Non-Written Enrolment System using Classification Methods

M Lestari*, A Darmawan, N W P Septiani and A Trihapsari
Universitas Indraprasta PGRI, Jakarta, Indonesia
*mei_lestari6@unindra.ac.id

Abstract. Institutions of higher education nowadays has different technique to organize their enrolment system (e.g. Written or non-written test). Classification methods are used to predict student’s GPA. These GPA’s prediction is used to determine whether students are accepted or not. Classification process include 2 phases, learning phase and classification phase. In learning phase training data are analyse by classification algorithm. Tested data is use to estimate the accuracy of classification rules. Aim of this study is to compare datamining classification methods i.e. C4.5 algorithm and naive Bayes algorithm implemented on non-written enrolment system. This study shows that naive Bayes has better accuracy than C4.5 algorithm.

1. Introduction

Institution of higher education system has different techniques to organize their enrolment system (e.g. written or non-written test). This paper use data mining classification methods to estimate student’s GPA to determine whether the students are accepted or not. Attributes that used in this paper are score of mathematic’ subject, physic’ subject, English subject, major interest, student’ name and school of origin. By using student’ report at class 10, 11 and 12 we can predict student’s GPA and determine whether student’s accepted or not.

There are various background of students that register to Universitas Indraprasta Jakarta; e.g. general high school, vocational high school, pharmaceutical high school, hence the announcement of new students are not accurate and slow. Therefor we made a system to determine accepted students using their GPA’s prediction (non written test enrolment system). By sending invitation to high school that considered have a good reputation (or rank) to send their students that interested to enroll Universitas Indraprasta (PGRI), students that interested and met the requirement have to submit their score report and their curriculum vitae including their major interest.

2. Methodology

Data mining is a multidisciplinary field that combines statistics, machine learning, artificial intelligence and database technology. Data mining means predicting the future by means of modelling. Classification is a task of mapping an input attribute set x into class label y [1]–[3]. Classification is a systematic approach to build classification models from input data. Classification might solve mathematical techniques such as decision tree, linier programming, neural network, and statistics. There are algorithms to preform classifications including decision tree, rule-based induction, neural network, Bayesian network and genetic algorithm. Classification process include 2 phases, learning phase and classification phase. In learning phase training data are analyze by classification algorithm. Tested data is use to estimate the accuracy of classification rules. This algorithm analyze input to produce the
prediction. Training data is applied on classification algorithm which generate classifier models with “if-then” rules to predict unknown record.

![Figure 1. Classification process](image)

2.1. C4.5 Algorithm
C4.5 algorithm is an algorithm used to generate a decision tree developed by Ross Quinlan [4]. This algorithm is an extension of ID3 algorithm. Decision tree that built from C4.5 algorithm can be use for classification. By knowing the classes of training set, we use algorithm to discover the way the attributes behave and estimates the new instances. The way to do this process by using decision tree. A tree is either a leaf node that connected to two or more nodes (or sub trees).

C4.5 algorithm based on ID3 (Iterative Dichotomiser 3) to find the simple (or small) decision trees based on some premises as follows: if all cases belong to the same class and the tree is a leaf, then it will be labelled to this class, and for each attribute calculate the potential information provided by a test on the attribute. The way to calculate the gain that is a measure of the disorder of the data is as follows:

\[
\text{Entropy}(\hat{y}) = -\sum_{j=1}^{n} \frac{|y_j|}{|\hat{y}|} \log \frac{|y_j|}{|\hat{y}|}
\]

\[
\text{Entropy}(j|\hat{y}) = \frac{|y_j|}{|\hat{y}|} \log \frac{|y_j|}{|\hat{y}|}
\]

\[
\text{Gain}(\hat{y}, j) = \text{Entropy}(\hat{y}) - \text{Entropy}(j|\hat{y})
\]

Aims of this calculation is to maximize the gain, dividing by overall entropy, splitting argument \( \hat{y} \) by value \( j \). Pruning is done after the complete creation of the tree. to reduce classification errors that caused by specialization in training set. There are two types of pruning, pre-pruning and post pruning. Pre-pruning is pruning process in which check whether tree is overfitting while building the decision tree. Post pruning is pruning process in which built the tree first then reduce non-significant branches and levels of decision tree is done.

2.2. Naïve Bayes Algorithm
Naïve bayes classifier is a simple probabilistic classifier based on bayes theorem with strong naive assumption [5]–[7]. It refers to the statistician philosopher Thomas Bayes and the theorem named after him, bayes’ theorem, which is base for the naïve bayes algorithm. Naïve bayes equation can be written as \( \text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \). Naïve bayes works well for certain nearly functional feature dependencies, thus reaching its best performance in two opposite cases: completely independent features (as expected) and functionally dependent features (which is surprising). And the accuracy of naïve Bayes is not directly correlated with the degree of feature dependencies measured as the class conditional mutual information between the features. Instead, a better predictor of naïve Bayes accuracy is the amount of information about the class that is lost because of the independence assumption [8].

2.3. Pruning Methods
Different pruning methods are discussed and it’s effectiveness are evaluated in terms of complexity and classification accuracy by using C4.5 algorithm on credit card database with and without pruning. Instead of classifying the transactions either fraud or non-fraud the transactions are classified into four risk level that is an innovative concept [9]. Pre-pruning and Post-pruning are two standard that use to handling noise in decision tree techniques. Difference of pre-pruning and post-pruning is that pre-
pruning deals with noise while building the tree, while post-pruning addresses deals with noise after an over fitting theory has been learned [10].

3. Result and Discussion

3.1. System Design

Student enrolment system use to predict high school student that accepted in Universitas Indraprasta PGRI by non written enrolment test. This student classification system is built using C4.5 Algorithm and Naïve Bayes Algorithm) based on training data. The output is GPA’ prediction and decision whether student accepted or not. This system shows raw training and testing data that will be processed, and this system will do discretising of scores and GPA’.

Use case diagram of student enrolment system are as follows:

![Use Case Diagram](image)

**Figure 2.** Use Case Diagram

Three class on student enrolment system i.e. boundary class, control class and entity class. Boundary class consist of TDataForm, TDataSetForm, TBayesForm and TC45Form. Control class consist of CClassifier, CNaive Bayes, CC4.5 and CNode, while entity class consist of T_STUDENTS, T_DATASETS, T_PROJECTS, T_FORECASTING, T_PROBABILITY, T_RULES, T_TREES.

Class analysis classification shows in Table 1.

| No. | Class Name    | Class Identification |
|-----|---------------|----------------------|
| 1.  | TMainForm     | Boundary             |
| 2.  | TDataForm     | Boundary             |
| 3.  | TDataSetForm  | Boundary             |
| 4.  | TBayesForm    | Boundary             |
| 5.  | TC4.5Form     | Boundary             |
| 6.  | CClassifier   | Control              |
| 7.  | CNaiveBayes   | Control              |
| 8.  | CC4.5         | Control              |
| 9.  | CNode         | Control              |
| 10. | T_STUDENTS    | Entity               |
| 11. | T_DATASETS    | Entity               |
| 12. | T_PROJECTS    | Entity               |
| 13. | T_FORECASTING | Entity               |
| 14. | T_PROBABILITY | Entity               |
There are four submenus in this prototype i.e. raw data, Data Set, Naïve Bayes and C4.5. Raw data menu is use to determine boundaries/limit of IPKTinggi (High GPA) to be used in discretizing process. Data set menu is use to determine size of training set and testing set. Naïve bayes menu is use to determine parameter and run naïve Bayes classifier to predict student’ GPA. C4.5 Naïve bayes menu is use to built decision tree and run c4.5 classifier to predict student’ GPA.

3.2. Naïve Bayes and C45 Performance Analysis
Testing set size that has been chosen are 10%, 15%, 20% and 25%. For each size of testing set has been randomly tested 10 times with GPA’ boundary 3.00. In this testing phase, given training data to built probability table. Performance of naïve Bayes algorithm and c4.5 algorithm can be presented by using confusion matrix or also known as an error matrix [11]. Confusion matrix is use to visualize performance of supervised learning algorithm.

Following table are confusion matrix of Naïve Bayes algorithm using 14000 data of year 2006 – 2010.

**First Testing**, using 10% testing data

| Actual | Accepted | Not Accepted |
|--------|----------|--------------|
| Class  | 6929     | 1112         |
|        | 2745     | 1791         |

**Precision**: \[ P = \frac{6929}{6929+2745} \times 100\% = 71.62\% \]

**Recall**: \[ R = \frac{6929}{6929+1112} \times 100\% = 86.17\% \]

**Accuracy**: \[ A = \frac{6929}{13256} \times 100\% = 65.78\% \]

**Second Testing**, using 15% of testing data

| Actual | Accepted | Not Accepted |
|--------|----------|--------------|
| Class  | 6535     | 1056         |
|        | 2507     | 1681         |

**Precision**: \[ P = \frac{6535}{6535+2507} \times 100\% = 72.27\% \]

**Recall**: \[ R = \frac{6535}{6535+1056} \times 100\% = 86.08\% \]

**Accuracy**: \[ A = \frac{6535}{11779} \times 100\% = 69.75\% \]

**Third Testing**, using 20% of testing data

| Actual | Accepted | Not Accepted |
|--------|----------|--------------|
| Class  | 6201     | 967          |
|        | 2459     | 1553         |

**Precision**: \[ P = \frac{6201}{6201+2459} \times 100\% = 71.60\% \]

**Recall**: \[ R = \frac{6201}{6201+967} \times 100\% = 86.50\% \]

**Accuracy**: \[ A = \frac{6201}{11180} \times 100\% = 69.35\% \]

**Fourth Testing**, using 25% of testing data

| Actual | Accepted | Not Accepted |
|--------|----------|--------------|
| Class  | 5776     | 928          |
|        | 2283     | 1495         |

**Precision**: \[ P = \frac{5776}{5776+2283} \times 100\% = 71.67\% \]

**Recall**: \[ R = \frac{5776}{5776+928} \times 100\% = 86.15\% \]

**Accuracy**: \[ A = \frac{5776+1495}{10482} \times 100\% = 69.36\% \]
Following table are confusion matrix of C4.5 algorithm using 14000 data of year 2006 – 2010.

**First Testing**, using 10% testing data

| Predicted Class | Accepted (T) | Not Accepted (R) |
|-----------------|--------------|------------------|
| Actual Class    | Accepted(T)  | 2857             | 342              |
|                 | Not Accepted (R) | 918             | 743              |

**Table 6. Confusion Matrix First Testing for C4.5 Algorithm**

**Precision**: 

\[ P = \frac{2857}{(2857+918)} \times 100\% = 75.68\% \]

**Recall**: 

\[ R = \frac{2857}{(2857+342)} \times 100\% = 89.30\% \]

**Accuracy**: 

\[ A = \frac{2857}{(2857+342)} \times 100\% = 74.07\% \]

**Second Testing**, using 15% of testing data

| Predicted Class | Accepted (T) | Not Accepted (R) |
|-----------------|--------------|------------------|
| Actual Class    | Accepted(T)  | 2524             | 320              |
|                 | Not Accepted (R) | 829             | 662              |

| Predicted Class | Accepted (T) | Not Accepted (R) |
|-----------------|--------------|------------------|
| Actual Class    | Accepted(T)  | 1899             | 258              |
|                 | Not Accepted (R) | 586             | 632              |

**Table 7. Confusion Matrix Second Testing for C4.5 Algorithm**

**Precision**: 

\[ P = \frac{2524}{(2524+829)} \times 100\% = 75.27\% \]

**Recall**: 

\[ R = \frac{2524}{(2524+320)} \times 100\% = 88.74\% \]

**Accuracy**: 

\[ A = \frac{2524}{(2524+320)} \times 100\% = 73.49\% \]

**Fourth Testing**, using 25% of testing data

| Predicted Class | Accepted (T) | Not Accepted (R) |
|-----------------|--------------|------------------|
| Actual Class    | Accepted(T)  | 1899             | 258              |
|                 | Not Accepted (R) | 586             | 632              |

| Predicted Class | Accepted (T) | Not Accepted (R) |
|-----------------|--------------|------------------|
| Actual Class    | Accepted(T)  | 1899             | 258              |
|                 | Not Accepted (R) | 586             | 632              |

**Precision**: 

\[ P = \frac{1899}{(1899+586)} \times 100\% = 76.41\% \]

**Recall**: 

\[ R = \frac{1899}{(1899+258)} \times 100\% = 88.03\% \]

**Accuracy**: 

\[ A = \frac{1899}{(1899+258)} \times 100\% = 75.39\% \]

**Third Testing**, using 20% of testing data

| Predicted Class | Accepted (T) | Not Accepted (R) |
|-----------------|--------------|------------------|
| Actual Class    | Accepted(T)  | 2857             | 342              |
|                 | Not Accepted (R) | 918             | 743              |

**Table 8. Confusion Matrix Third Testing for C4.5 Algorithm**

**Precision**: 

\[ P = \frac{2524}{(2524+829)} \times 100\% = 75.27\% \]

**Recall**: 

\[ R = \frac{2524}{(2524+320)} \times 100\% = 88.74\% \]

**Accuracy**: 

\[ A = \frac{2524}{(2524+320)} \times 100\% = 73.49\% \]

**Fourth Testing**, using 25% of testing data

| Predicted Class | Accepted (T) | Not Accepted (R) |
|-----------------|--------------|------------------|
| Actual Class    | Accepted(T)  | 1899             | 258              |
|                 | Not Accepted (R) | 586             | 632              |

**Precision**: 

\[ P = \frac{1899}{(1899+586)} \times 100\% = 76.41\% \]

**Recall**: 

\[ R = \frac{1899}{(1899+258)} \times 100\% = 88.03\% \]

**Accuracy**: 

\[ A = \frac{1899}{(1899+258)} \times 100\% = 75.39\% \]

4. Conclusion

From the above computation can be summarize the comparison of C4.5 and naïve bayes performance using four testing data.

| Naïve Bayes | C4.5 | Testing data size |
|-------------|------|-------------------|
| 65.78 %     | 74.07 % | 10%                |
| 69.75 %     | 73.49 % | 15%                |
| 69.35 %     | 73.49 % | 20%                |
| 69.36 %     | 75.39 % | 25%                |
Table 10 present performance of C4.5 algorithm and naïve bayes algorithm that use in non written
enrollment system. It shows that C4.5 Algorithm has better performance than naïve bayes algorithm.
This paper proposed student enrollment system based on GPA predictions using both C4.5 and Naïve
Bayes Algorithm.

References
[1] Tan P-N Steinbach M and Kumar V, 2006 Classification : Basic Concepts , Decision Trees , and
Introd. to Data Min. 67, 17 p. 145–205.
[2] Larose D T, 2006 Data Mining Methods and Models.
[3] Giudici P, 2005 DATA MINING MODEL COMPARISON Data Min. Knowl. Discov. Handb. p. 697–714.
[4] Quinlan J R, 1992 C4.5: Programs for Machine Learning 1, 3.
[5] Ren J Lee S D Chen X Kao B Cheng R and Cheung D, 2009 Naive bayes classification of
uncertain data in Proceedings - IEEE International Conference on Data Mining, ICDM p. 944–
949.
[6] Vikramkumar B V Trilochan Vijaykumar B Vikramkumar and Trilochan, 2014 Bayes and Naive
Bayes Classifier arXiv.
[7] Murphy K P, 2006 Naive Bayes classifiers Generative classifiers Bernoulli 4701, October p. 1–
8.
[8] Rish I, 2001 An empirical study of the naive Bayes classifier IJCAI 2001 Work. Empir. methods
Artif. Intell. 22230 p. 41–46.
[9] Patil D D Wadhai V M and Gokhale J A, 2010 Evaluation of Decision Tree Pruning Algorithms
for Complexity and Classification Accuracy Int. J. Comput. Appl. 11, 2 p. 975–8887.
[10] Fürnkranz J, 1997 Pruning methods for rule learning algorithms Mach. Learn. 27, 2 p. 139–171.
[11] Stehman S V., 1997 Selecting and interpreting measures of thematic classification accuracy
Remote Sens. Environ. 62, 1 p. 77–89.