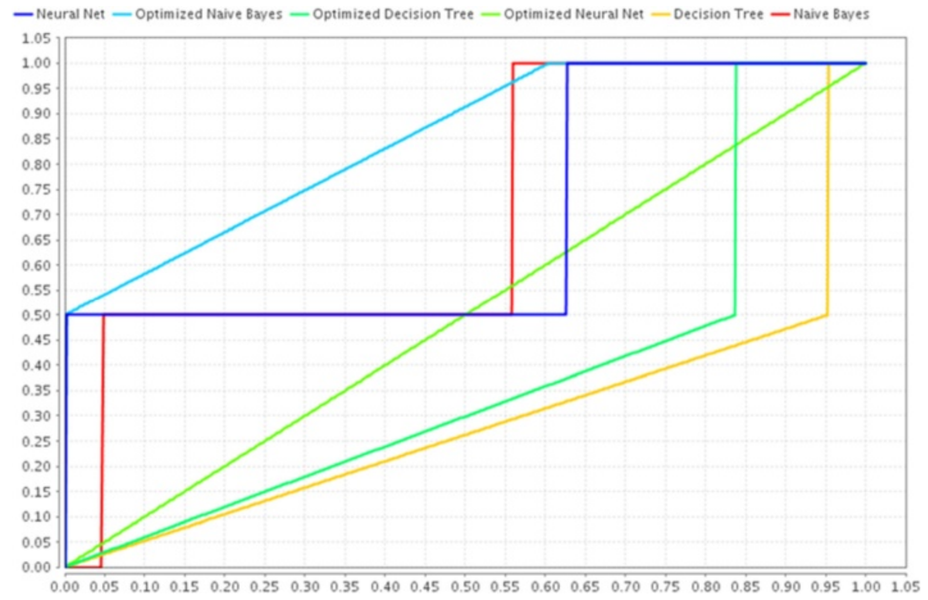


Figure 13 ROC curves for the class F.

that Neural Network models are more likely to choose positive instance, which for this case is Class B, than any other models.

Neural Network and Naive Bayes classifier models again prevailed for the Class C as we can see in the Figure 16. However this is not true for the Naive Bayes model where continuous data is being discretized using optimal equal width binning. Decision Tree models cover most area under the curve for this class compared to the area under the curve covered in other classes by these models.

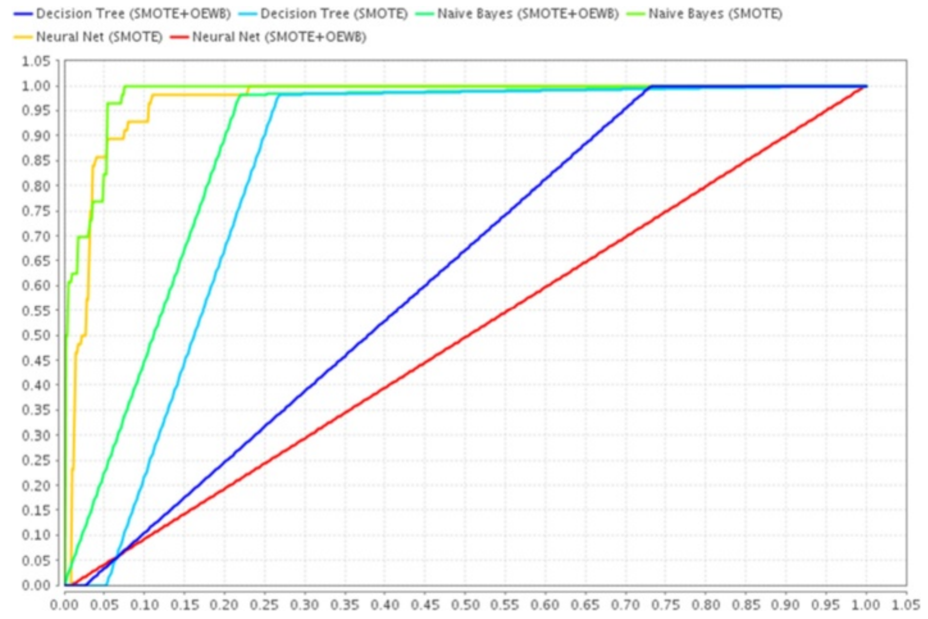


Figure 14 ROC curves for the class A (SMOTE Data).

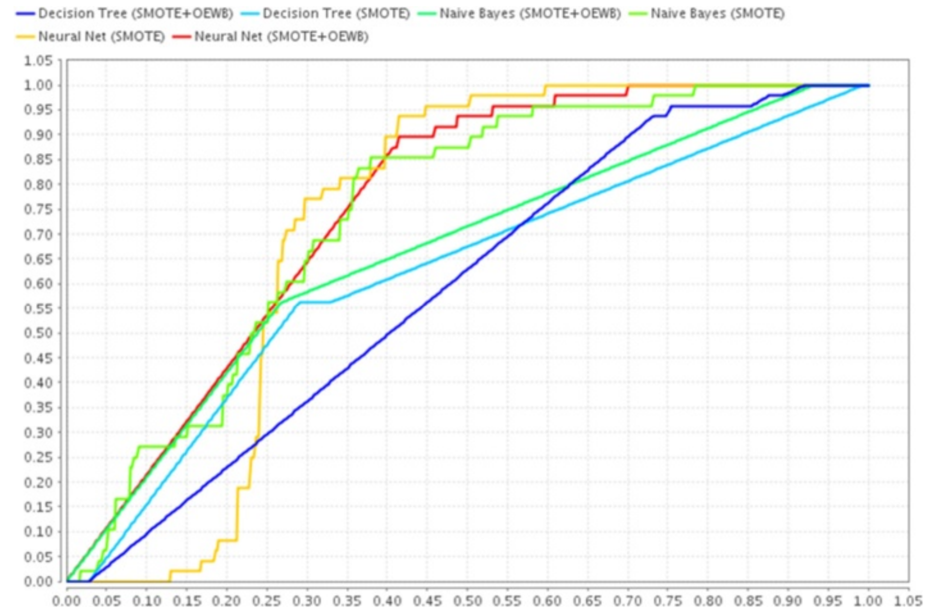


Figure 15 ROC curves for the class B (SMOTE Data).

For the Class D, as shown in Figure 17, the area under the curves is comparatively same as the Class A. There are major changes in Class D for over-sampled data compare to that of normal data. Model built by Neural Network classifier using over-sampled has area under the ROC curve of about 87% but when optimal equal width binning is used it drops down to roughly 66%.

Figure 18 describes the ROC curves for the class F. Model that is build by Naive Bayes classifier using over-sampled has area under the ROC curve of about 85% but when optimal equal width binning is use it drops down to roughly 45%. Same thing

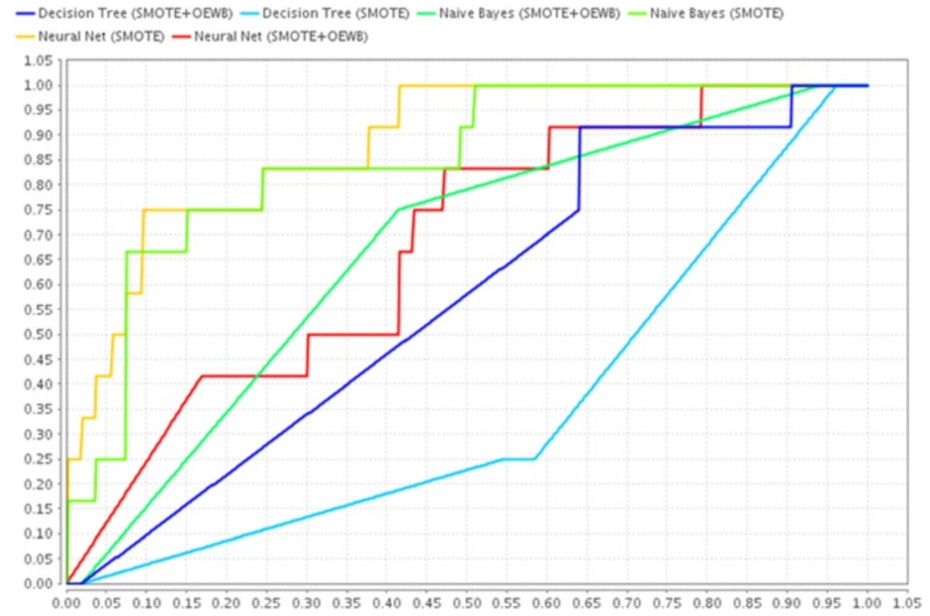


Figure 16 ROC curves for the class C (SMOTE Data).

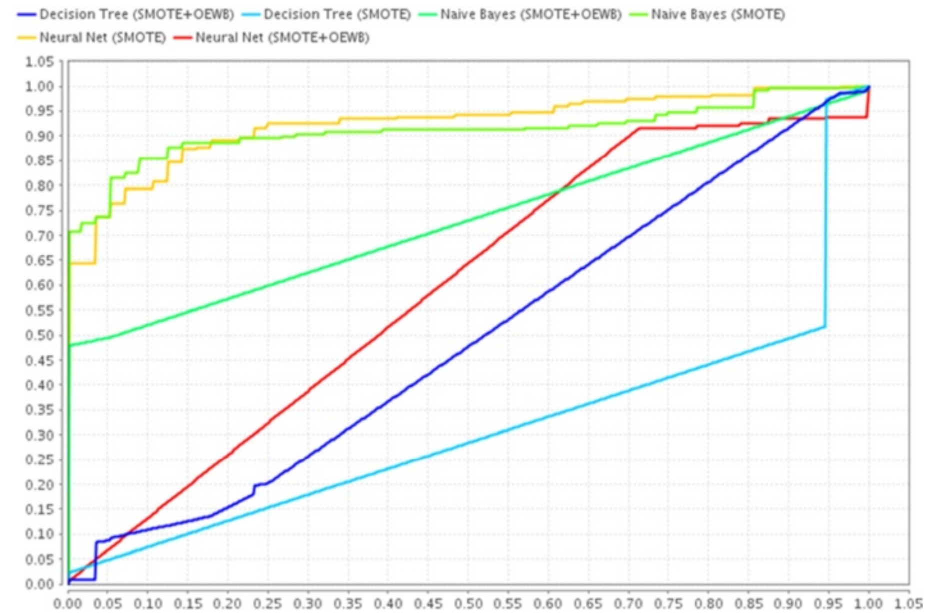


Figure 17 ROC curves for the class D (SMOTE Data).

happened in the case of the model built by Neural Net where area under the curve drops from 82% to 50% roughly.

Summary of the analysis

In the Table 16, all the models are listed along with the accuracy, average precision, average recall, average F-measure and average Area under the ROC Curve. When the optimal equal width binning is used on over-sampled data the Naive Bayes classifier and Neural Network classifier gives accuracy of about 75%. However when

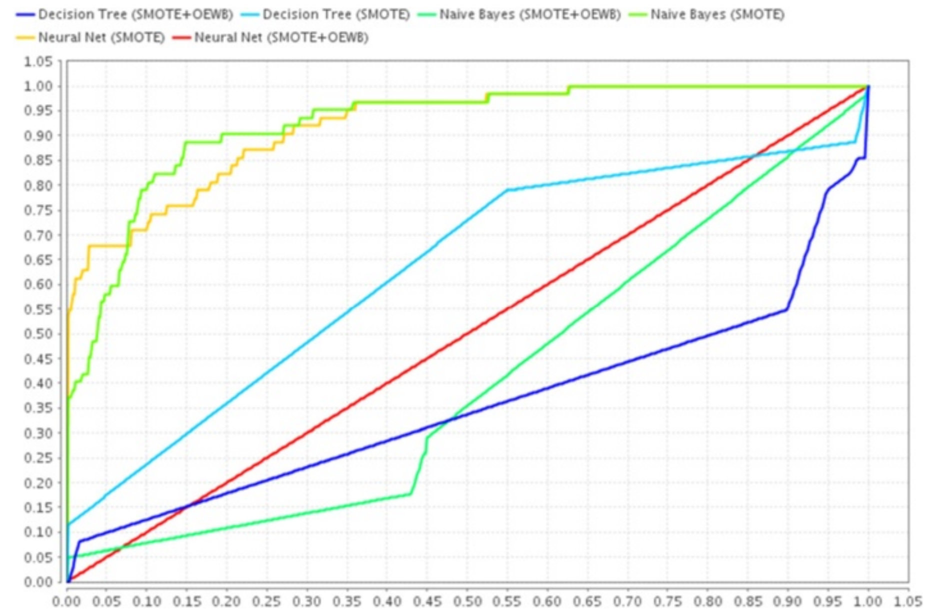


Figure 18 ROC curves for the class F (SMOTE Data).

Table 16 Analysis of the models

Model	Accuracy	Avg. Precision	Avg. Recall	Avg. F-Measure	Avg. AUC
Naive Bayes	61.11%	56.40%	53.30%	51.58%	75.6%
Naive Bayes (optimal binning)	68.33%	60.86%	60.35%	60.26%	68.9%
Naive Bayes (SMOTE)	66.67%	72.96%	66.67%	64.58%	81.4%
Naive Bayes (optimal binning + SMOTE)	75.28%	75.30%	75.58%	75.13%	71.8%
Decision tree	56.11%	56.90%	45.73%	43.44%	40.1%
Decision tree (optimal binning)	60.56%	50.56%	48.96%	49.54%	47.9%
Decision tree (SMOTE)	70.83%	72.48%	70.87%	70.61%	64.8%
Decision tree (optimal binning + SMOTE)	70.56%	70.91%	71.00%	70.03%	68.4%
Neural net	65.56%	70.21%	60.41%	62.65%	72.3%
Neural net (optimal binning)	68.89%	66.62%	69.89%	67.69%	73.1%
Neural net (SMOTE)	73.61%	73.54%	73.38%	73.59%	81.3%
Neural net (Optimal Binning + SMOTE)	75.28%	75.63%	75.42%	75.48%	71.6%

the classifiers were used on data where discretization and oversampling were not done then Naive Bayes classifier provides accuracy of about 61%, whereas, Neural Network classifier gives accuracy of about 66%. This means there is a greater accuracy gain when Naive Bayes classification is used.

In Figure 19 an overall graphical summary of the models are represented. On the figure multiple bar graphs are shown where on the *y*-axis each chunk highlights comparison of. The *x*-axis represents the percentage of the results obtained for accuracy,

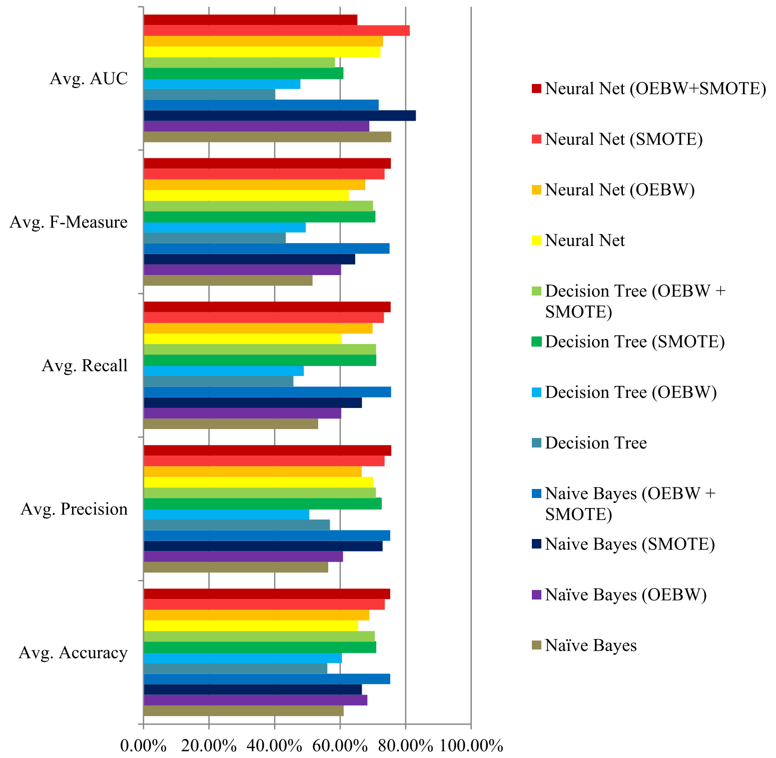


Figure 19 Analysis of the models.

average precision, average recall, average F-measure and average area under the ROC curves (AUC) for each model.

Pearson correlation coefficient for the validation measures

We wanted to determine the validity of the accuracy gained so we compared the accuracy of the models against the validation measures such as F-measure and Area under the ROC Curves. In order to compare, we used Pearson Correlation Coefficient (Chawla et al. 2002) to investigate the correlation between the classification model accuracy and the area under the ROC curve of each of the models Pearson Correlation Coefficient as well as correlation between classification models accuracy and the F-measures for those models. The coefficient varies between −1 to +1 where, anything above zero indicates positive correlation whereas anything below 0 indicates negative correlation. The correlation value of R is 0.6388 for Accuracy vs. AUC. This is a moderate positive correlation, which means there is a tendency that when the accuracy value will increase, AUC will also increase. The correlation value of R is 0.9793 for Accuracy vs. F-measure. This is a strong positive correlation, which means that if Accuracy increase F-measure will surely increase. The graphical representations of Accuracy vs. AUC and Accuracy vs. F-measure are shown in Figure 20 and Figure 21 respectively.

Conclusion

Our primary objective was to improve the models we build through preprocessing and then determining the model which gives the highest accuracy. As the number of instances in the dataset was small, oversampling was imminent. However, in order to distribute the instances we had to randomize the dataset twice. Two of the models which have the highest accuracy of about 75% are Neural Network and Naive Bayes classification with SMOTE oversampling and optimal equal width binning. Misclassification between two neighboring classes was high for the Classes D and F until the dataset was over-sampled and balanced. When Naive Bayes classifier was used on the original data

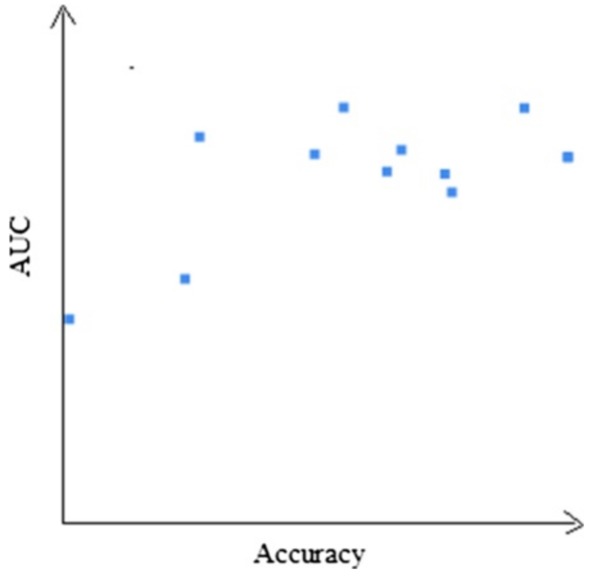


Figure 20 Pearson correlation between accuracy and AUC of the models.

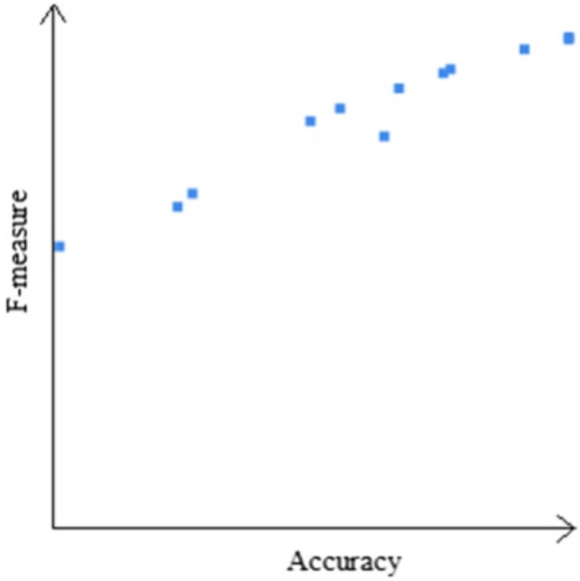


Figure 21 Pearson correlation between accuracy and F-measure of the models.

the accuracy was around 61%, which means there was almost 14% increase in accuracy when the discretization method was introduced on the class balanced data. We can observe that Naive Bayes and Neural Network models produced almost similar accuracy level. However, Naive Bayes classification is computationally faster than Neural Network Backpropagation algorithm and so it is the ideal choice. It can also be concluded that accuracy of any prediction system improves significantly when SMOTE oversampling and optimum equal width binning are used together to preprocess dataset which is small in size and contains continuous attributes. Perhaps the level of misclassification error can be minimized if more attributes can be taken into consideration, such as, students' grades in prerequisite courses. In future, we would also like to explore how the same optimization technique works for other data binning methods for example, binning by frequency, binning by size etc.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank the Department of Electrical and Computer Engineering, North South University for providing the students' course data for conducting the research.

Received: 12 October 2014 Accepted: 18 December 2014
Published online: 12 March 2015

References

Ayers, E, Nugent, R, & Dean, N. (2009). A comparison of student skill knowledge estimates. In *International Conference On Educational Data Mining, Cordoba, Spain* (pp. 1–10).

Bharadwaj, BK, & Pal, S. (2011). Mining educational data to analyze students' performance. *International Journal of Advance Computer Science and Applications, 2*(6), 63–69.

Bharadwaj, BK, & Pal, S. (2012). Data mining: a prediction for performance improvement using classification. *International Journal of Computer Science and Information Security, 9*(4), 136–140.

Chawla, NV, Bowyer, KW, Hall, LO, & Kegelmeyer, WP. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Chen, Y. Learning Classifiers from Imbalanced, Only Positive and Unlabeled Data Sets. Department of Computer Science, Iowa State University, USA (2009), Retrieved July 25, 2014, from https://www.cs.iastate.edu/~yetianc/cs573/files/CS573_ProjectReport_YetianChen.pdf.

Gedeon, TD, & Turner, HS. (1993). Explaining student grades predicted by a neural network. In *International conference on Neural Networks, Nagoya* (pp. 609–612).

Hämäläinen, W, & Vinni, M. (2006). Comparison of machine learning methods for intelligent tutoring systems. In *International Conference in Intelligent Tutoring Systems, Taiwan* (pp. 525–534).

Haykin, S. Neural Networks and Learning Machines, 3rd Edition, Pearson Education Inc., Upper Saddle River, New Jersey, USA, 2008.

Holmes, G, Donkin, A, & Witten, IH. (1994). Weka: a machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on (pp. 357–361)* (p. 357). IEEE.

Kayah, F. (2008). Discretizing Continuous Features for Naive Bayes and C4. 5 Classifiers. University of Maryland publications: College Park, MD, USA.

Nebot, A, Castro, F, Vellido, A, & Mugica, F. (2006). Identification of fuzzy models to predict students perfornance in an e-learning environment. In *International Conference on Web-based Education, Puerto Vallarta* (pp. 74–79).

Oladokun, VO, Adebanjo, AT, & Charles-owaba, OE. (2008). Predicting student's academic performance using artificial neural network: a case study of an engineering course. *Pacific Journal of Science and Technology, 9*(1), 72–79.

Pal, AK, & Pal, S. (2013). Analysis and Mining of Educational Data for Predicting the Performance of Students. International Journal of Electronics Communication and Computer Engineering, Vol. 4, Issue 5 (pp. 1377–1381).

Prati, RC, Batista, GE, & Monard, MC. (2004). Class imbalances versus class overlapping: an analysis of a learning system behavior. In *MICAI 2004: Advances in Artificial Intelligence* (pp. 312–321). Heidelberg: Springer Berlin.

Quinlan, JR. (1993). "C4. 5: programs for machine learning". Morgan kaufmann. Morgan Kaufmann: San Francisco, CA, USA

Rahman, MM, and Davis, DN. Addressing the Class Imbalance Problem in Medical Datasets, International Journal of Machine Learning and Computing, Vol. 3, No. 2, April 2013, (pp. 224-228).

Romero, C, & Ventura, S. (2010). Educational data mining: a review of the state of the art. Systems, man, and cybernetics, part C: applications and reviews. *IEEE Transactions on, 40*(6), 601–618. Chicago.

Superby, JF, Vandamme, JP, & Meskens, N. (2006). Determination of factors influencing the achievement of the first-year university students using data mining methods. In *International Conference on Intelligent Tutoring Systems, Educational Data Mining Workshop, Taiwan* (pp. 1–8).

Tan, P, Kumar, V, & Steinbach, M. (2006). *"Introduction to Data Mining"*. New Delhi: Dorling Kindersley(India) Pvt. Ltd.

Want, T, & Mitrovic, A. (2002). Using neural networks to predict student's performance. In *International Conference on Computers in Education, Washington, DC* (pp. 1–5).

Yadav, SK, Bharadwaj, B, & Pal, S. (2012). *Data Mining Applications: A Comparative Study for Predicting student's Performance. arXiv preprint arXiv:1202.4815.*

Zhu, F, Ip, HH, Fok, AW, & Cao, J. (2007). Peres: a personalized recommendation education system based on multi-agents & scorm. In *Advances in Web Based Learning–ICWL* (pp. 31–42). Heidelberg: Springer Berlin.

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Immediate publication on acceptance
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com