Expert Systems with Applications 38 (2011) 11421–11436

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using and comparing different decision tree classification techniques for mining ICDDR,B Hospital Surveillance data

Rashedur M. Rahman *, Fazle Rabbi Md. Hasan

Department of Electrical Engineering and Computer Science, North South University, Bashundhara, Dhaka, Bangladesh

ARTICLE INFO

Keywords:
Medical surveillence
Decision tree
Artificial intelligence
Classification
Data mining

ABSTRACT

In this research we have used decision tree induction algorithm on Hospital Surveillance data to classify admitted patients according to their critical condition. Three class labels, low, medium and high, are used to distinguish the criticality of the admitted patients. Several decision tree models are developed, evaluated, and compared with different performance metrics. Finally an efficient classifier is developed to classify records and make decision/predictions on some input parameters. The models developed in this research could be helpful during epidemic when huge number of patients arrive daily. Due to rush of duty doctors and scarcity of required number of physicians, it is hard to diagnose every patient. Any computer application could be helpful to diagnose and measure the criticality of the newly arrived patient with the help of the historical data kept in the surveillance database. The application would ask few questions on physical condition and on history of disease of the patient and accordingly determines the critical condition of the patient as low, medium or high.

1. Introduction

A diarrhoeal disease surveillance system was established at ICDDR,B, Dhaka, Bangladesh (ICDDR,B, 2008) in 1979 and extended to the Matlab hospital at Comilla, Bangladesh in 2003. The surveillance system collects information on clinical, epidemiological and demographic characteristics of patients. A systematic 2% sub-sample of patients attending Clinical Research and Service Centre (CRSC) and all patients from the Health and Demographic Surveillance System (HDSS) area attending the Matlab hospital are enrolled into the surveillance program. Trained personnel interview the patients and/or their attendants to obtain information on socioeconomic and demographic characteristics, housing and environmental conditions, feeding practices, particularly among infants and young children, and on the use of drugs and fluid therapy at home. Clinical characteristics, anthropometric measurements, treatments received at the facility, and outcomes of patients are also recorded. Extensive microbiological assessments of faecal samples (microscopy, culture, and ELISA) of patients are performed to identify diarrhoeal pathogens and to determine antimicrobial susceptibility of bacterial pathogens.

The program stores important information that helps hospital clinicians to provide care to patients. It also enables the centre to detect the emergence of new pathogens and responds to early identification of outbreaks and their locations to suggest the Government of Bangladesh to take preventive and other control measures and to monitor the changes in the characteristics of patients and antimicrobial susceptibility of bacterial pathogens. Collected information is representative of the population. Hence, it provides an important data repository for conducting epidemiological studies, validation of results of clinical studies, helps develop new research ideas and study design, and introduce improved patient-care strategies and preventive programs.

1.1. Motivation

When patients arrive at hospital, an initial diagnosis is carried out by the duty physician to find out the criticality of the patients condition. According to the criticality of patient's physical condition the duty doctor takes necessary action. Sometimes when there is epidemic like in the year 2008 during flood, when 1000 patient on an average admitted in the hospital, it is hard to diagnose every patient satisfactorily due to rush of duty doctors and scarcity of required number of physicians. Any computer application could be helpful here to diagnose and measure the criticality of the newly arrived patient with the help of the historical data kept in the surveillance database. The application would ask few questions on physical condition and on history of disease of the patient and accordingly determines the critical condition of the patient as low, medium or high.

* Corresponding author.
E-mail addresses: rashedur@northsouth.edu (R.M. Rahman), fazle@icddrb.org (F.R. Md. Hasan).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.015

1.2. Objective

Hospital Surveillance database contain records that store patient's initial physical condition, his/her history of disease, as well the final outcome of treatment and duration of his/her stay in the hospital.

The outcome field has the following values stored: 1 = Cured, 2 = Illness cont, 3 = Died, 4 = Absconded, 5 = Others, 9 = Unknown. In this research, we consider only the records of the patients with outcome = 1 as records with other outcome (outcome <> 1) are incomplete in most of the cases. However, we do not find any attribute field that stores the initial diagnosis for "Criticality" with value: low, mid, high. We use an intuition here that the duration of stay in the hospital would indicate to the criticality of the patient. We create a derived attribute "Criticality" by banning the duration of stay field as follows:

0 to ⩽48 h: Low,
48> to ⩽96 h: Mid,
>96 High.

1.3. Contribution

In this research we have used decision tree induction algorithm on Hospital Surveillance data to classify admitted patients according to their critical condition. Several decision tree models are developed, evaluated, and compared with different performance metrics. Finally an efficient classifier is developed to classify records and make decision/predictions on some input parameters.

2. Related work

Several research works have been conducted to identify new, unexpected and interesting patterns from hospital infection control and public health surveillance data. In Brossette (1998), Ma, Tsui, Hogan, Wagner, and Ma (2003) and Moser, Jones, and Brossette (1999) data mining techniques are used for monitoring emerging infections and antimicrobial resistance.

The monitoring and detection of nosocomial infections is a important problem arising in hospitals. A hospital-acquired or nosocomial infection is a disease that develops after admission into the hospital and it is the consequence of a treatment, not necessarily a surgical one, performed by the medical staff. Nosocomial infections are dangerous because they are caused by bacteria which have dangerous (critical) resistance to antibiotics. This problem is very serious all over the world. In order to support them in this complex task, a system have been developed, called MERCURIO (Lamma et al., 2006). The objectives of this system are the validation of microbiological data and the creation of a real time epidemiological information system. The system is useful for laboratory physicians, because it supports them in the execution of the microbiological analyses; for clinicians, because it supports them in the definition of the prophylaxis, of the most suitable antibiotic therapy and in monitoring patients' infections; and for epidemiologists, because it allows them to identify outbreaks and to study infection dynamics.

In order to achieve these objectives data mining techniques have been adopted. Data mining techniques have been used for improving the system knowledge base. In order to verify the reliability of the tasks performed by MERCURIO and the usefulness of the knowledge discovery approach, a test was performed based on a dataset of real infection events. In the validation task MERCURIO achieved an accuracy of 98.5%, a sensitivity of 98.5% and a specificity of 99%. In the therapy suggestion task, MERCURIO achieved very high accuracy and specificity as well.

Patterns embedded in large volumes of clinical data may provide important insights into the characteristics of patients or care delivery processes, but may be difficult to identify by traditional means. Research have been carried out to develop method using data mining that can recognize patterns in these large data sets. In Brossette and Hymel (2008), examples have been presented of this capability in which, data mining has been successfully applied to hospital infection control. The Data Mining Surveillance System (DMSS) uses data from the clinical laboratory and hospital information systems to create association rules linking patients, sample types, locations, organisms, and antibiotic susceptibilities. Changes in association strength over time, signal epidemiologic patterns potentially appropriate for follow-up, and additional heuristic methods identify the most informative of these patterns for alerting.

Data mining is the process of discovering interesting knowledge, such as patterns, associations, changes, anomalies and significant structures, from large amounts of data stored in databases, data warehouses, or other information repositories. In Healthcare, association rules are considered to be quite useful as they offer the possibility to conduct intelligent diagnosis and extract invaluable information and build important knowledge bases quickly and automatically. The problem of identifying new, unexpected and interesting patterns in medical databases in general, and diabetic data repositories in specific, is considered in a research (Stilou, Bamidis, Maglaveras, & Pappas, 2001). In this paper the apriori algorithm to a database containing records of diabetic patients and attempted to extract association rules from the stored real parameters. The results indicate that the methodology followed may be of good value to the diagnostic procedure, especially when large data volumes are involved. The followed process and the implemented system offer an efficient and effective tool in the management of diabetes.

Another research paper (Houston et al., 1999) discusses several data mining algorithms and techniques that have been developed at the University of Arizona Artificial Intelligence Lab. In this paper those algorithms and techniques are implemented into several prototypes, one of which focuses on medical information developed in cooperation with the National Cancer Institute (NCI) and the University of Illinois at Urbana-Champaign. An architecture have been proposed for medical knowledge information systems that will permit data mining across several medical information sources and discuss a suite of data mining tools that have been developed to assist NCI in improving public access to and use of their existing vast cancer information collections.

Research work has also been done to demonstrate and test usefulness and performance of data mining tools and techniques being applied to academic research conducted on Asthma patients. (Bereznicki et al., 2008; Tseng, Chao-Hui, Lee, & Chia-Yu Chen, 2008).

The prediction of survival of Coronary Heart Disease (CHD) has been a challenging research problem for medical society. The objective of research presented in Xing Yanwei, Jie Wang, and Zhihong Zhao (2007) was to develop data mining algorithms for predicting survival of CHD patients based on 1000 cases. A clinical observation and a 6-month follow up were carried out to include 1000 CHD cases. The survival information of each case is obtained via follow up. Based on the data, three popular data mining algorithms were employed to develop the prediction models using the 502 cases. Also 10-fold cross-validation methods were used to measure the unbiased estimate of the three prediction models for performance comparison purposes. The results indicated that the SVM is the best predictor with 92.1% accuracy on the holdout sample artificial neural networks came out to be the second with 91.0% accuracy and the decision trees models came out to be the worst of the three with 89.6% accuracy. The comparative study of multiple prediction models for survival of CHD patients along with a 10-fold cross-validation provided an insight into the relative prediction ability of different data.

Research have been done to develop an artificial intelligence-based data mining engine (CureHunter) (CureHunter- precision medical data mining, 2008) that can autonomously search all the known biomedical research journals, collate the published drug efficacy evidence for specific diseases and present it in a format that is available in real-time (10–20 s) for patients and physicians to review. With integration into existing physician record management systems, physicians can use (free of charge) the drug research interface and obtain up-to-date summarized clinical effectiveness information on a wide range of drugs and diseases while the patient is sitting in the room!

For patients, it tries to answer the question: what does the scientific community think the best treatment options for disease Y are? Patients simply need to enter the disease they wish to know more about in the search box on the front page. They utilize the Mesh-based ontological terms to help narrow their search down to the specific disease they are searching for and CureHunter returns to them: (a) Key Drugs and Agents for the treatment of that disease, (b) Other Related Diseases, and (c) Key Therapies for that disease. When one thinks how difficult it would be for physicians to realistically and comprehensively review the literature on all drugs they prescribe, this kind of engine has significant potential.

The data mining engine has been in development for a few years now, and after two prototyping iterations was released to the public as a beta in July 2007. The CureHunter Corporation is still at early startup stage and is still seeking venture funding and/or a partner.

In a data mining research paper (Buntinx, Truyen, Embrechts, Moreel, & Peeters, 1992) on medical records, data have been collected on 320 patients complaining to their general practitioner of a new episode of chest pain, discomfort or oppression. Relationships were examined between initial signs and symptoms and a follow-up diagnosis after a period of 2 weeks to 2 months. The data were analysed with CART, a statistical decision theory software package. In the first run, the number of misclassifications by CART was 56%. After regrouping of the data and diagnostic categories, there were 37% misclassifications. The most discriminating variable turned out to be pain on palpation. When comparing each of five diagnostic groups to all others, it was found that a positive predictive value of 27% for gastrointestinal diseases, 72% for cardiovascular disorders, 69% for respiratory diseases, 58% for psychopathology and 73% for chest wall pathology. The CART methodology needs further investigation and testing before any clinical application will be possible in general practice.

In Mair, Smidt, Lechleitner, Dienstl, and Puschendorf (1995) a study was carried out to find an accurate algorithm for the diagnosis of acute myocardial infarction in nontraumatic chest pain patients on presentation to the emergency department. In a prospective clinical study, the diagnostic performances of clinical symptoms were compared, presenting ECG, creatinine kinase, creatine kinase MB activity and mass concentration, myoglobin, and cardiac troponin T test results of hospital admission blood samples. By classification and regression trees, a decision tree for the diagnosis of acute myocardial infarction was developed. The research was conducted at Emergency room of a Department of Internal Medicine (University Hospital) on 114 nontraumatic chest pain patients: 26 Q-wave and 19 non-Q-wave myocardial infarctions, 49 patients with unstable angina pectoris, and 20 patients with chest pain caused by other diseases. In this study an algorithm was found that could accurately separate the myocardial infarction patients from the others on admission to the emergency department. Therefore, this classifier could be a valuable diagnostic aid for rapid confirmation of a suspected myocardial infarction.

In another research paper (Hadzikadic et al., 1995) two classification models were presented, one based on concept formation and the other using standard logistic regression. The models were first explained in some detail and then evaluated on the same population of trauma patients. The goal of both systems is to predict the outcome of those patients. The results are summarized and explained in terms of differing algorithms of the two models.

3. Decision tree classifier

We could solve a classification problem by asking a series of carefully crafted questions about the attributes of the test record. Each time we receive an answer, a follow-up question is asked until we reach a conclusion about the class label of the record. The series of questions and their possible answers can be organized in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edges.

Classifying a test record is straightforward once a decision tree has been constructed. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. This will lead us either to another internal node, for which a new test condition is applied, or to a leaf node. The class label associated with the leaf node is then assigned to the record.

3.1. Decision tree induction algorithm

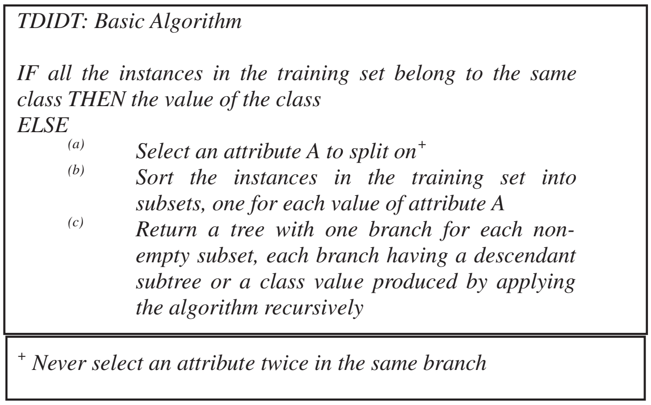
TDIDT (Top-Down Induction of Decision Tree) algorithm has formed the basis for many classification systems, two of the best known being ID3 and c4.5.

Decision trees are generated by repeatedly splitting on the values of attributes. This process is known as recursive partitioning. In the standard formulation of the TDIDT algorithm there is a training set of instances. Each instance belongs to an object class, which is described by the values of a set of attributes.

The basic algorithm can be given in just a few lines as shown in Fig. 1.

At each non-leaf node an attribute is chosen for splitting. This could be any attribute, except that the same attribute must not be chosen twice in the same branch. However this harmless restriction has a very valuable effect. Each split on the value of an attribute extends the length of the corresponding branch by one term, but the maximum possible length for a branch is M terms where there are M attributes. Hence the algorithm is guaranteed to terminate.

There is one important condition which must hold before the TDIDT algorithm can be applied. This is the adequacy condition: no two instances with the same values of all attributes may belong



```
TDIDT: Basic Algorithm

IF all the instances in the training set belong to the same
class THEN the value of the class
ELSE
    (a)   Select an attribute A to split on+
    (b)   Sort the instances in the training set into
          subsets, one for each value of attribute A
    (c)   Return a tree with one branch for each non-
          empty subset, each branch having a descendant
          subtree or a class value produced by applying
          the algorithm recursively

+ Never select an attribute twice in the same branch
```

Fig. 1. The TDIDT algorithm.

to different classes. This is simply a way of ensuring that training set is consistent.

A major problem with the TDIDT algorithm is that it is underspecified: no method is given to select the attribute A on which the split will be done. There are some techniques for selecting attributes. However, some of them may be more useful than others. Making a good choice of attributes to split on at each stage is crucial to the success of the TDIDT approach.

3.2. Choosing the attributes

To know the order in which attributes much be chosen to split the data, we need some measure that would allow us to compare the attributes on some scale and choose one above the other. One of the measures for selecting the "best" question or attribute is based on the level of *Impurity* in the resulting classes of data. *Impurity* could be defined as the amount of uncertainty present in the data and that the attribute which reduces the impurity most should be chosen.

Given probability p, some of the impurity measures are:

- Gini Index: $2p\ (1-p)$
- Entropy: $-[p\ \log p + (1-p)\ \log\ (1-p)]$
- Misclassification Rate: $1 - \max(p, 1-p)$

In general, when the dataset could be divided into two classes, then p is the proportion of instances in the database that has one value for the target attribute and $1-p$ is the proportion of instances in the database that has the second value for the same target attribute.

3.2.1. Generalization of the impurity measures

In the previous section, we assumed the instances in the database could be classified two-ways. When the number of classes becomes three or more i.e. C1, C2 and C3, where

- $P(C1) = p$
- $P(C2) = q$
- $P(C3) = 1 - p - q$

then the impurity measures could be generalized as follows.

- Gini Index

$$\sum_{i,j,i\neq j} p_i p_j = 1 - \sum_i p_i^2$$

- Entropy

$$[p\ \ \log p + q\ \ \log\ \ q + (1-p-q)\log\ (1-p-q)]$$

Or

$$\sum_i p_i \log p_i$$

- Misclassification Rate

$$1 - \max(p,\ 1-p,\ 1-p-q)$$

3.3. Using gain ratio for attribute selection

Whatever formula we use for the task of attribute selection, introduces an inductive bias, i.e. a preference for one choice rather than other, which is not determined by purely by the data itself but by external forces, such as our preference for simplicity or familiarity with something (Hadzikadic et al., 1995). Such bias can be helpful or harmful, depending on the dataset. We can choose a method that has a bias that we favor, but we cannot eliminate inductive bias altogether. There is no neutral, unbiased method. Clearly it is important to be able to say what is introduced by any particular method of selecting attributes. For many methods this is not easy to do, but for one of the best-known methods we can. Using entropy can be shown to have a bias towards selecting attributes with a large number of values.

In order to reduce effect of bias resulting from the use of information gain, a variant known as Gain Ratio was introduced by the Australian academic Ross Quinlan in his influential system C4.5. Gain Ratio adjusts the information gain for each attribute to allow for the breath and uniformity of the attribute values.

Gain Ratio is defined by the formula:

$$\text{GainRatio} = \text{InformationGain}/\text{SplitInformation}$$

where Split Information is a value based on the column sums.

4. Surveying data

Hospital surveillance unit of ICDDR,B keeps surveillance data and data related to patient in SPSS software. Different types of data are merged into a single file. From observing the data it is evident that the variables in the dataset could be divided into following 5 groups:

Group 1: Social and Behavioral Data: data related to patient's biological information, economic condition, living condition, his/her education level, habits and practices related to hygiene, social and behavioral attributes,
Group 2: Patient History and Primary Diagnosis: health condition of the patient when admitted and any history of previous disease,
Group 3: Pathology Reports: diagnosis report of blood, stool and other check-up,
Group 4: Antibiogram: isolated pathogens and their sensitivity analysis to different antibiotics,
Group 5: Treatment and outcome: treatment, duration of stay and outcome of treatment is also stored in this data set.

These five groups cover the main division of the variables in the Hospital Surveillance data. Our aim of the study is to classify the patient according to their duration of stay in the hospital. We assumed that the higher the duration of stay the more critical is the condition of the patient (a-priory). So we have created a derived variable from duration of stay as criticality (low, medium, high) in the data preprocessing phase.

According to the objective set by us to find out the criticality of the patient on admission we have selected the predictor variables relevant for the study from the groups: *Social and Behavioral Data, Patient History and Primary Diagnosis*, and the target variable criticality is a derivative variable from the "Duration of Stay" variable in the group: *Treatment.*

Hospital Surveillance unit has provided us data in SPSS format.

- It has 26,869 records.
- It has data from 1st January 1996 to 31st December 2007, about 12 years' data.
- It has about 227 attributes/variables.

From our observation we have divided the variables in five major groups which is already been discussed, according to their capture time and characteristics.

Discussing with experts of surveillance unit who have the necessary domain knowledge, according to relevance we have dropped some variables and selected some variable for our study for preprocessing, cleaning, transformation and modeling. Not all records in the group 1, group 2 and group 5 are not kept in the, fields are kept according to relevance and importance to the study.

These fields are further reduced in the data preprocessing phase. Through data survey using statistical tools some of these selected fields are further dropped for lack of variance or lack of valuable information.

Using domain knowledge the variable count decreased from 227 to 40. Frequency Distribution, histogram of the variables and statistical information related to the variables in the selected data set is taken to have a quick understanding about the information content and quality of the data.

We observe that there are missing values in the data and there are skewness and sparsity in some variables, which are dealt in the next section named data preprocessing.

All the records have numerical data. But most of the records have 37 attributes (out of 40) are categorical in nature i.e. they contain few distinct integer values, which corresponds to some categorical values, Like variable "outcome" it has 6 distinct integer values and each integer value correspond to a categorical value (1 = Cured, 2 = Illness continued, 3 = Died, 4 = Absconded, 5 = Others, 9 = Unknown).

Remaining 3 attributes have continuous values – durstady, durstahr, agemm.

5. Data preprocessing

Data mining is about working with data, which to a greater or lesser degree reflects some real-world activity, event, or object. Data need to be prepared so that the information enfolded within it is most easily accessed by the data mining tools.

Preparation of data is not a process that can be carried out blindly. There is no automatic tool that can be pointed at a dataset and told to "fix" the data. There will remain as much art as science in good data preparation. Because data preparation techniques cannot be completely automated, it is necessary to apply them with knowledge of their effect on the data being prepared (Pyle Dorian, Data Preparation for Data Mining, Morgan Kaufman, & CD-ROM, 1999).

SPSS software is used from step 1 to 14 and from step 15 to 19 both SPSS and Excel is used to find out missing values, empty values, misclassification error, field transformation, dimensionality reduction i.e., overall data cleaning activities are carried out in the following steps using SPSS and Excel.

Step 1: Data is filtered on the filed "outcome". Only record with "outcome" value = 1/2/3 is taken. Depth of the dataset reduces. Record count is 25,305.
This is done because by observation it is realized that in cases of records with outcome value = 3/4 records are incomplete.
Step 2: After this filtering activity "outcome" field is deleted. So the record width decreases. The number of attributes in the dataset becomes 39.
Step 3: Frequency distribution of "agemm" is taken to find out if there is any missing value. We found two records with missing values Most of the fields in these records are found blank. So these two records deleted. Now record count is 25,303.
Step 4: to reduce sparsity in the variable in "agemm" a new variable "Age" is created from "agemm" – banning "agemm" values in the following way:

0–⩽12: 1 (infants)
>12–⩽60: 2 (early childhood)
>60–⩽120: 3 (later childhood)
>120–⩽180: 4 (adolescent)
180>–⩽720: 5 (adult)
>720: 6 (old)

"Age" has six discrete value from 1 to 6.
"agemm" is deleted; step 4 and 5 compounded a transformation of continuous variable to a discrete variable: agemm to age.
Step 5: "DurationOfStay" a new field is created form this formula = (durstdy ∗ 24) + dursthr
Step 6: "durstdy" and "dursthr" these two fields are deleted; so data width is reduced. Variable count in the training dataset is now 37.
Step 7: "DurationOfStay" field found empty in 17 instances, so those records are deleted, since our target variable would be created based on this variable this field cannot be empty. Now record count is 25,286.
Step 8: A new variable "Criticality" is created from "DurationOfStay" – banning "DurationOfStay" in the following way:

0–⩽48: 1 (Low)
>48–⩽96: 2 (Mid)
>96: 3 (High)

Criticality is the target variable in the record. It has three discrete value ranging from 1 to 3.
Step 9: "DurationOfStay" is deleted. One variable deleted and in the previous step one variable is created, so overall dimensionality of the dataset remains 37 as before.
Step 10: Frequency distribution of all the variables are taken. Found that in five records many of the variables are empty. These records are deleted. Now record count becomes 25,281.
Step 11: Frequency distribution of the variables are taken again to find out if there is any empty values in any of the variables. Many variables found to have one missing value. All these missing values are in two records. These two records are deleted. Record count is now 25,279.
Step 12: Frequency distribution of the variables are taken. No missing value is found for any of the variables. In some variable some values are found unlabeled, which is deleted.
Record count is now 25,262 in the dataset.
Step 13: Frequency distribution table is taken again for all variables. Found that a value 6 entered in the "Criticality" field wrongly. Reason may be unwanted press in the keyboard. That record is deleted. Record count is now 25,261.
Step 14: Frequency distribution table is taken again for all variables. Found no missing or empty value. All missing and empty values and unlabeled values are dealt with.

Not all attributes may be needed to solve a given data mining problem. In fact, the use of some attributes may interfere with the correct completion of a data mining task. The use of other attributes may simply increase the complexity and decrease the efficiency of an algorithm. This problem is sometimes referred to as *dimensionality curse*, meaning that there are many attributes (dimensions) involved and it is difficult to determine which ones should be used. One solution to this high dimensionality problem is to reduce the number of attributes, which is known as *dimensionality reduction.*

5.1. Reducing the number of attributes

There are several ways in which the number of attributes (or 'features') can be reduced before a dataset is processed.

There are many possible criteria that can be used for determining which attributes to retain, for example:

- ◦ Only retain the best 20 attributes.
- ◦ Only retain best 25% of the attributes.
- ◦ Only retain attributes with an information gain that is at least 25% of the highest information gain of any attribute.

- Only retain attribute that reduce the initial entropy of the dataset by at least 10%.

There is no one choice that is best in all situations, but analyzing the information gain values of all the attributes can help make a good choice (Max, 2007).

In the following two steps we have reduced attributes from 40 to 22 by using Entropy calculation for attribute selection based on information gain.

Step 15: Frequency distribution of "Criticality" fields is taken for calculation of initial entropy E_{start}. Table 1 contains the frequency distribution of criticality. Entropy E_{start} of the dataset is calculated using the formula

$$E = -\sum_{i=1}^{K} p_i \log_2 p$$

$$\begin{aligned} E_{start} &= -(21510/25261)\log_2(21510/25261) \\ &\quad - (2549/25261)\log_2(2549/25261) \\ &\quad - (1202/25261)\log_2(1202/25261) \\ &= 0.197468822 + 0.333890491 + 0.209052308 \\ &= \mathbf{0.740411621} \end{aligned}$$

Step 16: Cross-Tab of all predictor variables with the target variable "Criticality" is taken for calculation of Entropy E_{new}, the average entropy of the training sets resulting from splitting on a specific attribute. For an Example, cross tabulation of attribute "Sex" is shown with target variable "Criticality" in Table 2 and calculation of E_{new} for "sex" is also shown.

E_{new} is now calculated by forming a sum as follows.

(a) For every non-zero value V in the main body of the table (i.e. the part which is colored Vanilla), subtract ($V \times \log_2 V$).

(b) For every non-zero value S in the column sum row (i.e. the green part), add $S \times \log_2 S$

Finally divide the total by number of instances N.

Example

For attribute "Sex",

$$\begin{aligned} E_{new} &= (-12,573 * \mathrm{LOG}(12,573,2) - 8,937 * \mathrm{LOG}(8,937,2) \\ &\quad - 1,521 * \mathrm{LOG}(1,521,2) - 1,028 * \mathrm{LOG}(1,028,2) \\ &\quad - 707 * \mathrm{LOG}(707,2) - 495 * \mathrm{LOG}(495,2) + 14801 \\ &\quad * \mathrm{LOG}(14801,2) + 10460 * \mathrm{LOG}(10460,2))/25,261 \\ &= \mathbf{0.74037094} \end{aligned}$$

Variables with information gain greater than 0.5% are kept for data mining model building: decision tree generation. In this way we reduce the variables from 37 to 22 variables.*Step 17:* Other variables are discarded i.e. dropped from the dataset. Dimensionality i.e. the width of the dataset is reduced. The count of the records in this dataset is finally 25,261 with variable count 22.

Table 1
Frequency distribution of target variable "Criticality".

		Criticality of patient condition			
		Frequency	Percent	Valid percent	Cumulative percent
Valid	Low	21,510	85.2	85.2	85.2
	Mid	2,549	10.1	10.1	95.2
	High	1,202	4.8	4.8	100.0
	Total	25,261	100.0	100.0	

Table 2
Crosstab Criticality × Sex.

Count		Sex of patients		Total
		Male	Female	
*Criticality of Patient Condition * Sex of patients Crosstabulation*				
Criticality of Patient Condition	Low	12,573	8937	21,510
	Mid	1521	1028	2549
	High	707	495	1202
Total	14,801	10,460	25,261	

Step 18: This data in SPSS is converted to Excel format to make it ready for the next phase i.e., for Data modeling.

Step 19: Criticality field has discrete data values 1, 2, 3 of integer data type. They are converted categorical values of strings data type by using the Excel formula:

= IF ("Criticality" = 1, "Low", IF ("Criticality" = 2, "Mid", IF ("Criticality" = 3, "High", "Default"))). This conversion is done because most of the data modeling tools [Sipina, Tanagra] need target values to be categorical.

5.2. Data modeling

In this research work we use SIPINA (SIPINA, 2008) for generating decision tree model. SIPINA is a Data Mining Software which implements various supervised learning paradigms.

5.2.1. Modeling activities

Step 1: At first all the records in the surveillance dataset that was cleaned in the data preprocessing phase are used in SIPINA.

Step 2: Used C4.5 decision tree classification algorithms for model building. In the dataset Criticality is the target class and other 21 variables are predictor variables. A snapshot of C4.5 algorithm in Sipina is presented in Fig. 2.

Step 3: About C4.5 the following is stated in SIPINA website (SIPINA, 2008):

It implements mainly two key ideas: gain ratio in order to select the right split attribute; post-pruning according the pessimistic error criterion in order to detect the right size of the tree.

5.2.2. Parameters

CL (Confidence level) for pessimistic pruning: It is the confidence level used for the computation of the pessimistic error rate i.e. the upper bound of the confidence interval of the error rate on a leaf.

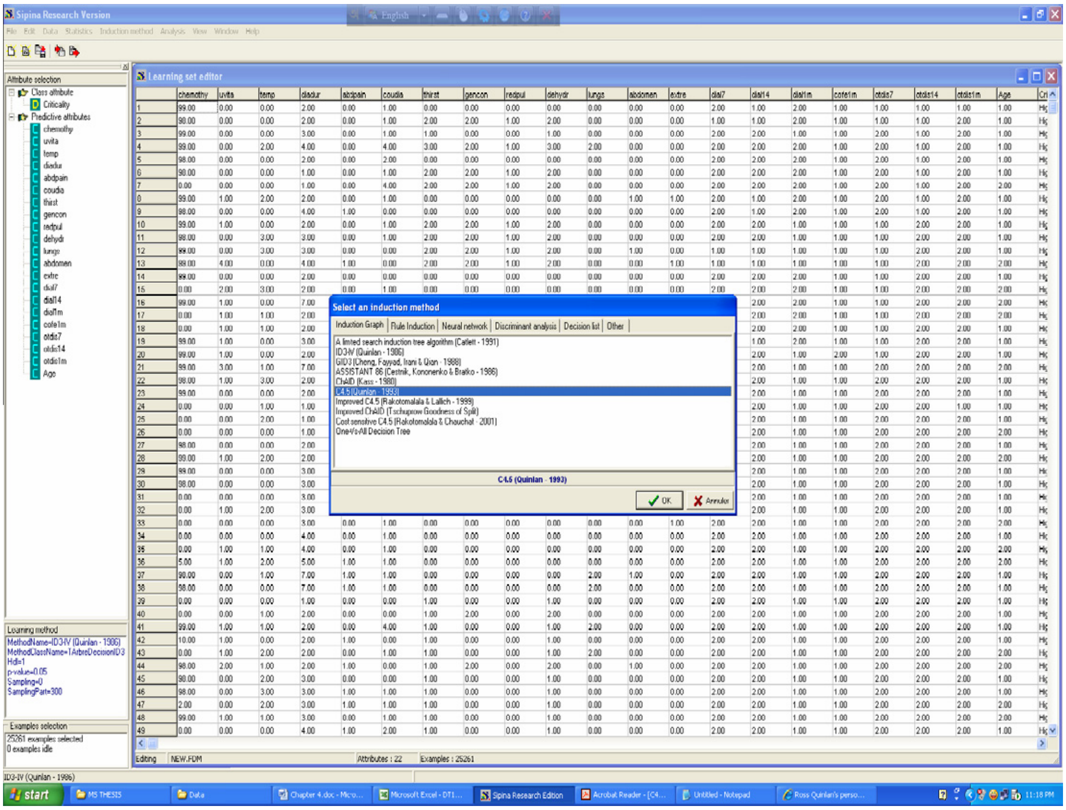
Size of leaves: A split is accepted if two leaves at least have size upper than this threshold (Quinlan, 1993; Kohavi et al., 2002).

Step 4: Two parameters need to be passed. Default value of C.L. for pessimistic pruning = 25% and Size of Leaves = 2. These values are kept in the first run. Selecting those parameters through Sipina is presented in Fig. 3.

Confidence level is used for prediction of tree error rates and affects the pruning process, the lower the confidence level, the higher the amount of pruning that will take place (Jelena Pješivac-Grbovi'c, Thara Angskun, & George Bosilca, 2007). These two parameters are changed and adjusted over and over to get an optimal decision tree model.

Step 5: For *Sampling* default value is "All dataset". The sampling selection is kept in its default value: all records are taken for experiment.

The result from the runs: the generated decision tree models are all evaluated and compared for performance in the next section, in the Evaluation and Implementation phase.



Select an induction method

Induction Graph | Rule Induction | Neural network | Discriminant analysis | Decision list | Other

A limted search induction tree algorithm (Catlett - 1991)
ID3-IV (Quinlan - 1986)
GID3 (Cheng, Fayyad, Irani & Qian - 1988)
ASSISTANT 86 (Cestnik, Kononenko & Bratko - 1986)
ChAID (Kass - 1980)
C4.5 (Quinlan - 1993)
Improved C4.5 (Rakotomalala & Lallich - 1999)
Improved ChAID (Tschuprow Goodness of Split)
Cost sensitive C4.5 (Rakotomalala & Chauchat - 2001)
One-Vs-All Decision Tree

C4.5 (Quinlan - 1993)

✓ OK   ✗ Annuler

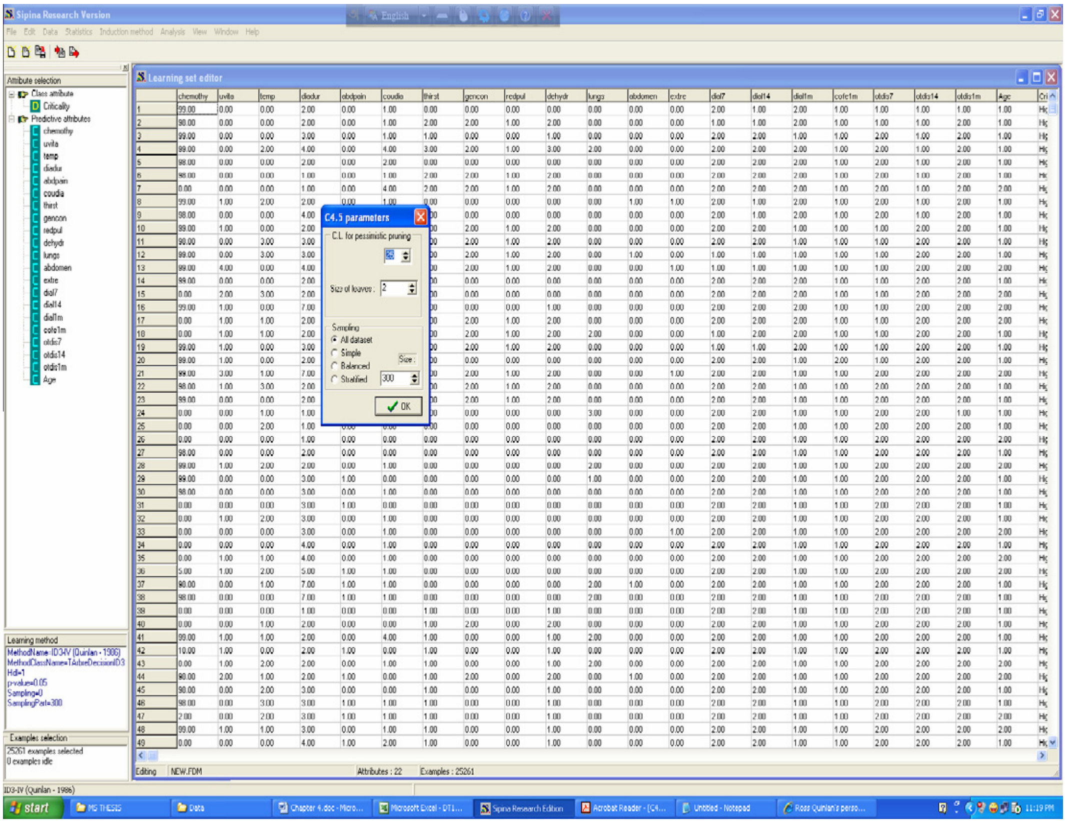
Learning method
MethodName=ID3-IV (Quinlan - 1986)
MethodClassName=TArbreDecisionID3
Hdl=1
p-value=0.05
Sampling=0
SamplingPart=300

Examples selection
25261 examples selected
0 examples idle

Editing   NEW.FDM   Attributes : 22   Examples : 25261

ID3-IV (Quinlan - 1986)

Fig. 2. Selecting the C4.5 algorithm.



C4.5 parameters

C.L. for pessimistic pruning
25

Size of leaves : 2

Sampling
● All dataset
○ Simple
○ Balanced
○ Stratified
Size : 300

✓ OK

Learning method
MethodName=ID3-IV (Quinlan - 1986)
MethodClassName=TArbreDecisionID3
Hdl=1
p-value=0.05
Sampling=0
SamplingPart=300

Examples selection
25261 examples selected
0 examples idle

Editing   NEW.FDM   Attributes : 22   Examples : 25261

ID3-IV (Quinlan - 1986)

Fig. 3. SIPINA C4.5 passing parameters.

6. Evaluations and deployment

For evaluation purpose, we use dataset that have been processed in the previous stage. We build decision trees on two data sets. First dataset contain all records from database and the other contains selected records from database such that all classes (high, low and mid) have equal number of records.

6.1. Model 1 (all instances from Dataset 1 used in classification)

At first we use the dataset that have been processed in preprocessing step. The following parameters are set in Sipina to generate a Decision Tree model:-

Learning Method: C4.5 (Quinlan – 1993)
Confidence Level = 25
Leaf size = 2
Sampling = 0 (All Dataset)

General information about the tree (Fig. 4):

Nodes: 29
Leaves: 15
Max Depth: 7

15 decision rules are generated.

For Class variable criticality the Confusion Matrix for Dataset 1 is given in Table 3.

The *accuracy* (*AC*) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{a+e+i}{a+b+c+d+e+f+g+h+i} \tag{1}$$

Here, *AC* = 85.43 %

The *recall* or *true positive rate* (*TP*) is the proportion of positive cases that were correctly identified, as calculated using the equation:

Table 3
Confusion Matrix for Dataset 1.

Actual	Classified		
	High	Mid	Low
High	118 (a)	2 (b)	1082 (c)
Mid	77 (d)	3 (e)	2469 (f)
Low	51 (g)	0 (h)	21459 (i)

$$TP = \frac{d}{c+d}$$

$$TP_{HIGH} = \frac{a}{a+b+c}$$

Here, TP_{HIGH} = 9.817%

$$TP_{MID} = \frac{e}{d+e+f}$$

Here, TP_{MID} = ***0.118 %***

$$TP_{LOW} = \frac{i}{g+h+i}$$

Here, TP_{LOW} = **99.76%**

The false positive rate (FP) is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$FP_{HIGH} = \frac{b+c}{a+b+c}$$

$$FP_{MID} = \frac{d+f}{d+e+f}$$

$$FP_{LOW} = \frac{g+h}{g+h+i}$$

FP(HIGH) = 90.18%
FP(MID) = 99.88%
P(LOW) = 0.237%

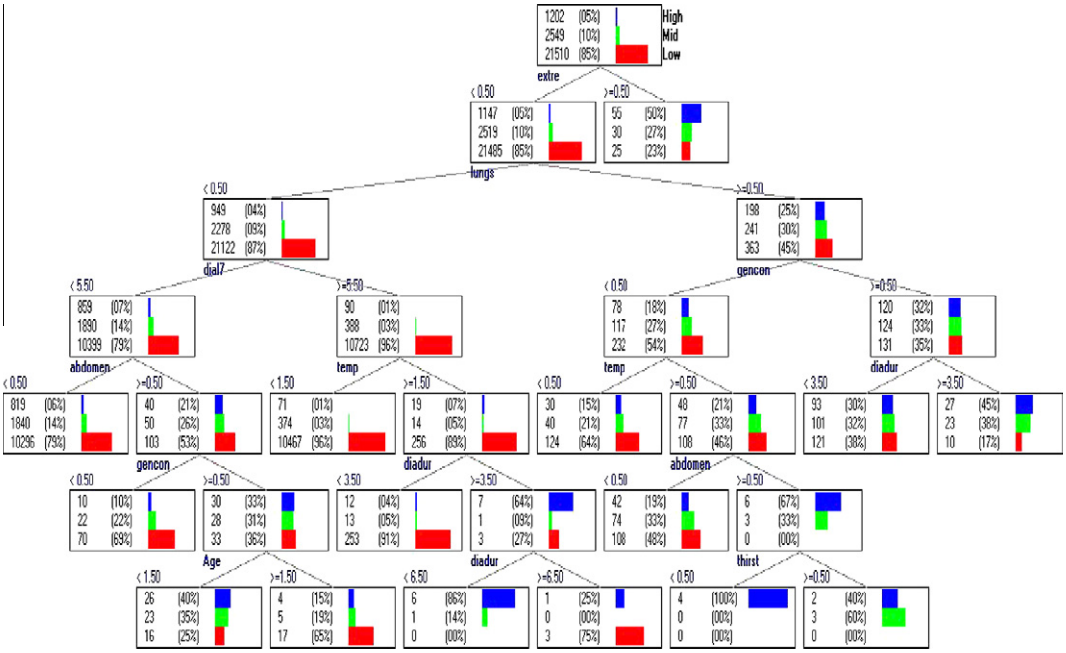


Fig. 4. Decision tree generated by C4.5.

Precision (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P_{HIGH} = \frac{a}{a+d+g}$$
$$P_{MID} = \frac{e}{b+e+h}$$
$$P_{LOW} = \frac{i}{c+f+i}$$

P(HIGH) = 47.97%
P(MID) = 60%
P(LOW) = 85.8%

Performance metrics for model 1 is presented in Fig. 5. It clearly shows a bias towards criticality class "low". The accuracy determined using Eq. 1 is not giving adequate performance measure, since number of "Low" criticality cases is much greater than the rest of the cases in the dataset (Michalski, Bratko, & Kubat, 1998). Even though the classifier shows poor performance to classify "High" and "Low" cases, due to **i** element in the equation 5.1 i.e. True Positive cases of "Low" value is so high that Accuracy becomes high (wrongly) for this classifier.

6.2. Model 2 (all instances from Dataset 2 is used for classification)

As we discussed in the previous section about the low performance of the tree built on the data set 1, we did not proceed further with the analysis with this dataset. We realize that we have to reprocess the dataset to remove some bias to the lower value and also need to reduce dimensionality to some extend to in hope to increase the overall performance of the classier.

The following data processing steps are followed:

Step 1: From frequency distribution table we can see that instance of class "Low" has 21,510, "Mid" has 2,549 and High has 1,202 records. Percentage-wise "Low" = 85.2%, Mid = 10.1% and High = 4.8%.
Discussing with domain expert we realize that this percentage is acceptable. Among the admitted patients this ratio Low : - Mid : High = 17 : 2 : 1 represents somewhat true picture.
Since for our research purpose this ratio or proportion is not much relevant and we are more concerned about the attributes of the dataset which represent the physical condition of the patients, we use random selection to select 1,202 records from "Mid" and Low" class, same as the instance count of class "High": to make a balance sample out of the biased surveillance dataset.

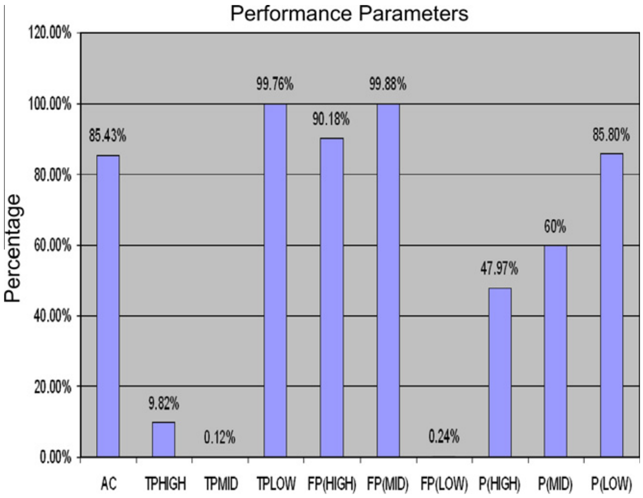


Fig. 5. Performance parameters for the Model 1.

Step 2: Frequency Distribution of the Dataset for all variables are taken. Depth of the dataset is now reduced–it has now 3607 records, with equal number of High, Mid, Low classes. We have following observation:
(i) One obvious fact is that variability of the values of the field "chemothy" is very high. That filed is removed.
(ii) Since the number of records has been dropped down, distribution of the variable have changed drastically. So we need to check for the Information Gain of the variables again. So we need to take Cross-tabulation criticality with all the remaining variables.

Step 3: Cross-tabulation is taken (Criticality × All other variables). Calculated information gain Applying the previous selection criteria of attributes (Select attribute with Information Gain $\geqslant$ 0.5%), three of the variables are dropped. Now variables count to 18, 17 predictor variables and 1 target/class variable.

Now again after this second time data cleaning and data preprocessing we start to model the dataset using SIPINA with C4.5 decision tree classification algorithm with the same default parameters as the first run:

Learning Method: C4.5 (Quinlan – 1993)
Confidence Level = 25
Leaf size = 2
Sampling = 0 (All Dataset)

General information about the tree:

Nodes: 61 (32 Nodes increased from 29)
Leaves: 31 (16 Leaves increase from 15)
Max Depth: 12 (5 Max Depth increased from 7)

If we compare to the previous tree this new tree with the new dataset has 32 more nodes, 16 more leaves and Max Depth has increase by 5 levels, i.e., the new decision tree is double in size compared to the previous tree. So complexity which depends on the terminal nodes has considerable increased. This shows problem of overfitting, which would result in excessively large rule sets with very low predictive power for previously unseen data.

Truly a large number of rules are generated: 31 decision rules; they are not mentioned since they are due to overfitting; they are bound to be very much specialized, with less generalization capacity: good for classification of training set but bad on unseen instances, i.e. misclassification error is less, but prediction error would be high (Ignizio, 1991).

We need a smaller and simpler tree in order to increase its predictive accuracy. For this purpose we need to employ tree optimization technique, tree pruning.

We need a tradeoff between misclassification error and tree complexity (Roman, 2004).

Before proceeding further with tree pruning, let us take the confusion matrix for the new tree; and also calculate AC, TP_{HIGH}, TP_{MID} and TP_{LOW} and compare the new values with the values for the previous tree (Fig. 4).

For class variable criticality the Confusion Matrix for Dataset 2 is presented in Table 4.

From the table it is visible that the classification accuracy has increased. Even record count is reduced in the new dataset, TP count for High and Mid class have increased to a considerable amount. Fig. 6 shows the classification accuracy is increased i.e.,

Table 4
Confusion Matrix for Dataset 2.

Actual	Classified		
	High	Mid	Low
High	680 (a)	416 (b)	106 (c)
Mid	30 (d)	944 (e)	228 (f)
Low	14 (g)	65 (h)	1123 (i)

the misclassification error is decreased with more complex tree penalizing generalization capacity of the tree.

Now for Dataset 2:

AC = 76.18%
TPHIGH = 56.57%
TPMID = 78.54%
TPLOW = 93.43%
FP(HIGH) = 43.43%
FP(MID) = 21.46%
FP(LOW) = 6.57%
P(HIGH) = 93.92%
P(MID) = 66.25%
P(LOW) = 77.076%

We can construct the following tables to represent features of the datasets (Table 5) and comparison of classification performance of the decision trees generated from the datasets (Table 6).

A comparison between model 1 and model 2 is presented in Fig. 7. We can observe that we have a considerable development in classification of high and mid classes, when we used the Dataset 2 with equal distribution of class instances instead of using Dataset 1 with class distribution of high variability for learning the decision tree.

6.3. Model 3 (Dataset 1 is divided into training and testing set)

Now the instances of Dataset 1 and Dataset 2 are divided into training set and test set.

Training and testing with Dataset 1:
Initial parameters same as before:

Learning method: C4.5 (Quinlan – 1993)
Confidence level = 25
Leaf size = 2
Sampling = 0 (All Dataset)

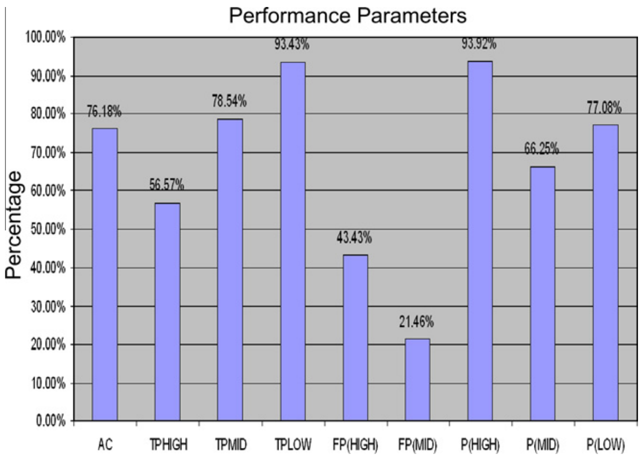


Fig. 6. Performance parameters for Data Set 2.

Table 5
Two datasets used for decision tree model learning.

Dataset	Description	Classes	Attributes		Instances	
			Categorical	Continuous	Training set	Test set
Dataset 1	Varying Distribution of Class instances	3	22		25,261	
Dataset 2	Equal Distribution of class instances	3	18		3,606	

Table 6
Comparison of classification accuracy between 2 models.

Dataset	Test set (instances)	Correctly Classified	Accuracy %	TP_{HIGH} %	TP_{MID} %	TP_{LOW} %
Dataset 1	25,261 (Learning Set)	21,580	85.43	9.817	0.118	99.76
Dataset 2	3,607 (Learning Set)	2,747	76.18	56.57	78.54	93.43

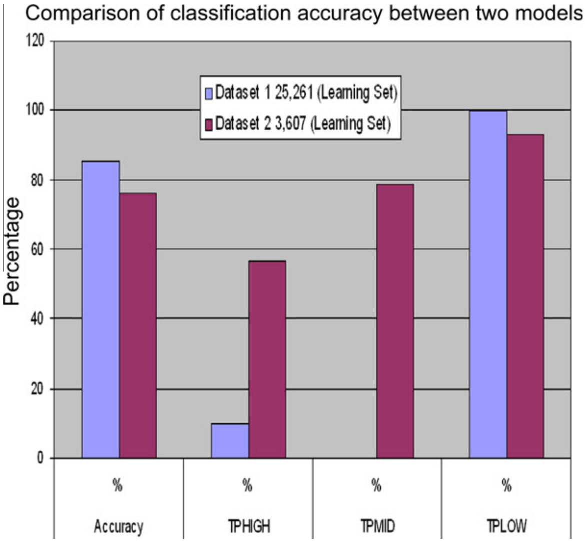


Fig. 7. Graphical comparison of classification accuracy between two Models.

General information about the tree:

Nodes: 31
Leaves: 16
Max Depth: 9

16 decision rules generated.

Confusion matrix for learning dataset for model 1 is presented in Table 7.

Classification accuracy *AC* = 85.15%
TPHIGH = 5.39%
TPMID = 1.45%
TPLOW = 99.83%
FP(HIGH) = 94.61%

Table 7
Confusion Matrix for Dataset 1 (Training).

Actual	Classified		
	High	Mid	Low
High	32	7	555
Mid	17	19	1279
Low	11	7	10703

Table 8
Confusion Matrix for Dataset 1 (Testing).

Actual	Predicted		
	High	Mid	Low
High	34	18	556
Mid	22	11	1201
Low	24	10	10755

FP(MID) = 98.56%
FP(LOW) = 0.168%
P(HIGH) = 53.33%
P(MID) = 57.58%
P(LOW) = 85.37%

Confusion matrix for testing dataset for model 1 is presented in Table 8.

Classification accuracy AC = 85.5%
TPHIGH = 5.59%
TPMID = 0.89%
TPLOW = 99.68%
FP(HIGH) = 94.41%
FP(MID) = 99.11%
FP(LOW) = 0.315%
P(HIGH) = 42.5%
P(MID) = 28.21%
P(LOW) = 85.96%

So we can see that classification accuracy and prediction accuracy of this tree is not satisfactory for the “High” and “Mid” classes.

If we analyze the ROC (Receiver Operating Characteristics) curve taken with parameter “age” we can also find that the classifier performs poorly for class “High” and “Mid” (the curve is far away from the best possible line (0, 1), it is under the diagonal i.e. random guessing line, it has the lesser prediction capacity then flipping a coin (0.5), only it case of predicting class “Low”, it works well, the line is nearer to the (0, 1) line. Figs. 9a–c, depict the ROC curve for different classes.

6.4. Model 4 (Dataset 2 is divided into training and testing dataset)

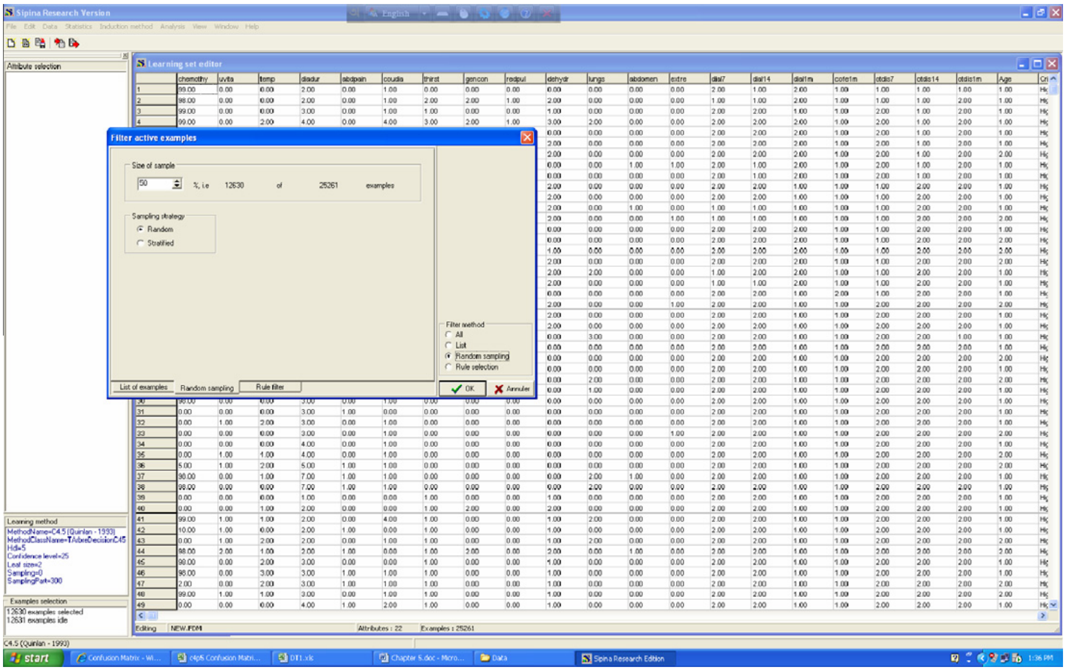
Finally, we generate a decision tree using 50% of the instances from data set 2 for learning and rest for testing. With the increase in size of the tree, misclassification error is decreased and in case of maximum tree, misclassification error will be equal to 0. Now In case of dataset 2 we generate a decision tree using 50% of the instances for learning and rest for testing prediction accuracy on unseen instance (1803). Selecting training and testing dataset from SIPINA is presented in Fig. 8.

Complex decision tree poorly perform on independent data. Performance of decision tree on independent data is called true prediction power of the tree. Therefore, the primary task is to find the optimal proportion between the tree complexity and misclassification error. The task is achieved through cost-complexity function:

$$R_\alpha(T) = R(T) + \alpha * (\widetilde{T})$$

where

$R(T)$: misclassification error of the tree T,
$\alpha(\widetilde{T})$: complexity measure,
$\widetilde{T}$: total number of terminal nodes in the tree,
α: parameter is found through the sequence of in-sample testing when a part of learning sample is used to build the tree,



Size of sample
50 %, i.e 12630 of 25261 examples

Sampling strategy
● Random
○ Stratified

Filter method
○ All
○ List
● Random sampling
○ Rule selection

Learning method
MethodName=C4.5 (Quinlan - 1993)
MethodClassName=TArbreDecisionC45
Hdl=5
Confidence level=25
Leaf size=2
Sampling=0
SamplingPart=300

Examples selection
12630 examples selected
12631 examples idle

Editing NEW.FDM Attributes : 22 Examples : 25261

C4.5 (Quinlan - 1993)

Fig. 8. Selecting active examples in SIPINA for learning and inactive examples will be used for testing prediction accuracy on unseen instances.

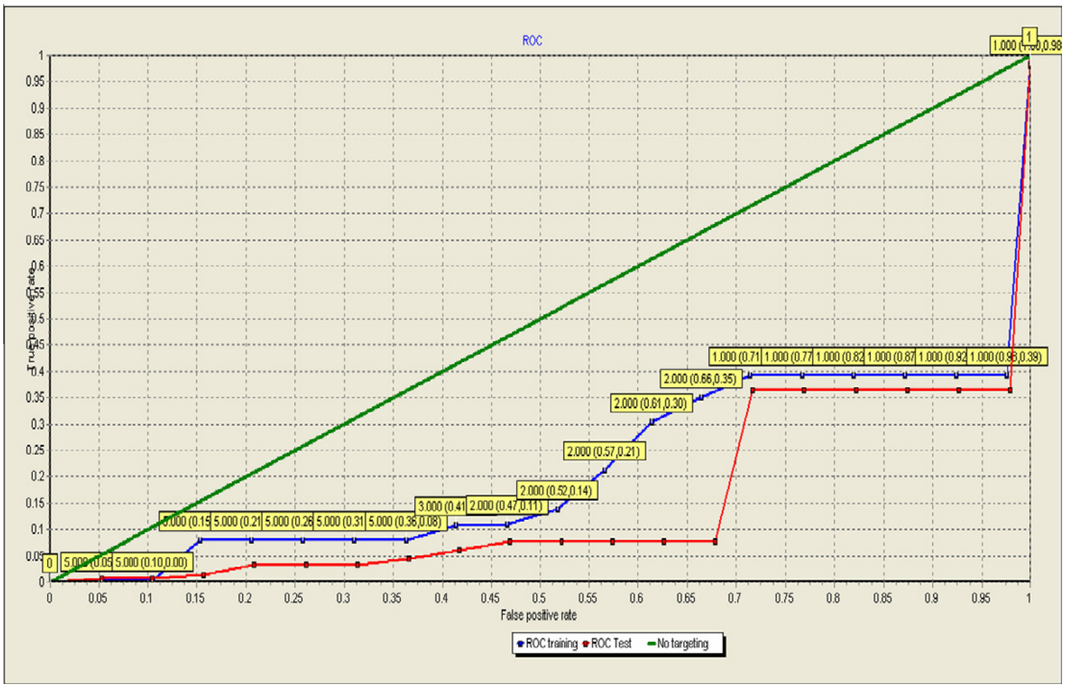


Fig. 9a. ROC Curve for Class = "High".

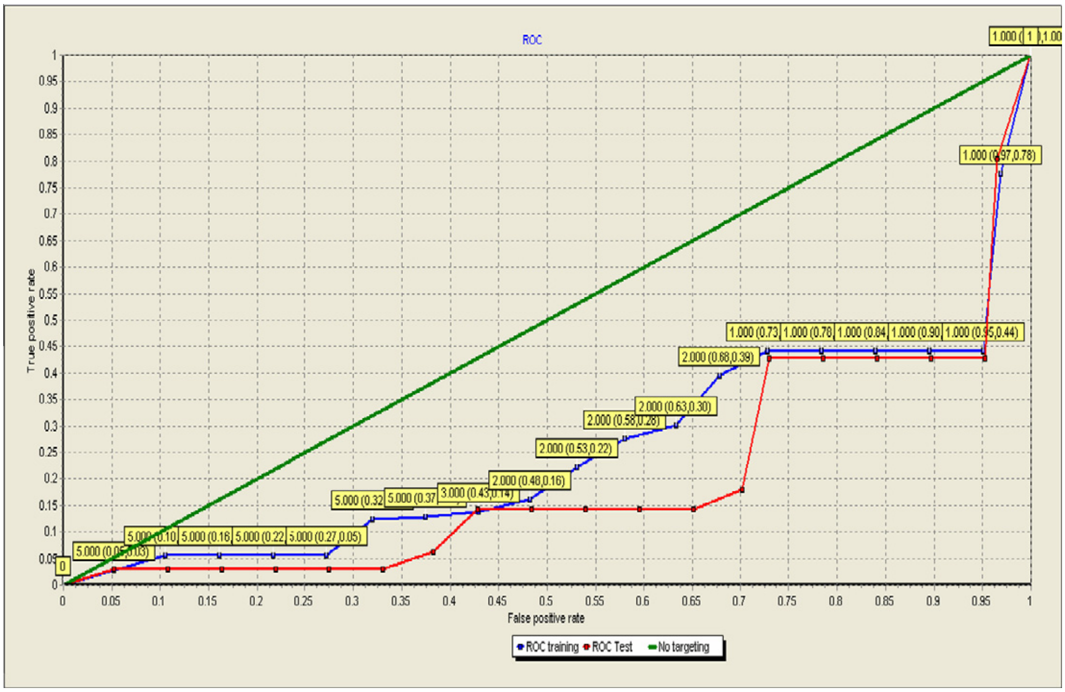


Fig. 9b. ROC Curve for Class = "Mid".

the other part of the data is taken as testing sample (Ignizio, 1991).

We adjust the initial parameters of the C4.5 decision Tree classification algorithm in SIPINA to find an optimal proportion between the tree complexity and misclassification error. We run the learning process over and over by adjusting the parameters: "C.L. for pessimistic pruning" and "Leaf Size" several times and tried to arrive at an optimal tree, where the tree may have more misclassification error but a greater generalization capacity and consequently increased prediction accuracy on unseen instances.

Final (optimal) tree:
Initial parameters:
Learning method: C4.5 (Quinlan – 1993)
Confidence level = 90
Leaf size = 25
Sampling = 0 (All Dataset)

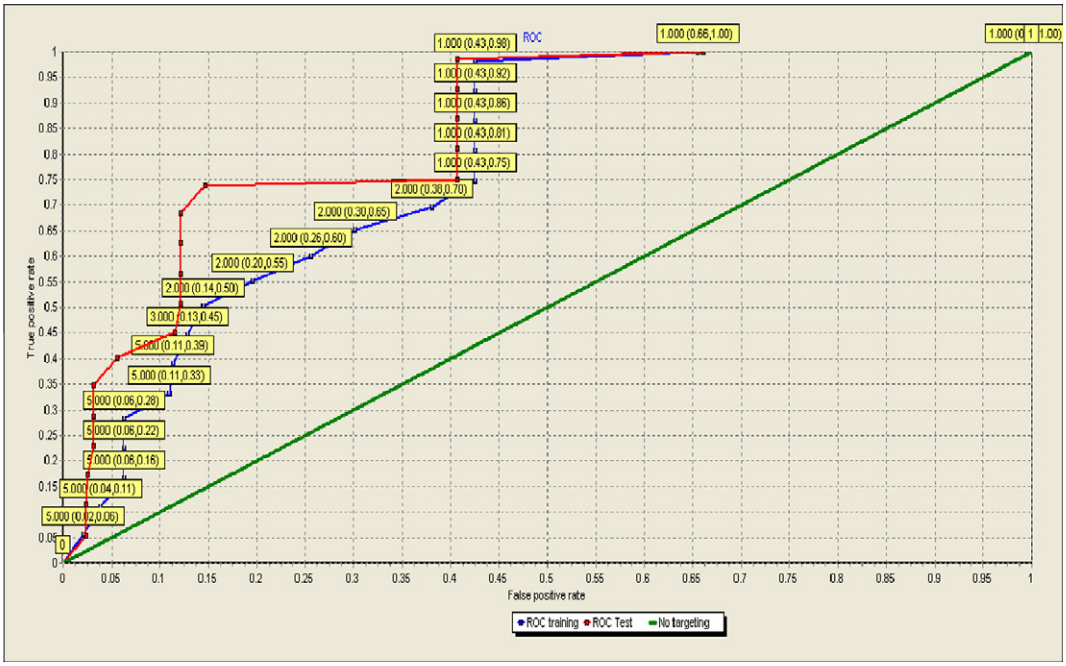


Fig. 9c. ROC Curve for Class = "Low".

Confidence level (C.L) changed from default value 25–90 and Leaf Size from 2 to 25.

This leads to through more pruning a decision tree model with less number of terminal nodes i.e., less complexity (Fig. 10).

General information about the tree (Fig. 10):

Nodes: 13
Leaves: 7
Max Depth: 6

Seven (7) decision rules are generated:

Rule 1:
IF cofe1m ⩾ 1.50 THEN Criticality in [High] with accuracy 0.8161 on (142, 2, 30)

Rule 2:
IF cofe1m < 1.50 and dial1m < 1.50 THEN Criticality in [Low] with accuracy 0.7451 on (40, 64, 304)

Rule 3:
IF cofe1m < 1.50 and dial1m ⩾ 1.50 and diadur < 1.50 and dehydr < 0.50 THEN Criticality in [Low] with accuracy 0.8306 on (12, 30, 206)

Rule 4:
IF cofe1m < 1.50 and dial1m ⩾ 1.50 and diadur < 1.50 and dehydr ⩾ 0.50 THEN Criticality in [Mid] with accuracy 0.4818 on (59, 66, 12)

Rule 5:
IF cofe1m < 1.50 and dial1m ⩾ 1.50 and diadur ⩾ 1.50 and thirst < 1.50 THEN Criticality in [Mid] with accuracy 0.6051 on (159, 308, 42)

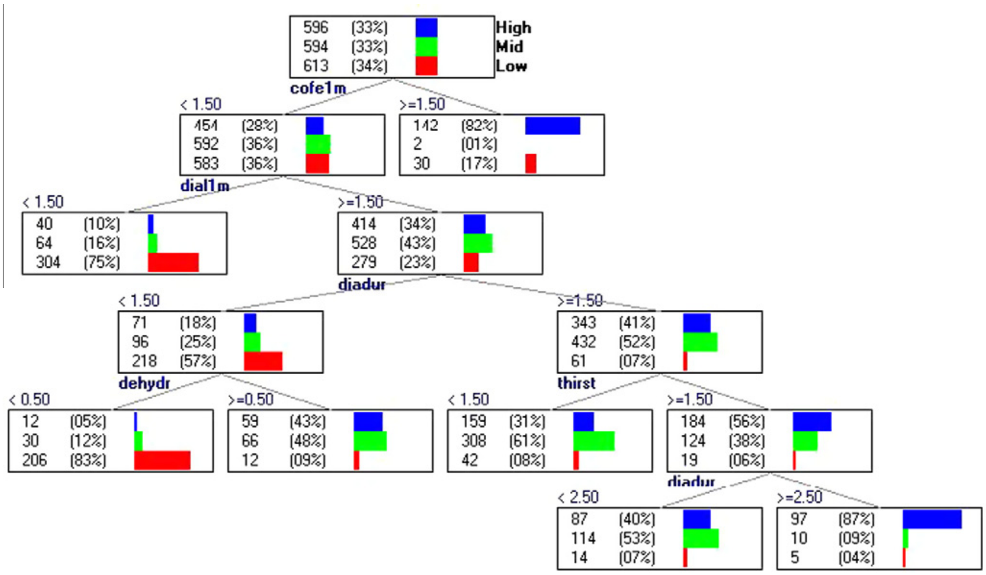


Fig. 10. Decision tree generated by C4.5 using Dataset 2 (Data is divided into traiing and testing data).

Table 9
Confusion Matrix for Dataset 2 (Training).

Actual	Classified		
	High	Mid	Low
High	239	305	52
Mid	12	488	94
Low	35	68	510

Table 10
Confusion Matrix for Dataset 2 (Testing).

Actual	Predicted		
	High	Mid	Low
High	254	296	56
Mid	14	483	111
Low	23	68	498

Table 11
Comparison of performance between two models learned from Dataset 1 and Dataset 2.

Dataset	Dataset 1	Dataset 2
Test Set (Instances)	12,631	18,03
Correctly Classified	10,800	1,235
Incorrectly Classified	1,831	568
Overall Accuracy %	85.5	68.5
TPHIGH %	5.592	41.91
TMID %	0.891	79.44
TPLOW %	99.68	84.55
FPHIGH %	94.41	58.09
FPMID %	99.11	20.56
FPLOW %	0.315	15.45
PHIGH %	42.5	87.29
PMID %	28.21	57.02
PLOW %	85.96	74.89

Rule 6:
IF $cofe1m < 1.50$ and $dial1m \geqslant 1.50$ and $diadur \geqslant 1.50$ and $thirst \geqslant 1.50$ and $diadur < 2.50$ THEN Criticality in [Mid] with accuracy 0.5302 on (87, 114, 14)
Rule 7:
IF $cofe1m < 1.50$ and $dial1m \geqslant 1.50$ and $diadur \geqslant 1.50$ and $thirst \geqslant 1.50$ and $diadur \geqslant 2.50$ THEN Criticality in [High] with accuracy 0.8661 on (97, 10, 5)

Confusion matrix for training dataset of optimal tree is presented in Table 9.

Classification accuracy $AC = 68.61\%$
TPHIGH = 40.10%
TPMID = 82.15%
TPLOW = 83.20%
FP(HIGH) = 59.90%
FP(MID) = 17.85%
FP(LOW) = 16.80%
P(HIGH) = 83.57%
P(MID) = 56.68%
P(LOW) = 77.74%

Confusion matrix for testing dataset of optimal tree is presented in Table 10.

Classification accuracy $AC = 68.50\%$
TPHIGH = 41.91%
TPMID = 79.44%
TPLOW = 79.44%
FP(HIGH) = 58.09%
FP(MID) = 20.56%
FP(LOW) = 15.45%
P(HIGH) = 87.29%
P(MID) = 57.02%
P(LOW) = 74.89%

If we compare between the performances of the DT Model learned from Dataset 1 and the DT Model learned from Dataset 2 (Comparison of Performance Table 11, Fig. 11), and see the ROC Graph (Fig. 12) plotted for the points (TP_{High}, FP_{High}), (TP_{Mid}, FP_{Mid}), (TP_{Low}, FP_{Low}), ROC curves (Figs. 13a–c) taken with parameter "age" and observe it we can clearly see that we have got a more optimized model which has optimal proportion between the tree complexity and misclassification error. This model has higher generalization capacity, i.e., it can predict with higher accuracy about the class of new unseen instances.

We can observe that the ROC curve has considerable showing better performance in prediction accuracy. Most of the time the curve is over the random guess line (the diagonal line connecting point (0, 0) to point (1, 1).

7. Conclusion and future works

The primary purpose of Hospital Surveillance data is to forecast emerging epidemics so that precautionary measures can be taken to reduce the level of damage caused by an epidemic. Traditionally time series analysis or cluster analysis has been used in this arena to recognize any cyclic patterns in epidemic waves and to generate regression formula which can be used for forecasting. In this thesis

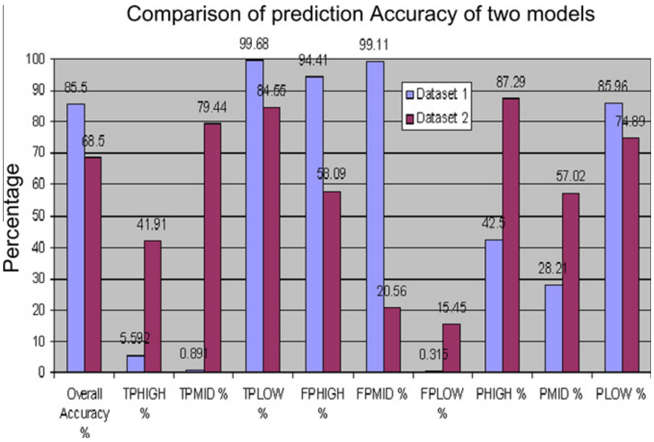


Fig. 11. Comparison of prediction accuracy between two models.

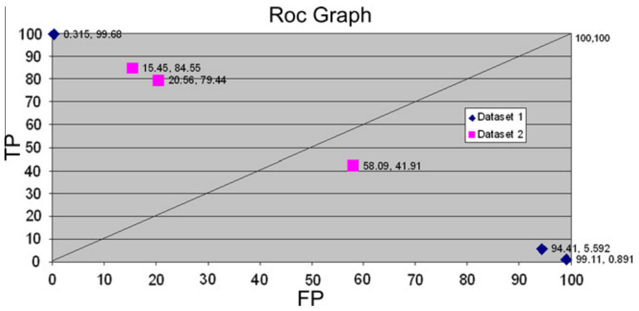


Fig. 12. Comparison of prediction accuracy using ROC graph: points (FP, TP) shown for low, mid and high classes in two models.

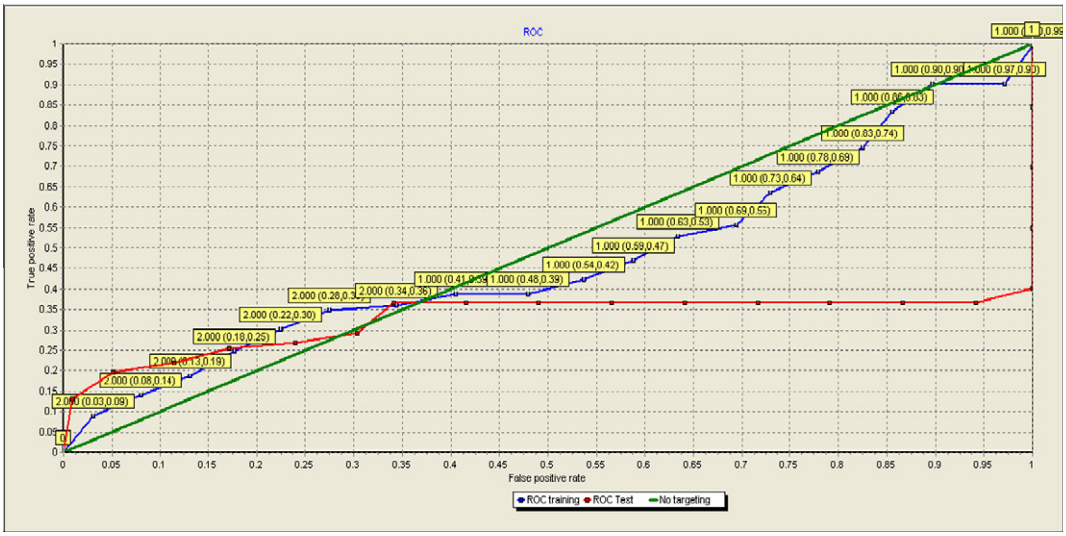


Fig. 13a. ROC Curve for Class = "High".

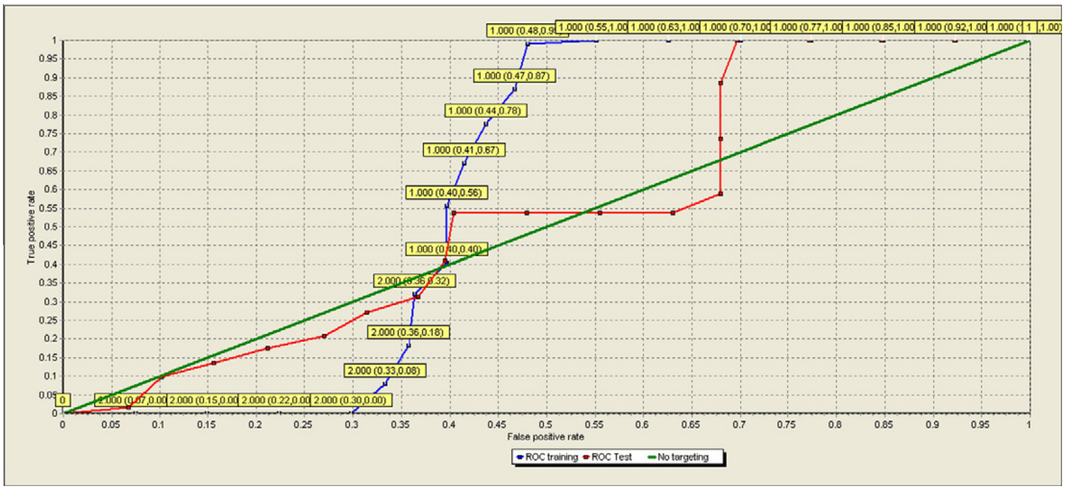


Fig. 13b. ROC Curve for Class = "Mid".

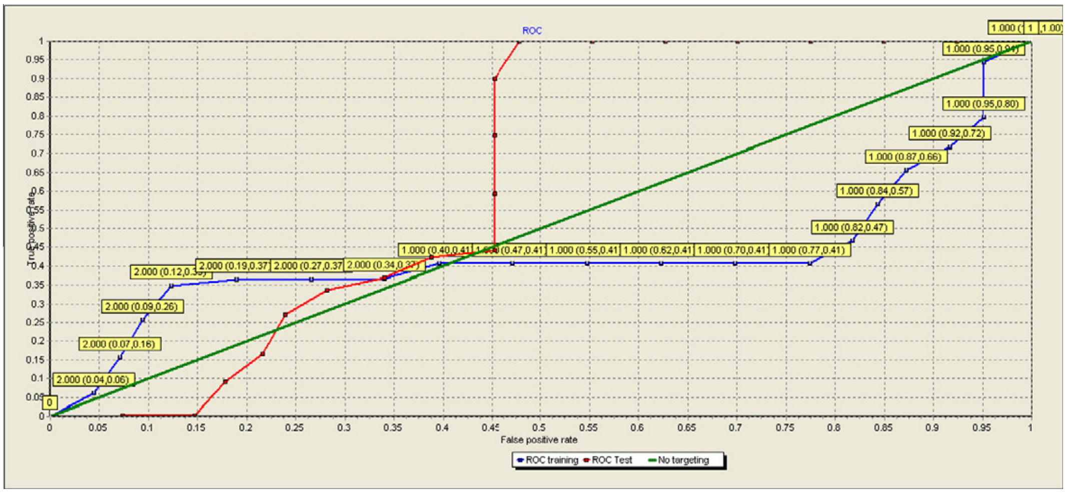


Fig. 13c. ROC Curve for Class = "Low".

we have used data mining tools and techniques to classify patient using decision tree models generated from Historical Data of Surveillance System.

References

Bereznicki, Bonnie J., Peterson, Gregory M., Jackson, Shane L., Haydn Walters, E., Fitzmaurice, Kimbra D., & Gee, Peter R. (2008). Data-mining of medication records to improve asthma management. *MJA, 189*(1), 21–25<http://www.mja.com.au/public/issues/189_01_070708/ber11059_fm.html> .

Brossette, Stephen E. et al. (1998). Association rules and data mining in hospital infection control and public health surveillance. *Journal of the American Medical Informatics Association, 5*, 373–381.

Brossette, S. E., & Hymel, P. A., Jr. (2008). Data mining and infection control. Clinics in Laboratory Medicine, 28 (1), 119–126.

Buntinx, F., Truyen, J., Embrechts, P., Moreel, G., & Peeters, R. (1992). Evaluating patients with chest pain using classification and regression trees. *Fam Pract, 9*(2), 149–153<http://www.ncbi.nlm.nih.gov/pubmed/1505701?ordinalpos=1&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_DiscoveryPanel.Pubmed_Discovery_RA&linkpos=1&log$=relatedarticles&logdbfrom=pubmed> .

CureHunter-precision medical data mining. (2008). <http://www.psychsplash.com/2007/09/14/curehunter-precision-medical-data-mining/>.

Pyle Dorian, Data Preparation for Data Mining, Morgan Kaufman, Morgan Kaufmann; Book & CD-ROM 1st edition (1999) ISBN-10: 1558605290, ISBN-13: 978-1558605299.

Hadzikadic, M., Hakenewerth, A., Bohren, B., Norton, J., Mehta, B., & Andrews, C. (1995). Concept formation vs. logistic regression: predicting death in trauma patients. In *Proc annu symp comput appl med care* (1995). (pp. 198–202). <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2579083#reference-sec>.

Houston, Andrea L., Chen, Hsinchun, Hubbard, Susan M., Schatz, Bruce R., Ng, Tobun D., Sewell, Robin R., et al. (1999). Tolle medical data mining on the internet: research on a cancer information system. *Artificial Intelligence Review, 13*(5–6), 437–466. 30.

ICDDR,B, (2008), <http://www.icddrb.org/org/orgunits.jsp?idDetails=103&searchID=103>.

Ignizio, J. P. (1991). An introduction to expert systems: the development and implementation of rule based expert systems, McGraw-Hill, Inc., (pp. 402).

Kohavi, R., & Quinlan, R. (2002). Decision tree discovery, Klosgen, & Zytkow (Eds.), handbook of data mining and knowledge discovery, Chapter 16.1.3, Oxford University Press. (pp. 267–276).

Lamma, E., Mello, P., Nanetti, A., Riguzzi, F., Storari, S., & Valastro, G. (2006). Artificial intelligence techniques for monitoring dangerous infections. *IEEE Transactions on Information Technology in Biomedicine, 10*(1), 143–155.

Ma, L., Tsui, F. C., Hogan, W. R., Wagner, M. M., & Ma, H. (2003). A framework for information control surveillance using association rules, AMIA annual symposium, Proc. (pp. 410–414).

Mair, J., Smidt, J., Lechleitner, P., Dienstl, F., & Puschendorf, B. (1995). A decision tree for the early diagnosis of acute myocardial infarction in nontraumatic chest pain patients at hospital admission. *Chest., 108*(6), 1502–1509<http://www.ncbi.nlm.nih.gov/pubmed/7497751?ordinalpos=1&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_DiscoveryPanel.Pubmed_Discovery_RA&linkpos=3&log$=relatedarticles&logdbfrom=pubmed> .

Max, Bramer. Principles of Data Mining, Springer London Ltd, Published, (2007), ISBN 13: 9781846287657.

Michalski, R. S., Bratko, I., & Kubat, M. (Eds.): Machine learning, and data mining: methods and applications, Wiley (1998).

Moser, S. A., Jones, W. T., & Brossette, S. E. (1999). *Application of data mining to intensive care unit microbiologic data Emerg Infect Dis., 5*(3), 454–457.

Jelena Pješivac-Grbovi'c, Graham E. Fagg, Thara Angskun, George Bosilca, & Jack J. (2007). Dongarra, Decision Trees and MPI Collective Algorithm Selection Problem, Innovative Computing Laboratory, Parallel Computing, 33 (9), 613–623 (Year of publication: 2007, ISSN:0167-8191).

Quinlan, R. (1993). *C4.5: programs for machine learning*. Morgan Kaufman Publishers.

Timofeev Roman, Classification and Regression Trees (CART), Theory and Applications, Humboldt University, Berlin, Berlin, December 20, (2004).

SIPINA. (2008). <http://eric.univ-lyon2.fr/~ricco/sipina.html>.

Stilou, S., Bamidis, P. D., Maglaveras, N., & Pappas, C. (2001). Mining association rules from clinical databases: an intelligent diagnostic process in healthcare. *Stud Health Technol Inform, 84*(Pt 2), 1399–1403.

Tseng, V. S., Chao-Hui, Lee, Chia-Yu Chen, J. (2008). An integrated data mining system for patient monitoring with applications on asthma care. computer-based medical systems, 2008. CBMS apos; 08. 21st IEEE international symposium on volume, Issue 17–19, (pp. 290–292).

Xing Yanwei, Jie Wang, Zhihong Zhao, Yonghong Gao. (2007). Combination data mining methods with new medical data to predicting.outcome of coronary heart disease, convergence information technology. International Conference on Volume, Issue 21–23.(pp. 868–872).<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/4420216/4420217/04420369.pdf?arnumber=4420369>.