T.C.

BAHÇEŞEHİR ÜNİVERSİTESİ

BENCHMARKING DATA MINING TECHNIQUES FOR SEGMENTING DIABETES PATIENTS

Master Thesis

İnayet ADALI

İSTANBUL, 2009

T.C

BAHÇEŞEHİR ÜNİVERSİTESİ

INSTITUTE OF SCIENCES

COMPUTER ENGINEERING

BENCHMARKING DATA MINING TECHNIQUES FOR SEGMENTING DIABETES PATIENTS

Master Thesis

İnayet ADALI

Supervisor: ASSOC.PROF.DR. ADEM KARAHOCA

İSTANBUL, 2009

T.C. BAHÇEŞEHİR ÜNİVERSİTESİ

INSTITUTE OF SCIENCES

COMPUTER ENGINEERING

Name of the thesis: BENCHMARKING DATA MINING TECHNIQUES FOR SEGMENTING DIABETES PATIENTS

Name/Last Name of the Student: İnayet ADALI
Date of Thesis Defense: Jun .12. 2009

The thesis has been approved by the Institute of Sciences.

Prof. Dr. Bülent ÖZGÜLER
Principal of Institute of Sciences

____

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Bülent ÖZGÜLER
Program Coordinator

____

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members	Signature
Assoc.Prof.Dr. Adem KARAHOCA	______________________
Prof.Dr. Nizamettin AYDIN	______________________
Asst.Prof.Dr. Yalçın ÇEKİÇ	______________________

ACKNOWLEDGEMENTS

I would like to express my gratitude to **Assoc. Prof. Dr. Adem Karahoca**, for not only being such a great supervisor but also encouraging and challenging me throughout my academic program.

I also wish to thank to my mother **Engin ADALI,** my father **Ziver ADALI** and **İren AĞAN** for their mental helps on finishing this thesis. Lastly I want to thank to my grandfather **Sabahattin ADALI** for his helps on me to maintain in Bahçeşehir University.

İnayet ADALI

ABSTRACT

BENCHMARKING DATA MINING TECHNIQUES FOR SEGMENTING DIABETES PATIENTS

İnayet ADALI

M.S. Department of Computer

Engineering Supervisor: Assoc. Prof.

Dr. Adem Karahoca

June 2009, 45 pages

Medical researches and questionnaires declare that there are approximately 5 million diabetic patients in Turkey. Unfortunately majority of them don't realize that they are in danger of diabetes. It is thought difficult to visit a doctor and examine the results of their insulin measurement. I intend to make a benchmarking on data mining techniques for segmenting diabetes patients, which will help on examining the medical results of potential patients. I intend to use datas from İstanbul Diabetes Hospital. It's needed to benchmark of data mining techniques using socio-demographic data of diabetic patients, in order to reveal diabetes map of Turkey, to find association rules among the social-demographic data and to apply Adaptive Neuro Fuzzy Inference System (ANFIS), multinomial logistic regression (MLR), Bayesian network and rough set. Via benchmarking these used methods, it's seen that ANFIS is more effective than other methods using diabetes data.

Key words: expert system, Prolog Server Pages (PSP), diabetes diagnosis, WEKA, data mining, adaptive neuro fuzzy inference system (ANFIS), logistic regression

ÖZET

DİYABET HASTALARININ VERİLERİ KULLANILARAK VERİ MADENCİLİĞİ TEKNİKLERİNİN KARŞILAŞTIRILMASI

İnayet ADALI

Yüksek Lisans, Bilgisayar Mühendisliği

Bölümü Tez Yöneticisi: Doç. Dr. Adem

Karahoca

Haziran 2009, 45 sayfa

Tıbbi araştırmalar ve anketler, Türkiye'de 5 milyondan fazla diyabet (şeker) hastası bulunduğunu ortaya koymaktadır. Ancak bu hastalann büyük çoğunluğu maalesef diyabet tehlikesinde olduklannın farkında değildirler. Uzman bir doktoru ziyaret etmek, muayene olmak ve insülin tedavisinde dozajı ayarlamak için doktorla görüşmek hastalara zor gelmektedir. Hem potansiyel hastalann risk oranını belirlemek, hem de diyabetlilerin tedavileri boyunca yol gösterici bir uzman sistem geliştirmek için veri madenciliği tekniklerinin karşılaştinlması istendi. Bu nedenle, İstanbul Diabet Hastenesi verileri kullanıldı. Diyabet hastalannın sosyo-demografik verilerini kullanarak veri madenciliği tekniklerinin karşılaştinlması istendi. Bu amaçla, diyabet hastalannın sahip olduğu veriler ANFIS, multinominal lojistik regresyon, bayes ağı yardımı ve rough set kullanılarak kestirimler yapıldı. Son olarak kıyaslamalar yapıldı ve sonuc olarak, ANFIS' in daha etkili bir öğrenme ve kestirim aracı olduğu görüldü.

Anahtar Kelimeler: uzman sistem, Prolog Server Pages (PSP), diyabet tanı ve tedavisi, WEKA, veri madenciliği, ANFIS, benchmarking, lojistik regresyon.

TABLE OF CONTENTS

TABLES

FIGURES

ABBREVIATIONS

AI	Artificial Intelligence
ANFIS	Adaptive Neuro Fuzzy Inference System
BMI	Body Mass Index
HDL	High Density Lipoprotein
LDL	Low Density Lipoprotein
MLR	Multinomial Logistic Regression
PHP	Personal Home Pages
PSP	Prolog Server Pages
RDBMS	Rational Database Management System
ROC	Receiver Operating Characteristic

1. INTRODUCTION

Medical researches and questionnaires declare that there are approximately 5 million diabetic patients in Turkey. But unfortunately most of diabetic patients either don't visit physician regularly or don't know he is already diabetic. My starting point is to benchmark data mining techniques for segmenting these diabetic patients.

I would like to help diabetic patients or the people who suspect if they have diabetes risk by benchmarking these techniques. I think this segmentation will help patients and physicians during medical treatment for dosage planning stage as well.

The thesis contains 4 main sections. In the first section after introducing the problem diabetes will be explained briefly than the stages of benchmarking will be seen. In addition to this, the 3rd party solutions that work integrated to the diabetes data mining are shortly mentioned about.

The methodologies and materials of thesis are exhibited in section 2. Both functionality and usage are defined in the section supported with formulas and graphics.

The third section has experiments on Weka, Matlab and Rosetta. Bench markings are executed in the section in order to examine performances of data mining techniques and statistical methods.

In the final section, the results that we have reached, and the conclusions that the thesis has given us are shared.

1.1 Problem Definition

Life is difficult to diabetic patients. They must measure their glucose rate, inject insulin regularly, visit physician and examine the results. After the data measured, by using which technique we can make more time for detecting glucose level. After that Insulin dosage may be planned effectively.

Once data mining is finished, I saw that some of the techniques were not giving me the quite similar results, but by using some techniques I may have better results. Then why should people use worse techniques for segmenting diabetes patients.

Another problem that must be solved by segmentation diabetes data is the diabetic map of Turkey. It's needed to work on data mining with a knowledge based of diabetic patients. However the knowledge base must be pre-processed before applying data mining techniques. The diabetes segmentation should solve the pre-processing problem and apply data mining techniques such as classification and apply Neuro-Fuzzy Inference System like ANFIS (Adaptive Neuro-Fuzzy Inference System) (Polat & Gunes 2006).

1.2 What is Diabetes?

Diabetes is a disorder of metabolism—the way the body uses digested food for growth and energy. Most of the food people eat is broken down into glucose, the form of sugar in the blood. Glucose is the main source of fuel for the body.

After digestion, glucose passes into the bloodstream, where it is used by cells for growth and energy. For glucose to get into cells, insulin must be present. Insulin is a hormone produced by the pancreas, a large gland behind the stomach.

When people eat, the pancreas automatically produces the right amount of insulin to move glucose from blood into the cells. In people with diabetes, however, the pancreas either produces little or no insulin, or the cells do not respond appropriately to the insulin that is produced. Glucose builds up in the blood, overflows into the urine, and passes out of the body in the urine. Thus, the body loses its main source of fuel even though the blood contains large amounts of glucose.

1.2.1 What are the types of diabetes?

The three main types of diabetes are

- type 1 diabetes
- type 2 diabetes
- gestational diabetes

1.2.1.1 Type 1 Diabetes:

Type 1 diabetes is an autoimmune disease. An autoimmune disease results when the body's system for fighting infection—the immune system—turns against a part of the body. In diabetes, the immune system attacks and destroys the insulin-producing beta cells in the pancreas. The pancreas then produces little or no insulin. A person who has type 1 diabetes must take insulin daily to live.

At present, scientists do not know exactly what causes the body's immune system to attack the beta cells, but they believe that autoimmune, genetic, and environmental factors, possibly viruses, are involved. Type 1 diabetes accounts for about 5 to 10 percent of diagnosed diabetes in Turkey. It develops most often in children and young adults but can appear at any age.

Symptoms of type 1 diabetes usually develop over a short period, although beta cell destruction can begin years earlier. Symptoms may include increased thirst and urination, constant hunger, weight loss, blurred vision, and extreme fatigue. If not diagnosed and treated with insulin, a person with type 1 diabetes can lapse into a life-threatening diabetic coma, also known as diabetic ketoacidosis (http://www.hastane.com.tr-translation).

1.2.1.2 Type 2 Diabetes:

The most common form of diabetes is type 2 diabetes. About 90 to 95 percent of people with diabetes have type 2. This form of diabetes is most often associated with older age, obesity, family history of diabetes, previous history of gestational diabetes, physical inactivity, and certain ethnicities. About 80 percent of people with type 2 diabetes are overweight.

Type 2 diabetes is increasingly being diagnosed in children and adolescents, especially among African American, Mexican American, and Pacific Islander youth.

When type 2 diabetes is diagnosed, the pancreas is usually producing enough insulin, but for unknown reasons the body cannot use the insulin effectively, a condition called insulin resistance. After several years, insulin production decreases. The result is the same as for type 1 diabetes—glucose builds up in the blood and the body cannot make efficient use of its main source of fuel.

The symptoms of type 2 diabetes develop gradually. Their onset is not as sudden as in type 1 diabetes. Symptoms may include fatigue, frequent urination, increased thirst and hunger, weight loss, blurred vision, and slow healing of wounds or sores. Some people have no symptoms (http://www.hastane.com.tr-translation).

1.2.1.3 Gestational Diabetes

Some women develop gestational diabetes late in pregnancy. Although this form of diabetes usually disappears after the birth of the baby, women who have had gestational diabetes have a 40 to 60 percent chance of developing type 2 diabetes within 5 to 10 years. Maintaining a reasonable body weight and being physically active may help prevent development of type 2 diabetes.

About 3 to 8 percent of pregnant women in Turkey develop gestational diabetes. As with type 2 diabetes, gestational diabetes occurs more often in some ethnic groups and among women with a family history of diabetes. Gestational diabetes is caused by the hormones of pregnancy or a shortage of insulin. Women with gestational diabetes may not experience any symptoms (Swapnil N. Rajpathak, 2008).

1.2.2 Diabetes in Youth

The SEARCH for Diabetes in Youth multicenter study, funded by the Centers for Disease Control and Prevention and the National Institutes of Health, has determined that (Polat & Gunes, July 2009);

- based on data from 2002 to 2003, a total of 8,000 youth in Turkey were newly diagnosed with type 1 diabetes each year. In addition, about 1,700 youth were newly diagnosed with type 2 diabetes each year.
- non-Hispanic white youth had the highest rate of new cases of type 1 diabetes.
- type 2 diabetes was rarely diagnosed among youth younger than 10 years of age.

1.2.3 Other Types of Diabetes

A number of other types of diabetes exist. A person may exhibit characteristics of more than one type. For example, in latent autoimmune diabetes in adults (LADA), also called type 1.5 diabetes or double diabetes, people show signs of both type 1 and type 2 diabetes(T.Zhang - S. Pan, 2009).

Other types of diabetes include those caused by

- genetic defects of the beta cell—the part of the pancreas that makes insulin—such as maturity-onset diabetes of the young (MODY) or neonatal diabetes mellitus (NDM)
- genetic defects in insulin action, resulting in the body's inability to control blood glucose levels, as seen in leprechaunism and the Rabson-Mendenhall syndrome
- diseases of the pancreas or conditions that damage the pancreas, such as pancreatitis and cystic fibrosis
- excess amounts of certain hormones resulting from some medical conditions—such as cortisol in Cushing's syndrome—that work against the action of insulin

- medications that reduce insulin action, such as glucocorticoids, or chemicals that destroy beta cells
- infections, such as congenital rubella and cytomegalovirus
- rare immune-mediated disorders, such as stiff-man syndrome, an autoimmune disease of the central nervous system
- genetic syndromes associated with diabetes, such as Down syndrome and Prader-Willi syndrome

1.2.4 How is diabetes diagnosed?

The fasting blood glucose test is the preferred test for diagnosing diabetes in children and nonpregnant adults. The test is most reliable when done in the morning. However, a diagnosis of diabetes can be made based on any of the following test results, confirmed by retesting on a different day:

- A blood glucose level of 126 milli grams per deciliter (mg/dL) or higher after an 8-hour fast. This test is called the fasting blood glucose test.
- A blood glucose level of 200 mg/dL or higher 2 hours after drinking a beverage containing 75 grams of glucose dissolved in water. This test is called the oral glucose tolerance test (OGTT).
- A random—taken at any time of day—blood glucose level of 200 mg/dL or higher, along with the presence of diabetes symptoms.

Gestational diabetes is diagnosed based on blood glucose levels measured during the OGTT. Glucose levels are normally lower during pregnancy, so the cutoff levels for diagnosis of diabetes in pregnancy are lower. Blood glucose levels are measured before a woman drinks a beverage containing glucose. Then levels are checked 1, 2, and 3 hours afterward. If a woman has two blood glucose levels meeting or exceeding any of the following numbers, she has gestational diabetes: a fasting blood glucose level of 95 mg/dL, a 1-hour level of 180 mg/dL, a 2-hour level of 155 mg/dL, or a 3-hour level of 140 mg/dL. (Notes from the doctor of İstanbul Diabet Hospital. Dr. Huriye Alasya)

1.3 Related Works

Any researcher could find Data mining articles on diabetes, when he searches about data mining on Internet or scientific libraries. Some related works about diabetes and data mining are found in the literature research, and mentioned about them in the following section. When we investigated them, we saw they usually do 2 different methods also none of them make benchmarking between the methods they used so none of the works fits the problems defined in the Section 1.1.

There is non-ignorable number of people who suffer from diabetes. And, most of them are not aware that they have diabetes. Predictability of diabetes plays an important role for the patient's early treatment process. However, the correct prediction percentage of current algorithms is low. A new approach, called Homogeneity-Based Algorithm (HBA), is applied together with traditional classification approaches to enhance the accuracy of diabetes prediction. In contrast to traditional approaches, this new approach proposes a new method to optimally control the over fitting and overgeneralization behaviors of classification on this dataset. Some computational results seem to indicate that the proposed approach significantly performs better than current approaches. Over fitting and overgeneralization behaviors of current classification algorithms cause low accuracies in processing the dataset. The obtained results in relation with HBA appear to be very important both for accurately predicting diabetes and also for the data mining community, in general. (H.Nguyen and A.Pham 2007)

In this study, they get better result when compared to MLR. We will try to get better results from diabet data by using different methods and benchmarked the methods used.

Pima Indians living in near Phoenix, Arizona, who are usually susceptible to diabetes, were tested in accordance with the criteria of World Health Organization. An analysis of findings of project conducted is expected to be done based on a number of sources which are available publicly including at http://ics.uci.edu/mlearn/MLRepository.html.

The analysis may include following:

1. An analysis based on 798 females with eight continuous variables measured for each female.
2. An exploratory statistical analysis performed on the sample dataset.
3. Use of Ripley's training dataset to construct a logistic model for prediction the variable "type.
4. A formation of classification tree grown for Pima Indians diabetes data based on a training set of size 200.
5. Control of reduction of dimension of the problem by applying the principal components analysis to dataset.
6. Summary of the problem and results of the analysis.

Pima Indians network is a great dataset for segmenting, they used exploratory statistical analysis on this dataset after the analysis they get pretty good results on the dataset. When compared to Pima India dataset our dataset is similar it has less patients but more variables we will try to have better results from Pima India dataset if we can.

Diabetes is a disease in which the body has a difficulty in regulating insulin. This may lead to many diseases such as kidney disease, blindness, nerve damage, blood vessel damage besides contributing to heart disease. The principal component analysis (PCA) and adaptive neuro-fuzzy inference system (ANFIS) have been used as a method to diagnose the diabetes disease. The dataset for analysis was obtained from the sources held by the National Institute of Diabetes and Digestive and Kidney Disease, where all patients are Pima-Indian women at least 21 years old and living in Phoenix, Arizona, USA. To do this, an expert system which works in two stages has been applied. In first stage, the dimension of diabetes database was reduced from 8 to 4 features using PCA. And, in the second, an adaptive neuro-fuzzy inference system (ANFIS) classifier was used to diagnosis of diabetes disease. As a result, it has been found that the use of PCA and ANFIS in combination give a diagnostic and classification accuracy of 89.47%, making it very promising with regard to other classification applications in the literature for this problem. (K. Polat, S. Güneş-2008)

In this study ANFIS used on diabetic patients. Our approach is to see differences between ANFIS and other methods. We will see if there are any better result can be found.

2. MATERIALS & METHODS

2.1 Data Set Preparation for diabetics

The dataset that was used in the thesis consist of 24 variables of 470 subjects 277 male and 193 female who were interviewed in a **İstanbul Diabet Hospital** and also 16 different treatment type. Most of subjects are known as diabetic and all of them are under diabetes treatment. In our work it's tried to find any relation between diabetes risk and height, weight, gender, HbA%, uric glucose, blood pressure, keton, protein, micros, Mic. Alb, polifaji, poliuri, polidipsi, total cholesterol, side effects such as xerostomia and a ratio that is called frame. The waist/hip ratio (Frame) may be a predictor in diabetes.

Having searched the literature, it's seen that works were dealing with binary results (1=healthy, 0=diabetic). In this work it's desired to make a step beyond, and worked on fuzzy dependent variable.

The dependent variable in the dataset is Glucose rate and independent variables are:

i. Height
ii. Weight
iii. HbA%
iv. Uric glucose
v. Blood Pressure
vi. Keton
vii. Protein
viii. Mikros
ix. Mic.Alb
x. Polifaji
xi. Poliuri
xii. Polidipsi
xiii. Total cholesterol
xiv. Gender

xv. Frame (Waist/Hip ratio)

xvi. Side Effects(7 Different)

xvii. Medicine(16 Types)

Some variables of the dataset had already fuzzy values, such as frame. However some of them didn’t have fuzzy values such as polidipsi, glucose. Some variables of the dataset used in the thesis are shown in Table 2.1.

Table 2.1 : Some variables and descriptions of dataset

Name	Description
Glucose	Diabetes indicator, glucose level in blood (Dependent variable)
Uric glucose	Glucose value in patients’ urine
Blood Pressure	Upper Blood Pressure and Lower blood pressure of the patients
Protein	Protein value in patients’ urine
Side Effects	Side effects of Diabetes seen on patients like xerostomia
HbA%	HbA value of diabetic patients’ blood
Gender	Gender of diabetic patients
Frame	Waist / hip ratio of diabetic patients.
Total Cholesterol	Total Cholesterol values of diabetic patients

The main purpose of preprocessing data was to make fuzzy the variable in order to use them in ANFIS as a fuzzy inference system. Some variables and their fuzzy values are shown in Table 2.2.

Table 2.2 : Discretization N=470

Uric Glucose	**Gender**	**Frame**	**Cholesterol**	**Side Effects**	**Glucose (Out)**
1- above limit N=399	1 –Male N=277	1-Small N=76	1- (<100) N=102	1- Seen N=404	1- (<60) N=47
0- under limit N=71	2-Female N=193	2-Medium N=213	2- (101-200) N=198	0- Not seen N=66	2- (60-89) N=55
		3-Large N=181	3- (over 200) N=170		3- (90-120) N=35
					4- (121-300) N=270
					5- (over 300) N=63

Since the purpose is same the other estimation method, and it's to define a relationship between dependent variable and independent variables by using minimum variables; then it's needed to decrease the number of inputs to the least meaningful number.

The function exhsrch in MATLAB performs an exhaustive search within the available inputs to select the set of inputs that most influence the diabetes diagnosis. The first parameter to the function specifies the number of input combinations to be tried during the search.

And I also user evalfis in MATLAB to perform fuzzy inference calculations.

Rosetta data set

Table 2.3: Table of the attributes in Rosetta.

Attribute	Descpirtion	Range	Attribute	Descpirtion	Range
Gnd_code	Gender	162	Krn_code	Keratin	99
Hba_Code	HbA%	314	Prt_code	Protein	109
Chl_Code	Cholestrole	80	Ten_code	High tension	377
Medici0_code	Gluchopage	22	Plg_code	Polifaji	189
Glu_code	Glucose Lvl	377	Plu_code	Poliuri	258
Medici1_code	Insulin Lan.	33	Ex_code	Exermia	105
Medici2_code	Glifix	88	Mik_code	Mik.Albumin	407

2.2 Data Mining Methods

A variety of data mining methods exists. Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information –information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. In our thesis, some data mining methods on social-demographic data of the users were applied. Correction and actuality of the data is very important for data mining for diabetes. We have listed a few methods we have encountered in Table 2.4

Table 2.4: List of some Data Mining methods.

Method	Description
Rough Sets	Founded on logic, statistics and algebra
ANFIS	Founded on logic and statistics
Bayesian Network	Founded on statistics
MLR	Founded on statistics and algebra

Selection of Methods

We have chosen to study the methods Rough Sets, MLR, Bayesian and ANFIS. We chose them because we wanted to study methods which have been used on the diabetic patients data in the literature.

2.2.1 Bayesian Networks

Bayesian, or belief networks, is a method to represent knowledge using directed acyclic graphs (DAG's). The vertices are events which describe the state of some part of the desired universe in some time interval. The edges represent relations between events, and the absence of edges represents independence between events.

A variable takes on values that correspond to events. Variables may be discrete or continuous. A database is denoted D in the following, the set of all variables U, and a Bayesian network-structure is denoted B. If D consists of all variables in U, then D is complete. Two structures are said to be equivalent if they have the same set of probability distributions.

The existing knowledge of an expert or set of experts is encoded into a Bayesian network, then a database is used to update the knowledge, and thus creates one or several new networks.

Bayesian network is illustrated. A disease is dependent on age, work and environment. A symptom is again dependent on a disease.

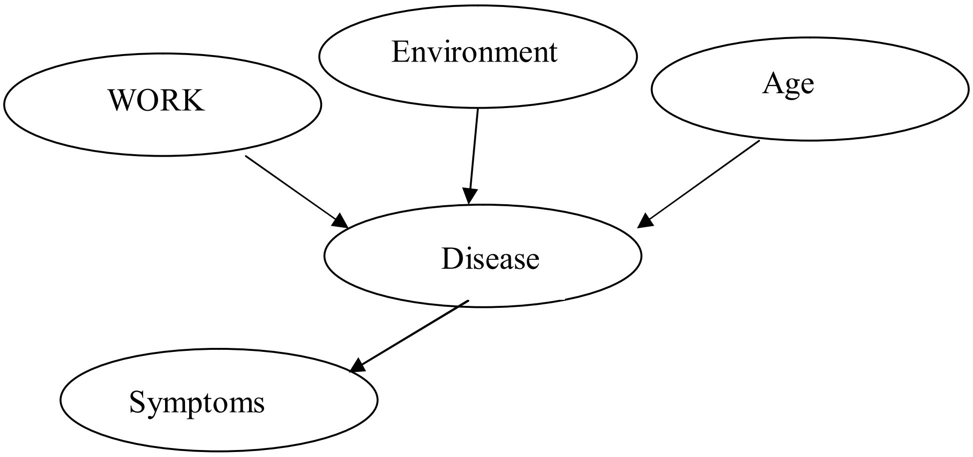


Figure 2.1: A Bayesian Network example.

2.2.1.1 Theoretical Overview

In probability theory, P(A) denotes the probability that event A will occur. Belief measures in Bayesian formalism obeys the three basic axioms of probability theory:

$$0 \le P(A) \le 1 \tag{2.1}$$

$$P(Sure\ proposition) = 1 \tag{2.2}$$

If A and B are mutually exclusive, then:

$$P(A\ or\ B) = P(A) + P(B) \tag{2.3}$$

The Bayes rule is the basis for Bayesian Networks:

$$P(H|e) = \frac{P(e|H)P(H)}{P(e)} \tag{2.4}$$

Where $P(e|H)$ is called the posterior probability and P(H) is called the the prior probability (N Friedman, D Geiger, M Goldszmidt ,1997).

Probability can be seen as a measure of belief in how probable outcomes are, given an event (subjective interpretation). It can also be interpreted as a frequency; how often it has happened in the past (objective interpretation).

2.2.1.2 Applied in Data Mining

When Bayesian Networks are applied in Data Mining, the user may want to apply techniques to learn the structure of a Bayesian network with one or more variables, or to reason on a structure that is already known.

2.2.1.2.1 One Variable

With only one variable, the task is to compute a distribution for the variable given any database. For instance, if we want to compute the probability distribution for a coin that is flipped, the distribution for the variable θ would be:

$$p\left(\theta \middle| h\ heads, t\ tails, \xi\right) = c\theta^{h}\left(1-\theta\right)^{t} p\left(\theta \middle| \xi\right) \tag{2.5}$$

Where ξ denote the background knowledge of the model, h denotes the number of heads and t the number of tails. This result is only dependent on the data in the database. In general, for a variable that can have more outcomes, the distribution for the set of physical probabilities of outcomes, $\Theta_x = \theta_{x=1},...,\theta_{x=r}$ will be:

$$p\left(\Theta_x \middle|\ D, \xi\right) = c\prod_{k=1}^{r} \theta_{x=k}^{N_k^{'}+N_k-1} \tag{2.6}$$

Where N_k is the number of times x = k in D, and c is a normalization constant (FV Jensen, TD Nielsen – 2001).

2.2.1.2.2 Known Structure

If the structure of the network, B_S^h, is known in advance, the aim of the analysis is to compute the probabilities on the vertices from the database D. If the database is complete, and the variables are discretized into r different outcomes, the probabilities can be computed according to the following formula:

$$p\left(C_{m+1}\middle|D,B_S^h,\xi\right)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{N_{ijk}^{'}+N_{ijk}}{N_{ij}^{'}+N_{ij}} \tag{2.7}$$

2.2.1.2.3 Learning Structure and Variables

To learn a structure, we want to generate candidate Bayesian Networks, and choose the structure that is best according to the database we have.

The probability distribution in equation gives the probability distribution for the next observation after looking in the database:

$$p\left(C_{m+1}\middle|D,B_S^h,\xi\right)=\sum_{B_S^h}p\left(C_{m+1}\middle|D,B_S^h,\xi\right)p\left(B_S^h\middle|D,\xi\right) \tag{2.8}$$

To find a Bayesian Network B for a database D, we can use metrics together with a search algorithm. The metrics approximate equation , and some of the metrics are:

- Bayes Factor (BF)
- Bayesian Dirichlet (BD)
- Maximum a Posteriori (MAP)
- A Information Criterion (AIC)
- Bayes Information Criterion (BIC)
- Minimum Description Length (MDL)

Some search methods that exist are:

- Local Search - a heuristic search algorithm on the graph.
- Iterated Hill-Climb - uses Local Search, but avoids getting stuck in a local maxima.

If data are missing in the database, it can either be filled in using the posterior distribution. Or we can use Gibbs sampling, which approximates the expectation of any function with respect to it's probability distribution. Another alternative is to use the expectation-maximization (EM)-algorithm which can be used to compute a score from the BIC-metric. (RE Neapolitan – 2003)

2.2.1.3 Properties

Noise: Missing data can be filled in using Gibbs sampling or the posterior distribution. The method is relatively resistant to noise.

Consistency: Bayesian Networks are able to reason with uncertainty. Uncertainty in the data will be included in the model as probabilities of different outcomes.

Prior knowledge: In order to make the Bayesian network structure, it has to be learned from the database. It can also exist as prior knowledge.

Output: The output from Bayesian Network algorithms is a graph, which is easy to interpret for humans.

Complexity: There exist classifiers that perform in linear time, but further iterations give better results.

Retractability: The method is statistically founded, and is always possible to examine why the bayesian method produced certain results.

2.2.2 Rosetta Rough Sets

The Rough Sets method was designed as a mathematical tool to deal with uncertainty in AI applications. Rough sets have in the last years also proven useful in Data Mining. The method was first introduced by Zdzislaw Pawlak in 1982, the work have been continued by others. This section is a short introduction to the theory of Rough Sets.

2.2.2.1 Theoretical Overview

The starting point of Rough Sets theory is an information system, an ordered pair $\mathcal{A}=(U,A)$. U is a non empty finite set called universe, A is a non empty finite set of attributes. The elements $\{x_1,...,x_n\}$ of the universe are called objects, and for each attribute $a \in A$, a(x) denotes the value of attribute a for object $x \in U$. It is often useful to illustrate an information system in a table, called an information system table. One of the most important concepts in Rough Sets is indiscernibility, denoted IND(). For an information system $\mathcal{A}=(U,A)$, an indiscernability relation is:

$$IND(B)=\{(x,y)\in U^2 : a(x)=a(y) \textit{ for every } a \in B\}, \qquad (2.10)$$

Where $B \subseteq A$. The interpretation of $x_1 IND(B) x_2$ is that the objects x_1 and x_2 can not be separated by using the attributes in the set B. The indiscernible objects can be classified in equivalence classes:

$$[x]_B=\{y\in U:(x,y)\in IND(B)\}, \qquad (2.11)$$

again $B \subseteq A$. $[x]_B$ denotes an equivalence class based in the attributes in set B. The set U/IND(B) is the set of all equivalence classes in the relation IND(B). Intuitively the objects of an equivalence class are indiscernible from all other objects in the class, with respect to the attributes in the set B. In most cases we therefore consider the equivalence classes and not the individual objects. The discernibility matrix of an information system A, with n attributes, is a symmetric n x n matrix with c_{ij} given by:

$$c_{ij} = \left\{ a \in A \middle| a(x_i \neq a(x_j \right\} \; for \; i,j = 1,\ldots,n \tag{2.12}$$

hence, $c_{i,j}$ contains the attributes upon witch classes x_i and x_j differs (Wieczorkowska, 1999).

Some of the objects attributes may be less important than others when doing the classification. An attribute a is said to be dispersible or superfluous in $B \subseteq A$ if IND(B)=IND(B-{a}) Otherwise the attribute is indispensable in B. A minimal set of all the attributes that preserves the partitioning of the universe is called a reduct, RED. Thus a reduct is a set of attributes $B \subseteq A$ such that all attributes $a \in A - B$ are dispensible and IND(B)=IND(A). There are also object-related reducts that gives the set of attributes needed to separate one particular object/class from all the other. The theory is illustrated in the example below.

Table 2.5. An information system

object	a	b	c
1	1	2	3
2	1	3	2
3	1	2	3
4	2	1	3
5	2	1	3
6	1	2	3
7	1	3	3

Table 2.6. Discernibility matrix

	E_1	E_2	E_3	E_4
E_1	∅	ab	bc	b
E_1	ab	∅	abc	ab
E_1	bc	abc	∅	c
E_1	b	ab	c	∅

Example 1 Consider the small information System $\mathcal{A} = (U, A)$, illustrated in Tab??. We see that some of the objects have the same value for all of the attributes, for instance objects 1 and 6, and objects 4 and 5. These objects are indiscernible by attributes a through c, and theses objects are of the same equivalence class. Table 2.5 illustrate the equivalence classes, the attributes and the number of objects from the universe that are indiscernible by attributes a through c. The discernibility matrix for $\mathcal{A}$ is shown in Table 2.6, and the reduct for $\mathcal{A}$ is, $RED(A) = \{b, c\}$ (Z Pawlak , 2000).

Table 2.7. Equivalence classes

Class	a	b	c	
E_1	1	2	3	50
E_2	2	1	3	11
E_3	1	3	2	10
E_4	1	3	3	19

The next to define is the lower and upper approximation, Rough definability of sets and the Rough membership function. The lower and upper approximation are used to classify a set of objects, X, from the universe that may belong to different equivalence classes. Let $B \subseteq A$ and $X \subseteq U$, the B-lower, $\underline{B}X$ and B-upper, $\overline{B}X$ approximation are:

$$\underline{B}X = \{x \in U : [x]_B \subseteq X\} \tag{2.13}$$

$$BX = \{x \in U : [x]_B \cap X \neq \theta\} \tag{2.14}$$

Hence this is a way to approximate X using only the attributes in the set $B \subseteq A$. The set $\overline{B}X - \underline{B}X$ is called the boundary of X. Intuitively, the objects in $\underline{B}X$ are the objects that are certain to belong to X, while the objects in $\overline{B}X$ are possible members of X, as illustrated in Fig 2.2

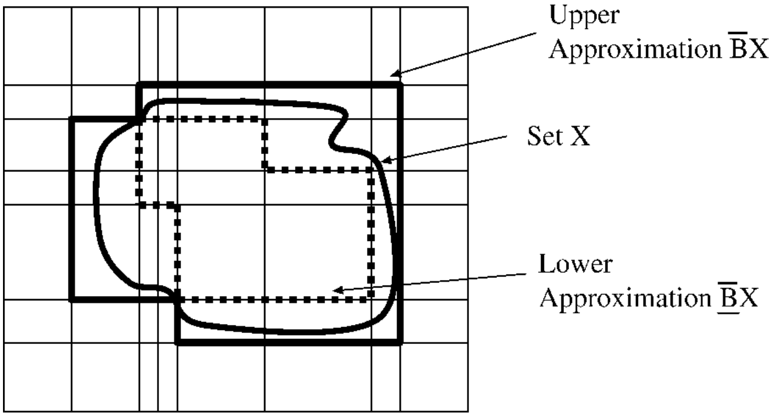


Figure 2.2: The B-upper and B-lower approximations of X.

If $\underline{B}X \neq \phi$ and $\overline{B}X \neq U$ then X is said to be Roughly B-definable.

If $\underline{B}X = \phi$ and $\overline{B}X \neq U$ then X is said to be Internally B-undefinable.

If $\underline{B}X \neq \phi$ and $\overline{B}X = U$ then X is said to be Externally B-undefinable.

If $\underline{B}X = \phi$ and $\overline{B}X = U$ then X is said to be Totally B-undefinable

The rough membership function $\mu_B(x, X)$, where $x \subseteq X$, is given by:

$$\mu_B(x, X) = \mu([x]_B, X) = \frac{|[x]_B \cap X|}{|[x]_B|}, \; hence 0 \leq \mu_B(x, X) \leq 1 \qquad (2.15)$$

The rough membership function denotes the frequency of $[x]_B$ in X, or the probability of X being a member of $[x]_B$ (L Polkowski, A Skowron – 1998).

A natural extension of the information system is the decision system. A decision system is an information system with an extra set of attributes, called decision attributes, hence $\mathcal{A} = (U, A \cup D)$. In most cases the set D has only one element, $D = \{d\}$, and we will use this simplification in the rest of this section. The decision attribute, d, reflects a decision made by an expert. The value of d, V_D is often a set of integers {1,...,r(d)}, where r(d) denotes the range of d. It is also possible to represent non numerical decisions, but then it is much simpler to encode the decision into integers, e.g. false and true can be encoded 0 and 1 respectively. The partitioning $U/IND(\{d\}) = \{X_1,...,X_{r(d)}\}$ of the universe is denoted a decision class, and is a classification of the objects in $\mathcal{A}$ determined by the decision attribute d. A decision system is said to be **consistent** if for all (Z Pawlak , 2002).

$$E_i \subseteq U/IND(A),\ there\ excists\ an\ X_j \subseteq U/IND(d)\ such\ that\ E_i \subseteq X_j \qquad (2.16)$$

Otherwise, the decision system is inconsistent. An inconsistent system is a system where two or more of the objects have the same value for the set of attributes A, hence belonging to the same equivalence class, but the decision attribute d differs. A Decision table is a table of the indiscernability classes, the attributes and the decision attribute. The example below illustrates this.

Table 2.8. A decision system

Class	a	b	c	d	
E_1	1	2	2	1	50
E_2	2	3	2	2	30
E_3	1	2	3	2	10
E_{4a}	2	1	3	2	9
E_{4b}	2	1	3	3	1

Example 2 Consider the decision system $\mathcal{A}=(U, A, \cup\{d\})$ as an extension of the information system in Table 2.6 , illustrate in Table 2.7. We see that there is an inconsistency in class E_4, where 10 of the objects have decision 2 and one object has decision 1. (L Polkowski – 2002)

2.2.2.2 Applied in Data Mining

When using rough sets as a algorithm for data mining, we generate decision rules that map the value of an objects attribute to a decision value. The rules are generated from a test set. If $\mathcal{A}=(U, A\cup\{d\})$ is a decision system, and a descriptor a=v, where $a \in A\cup d,\ v \in V_d$, then a decision rule is an expression generated from descriptor:

$$\left(a_{i_1}=v_{i_1}\right)\cap\left(a_{i_2}=v_{i_2}\right)\cap\cdots\cap\left(a_{i_k}=v_{i_k}\right)\Rightarrow d=v \tag{2.17}$$

Where $a_{i_1}\cdots a_{i_k}\in A$, $v_{i_1}\cdots v_{i_k}$ are values of the attributes in A, d is the decision attribute and v is the value of d. There are two types of decision rules, definite and default decision rules. Definite decision rules are rules that map the attribute values of an object into exactly one decision class. A deterministic decision system can be completely expressed by a set of definite rules, but an indeterministic decision system can not.

Example 3 We may get these definite rules from the decision system in Table 2.7:

- $E_1 : a_1 b_2 c_3 \rightarrow d_1$
- $E_2 : a_2 b_3 c_2 \rightarrow d_2$
- $E_3 : a_2 b_2 c2 \rightarrow d_2$

Where a_1 denotes that $a=1$ and $b_2$denotes that $b=2$, etc. but for class E_4 we can not make rules that maps into exactly one decision class.

Given a decision system $\mathcal{A}=(U,(C,D))$ and a threshold value $\mu_{tr},\ 0\le\mu_{tr}\le 1$. A default rule for $\mathcal{A}$ with respect to the threshold is any decision rule where:

$$\mu(E,X)=\frac{|E\cap X|}{|E|}\ge\mu_{tr} \tag{2.18}$$

Depending on the threshold value, it will be generated zero, one or more mutually inconsistent rules. A threshold of 1 gives only the set of default rules that are exactly identical to the set of definite rules. A threshold value greater than 0.5 maps the value of an objects attributes of an inconsistent equivalence class into only one decision class, where as a threshold value below 0.5 may map into several decision classes (RW Swiniarski, A Skowron , 2003).

Example 4 Assume thresholds for decision system from Table 2.8:

- $\mu_{tr} = 1\, for\ E_4\ we\ get\ no\ rules.$
- $\mu_{tr} = 0.6\, for\ E_4\ we\ get\ one\ rule: E_4 : a_2b_1c3 \rightarrow d_2$
- $\mu_{tr} = 0.1\, for\ E_4\ we\ get\ two\ rules:$
  - $I : E_4 : a_2b_1c3 \rightarrow d_2$
  - $II : E_4 : a_2b_1c3 \rightarrow d_3$

Definite and default rules generated directly from the training set, tends to be very specialized. This can be a problem when the rules are applied to unknown data, because many of the objects may fall outside, i.e. not match, of the specialized rules. If we make the rules more general, chances are that a greater number of the objects can be matched by one or more of the rules. One way to do this is to remove some of the attributes in the decision system, and by this glue together equivalence classes. This will certainly increase the inconsistency in the decision system, but by assuming that the attributes removed only discerns a few rare objects from a much larger decision class, this may not be a problem in most cases.

Example 5 Assume we want to classify a new object, with attributes: $a_2b_1c_2$ and d_2. With the specialized rules from example 1.3 we can not make a match. If we glue together classes E_2 and E_3 by removing attribute b we get these new rules:

- $E_1 : a_1c_3 \rightarrow d_1$
- $E_2, E_3 : a_2c_2 \rightarrow d_2$

We can now classify the object.

2.2.2.3 Properties

The subsections below compare the Rough Set method to the criterias outlined in formulas section. This comparison will give us a framework that can be used to compare the various data mining techniques.

Noise: The rough set method is capable of handling most types of noise. If the input data is missing an attribute that is dispencible then the classification process is not affected. If the input data are missing important attributes, then the Rough Sets can not classify the object and the object must be thrown away. There are algorithms implemented that can 'guess' the value of a missing attribute, by doing statistical insertion among others, and thereby keeping the object.

Consistency: As described before, the default rules in Rough Sets can handle inconsistency in the information system quite nicely. In fact, as pointed out, it may be a advantage to introduce inconsistency, by removing attributes, to make the decision rules more general.

Prior Knowledge: Rough set method does not use prior knowledge when generating default rules. As mentioned before the default rules generated by Rough Set algorithms are generated only from the objects in the input data.

Output: The output from the Rough Set algorithms are rules, preferably default rules. These rules can then be uses to predict decisions based on the input attributes. By making variations in the threshold value, μ_{tr} one can tune the number of default rules to get the most optional result. Rules are easy for humans to interpret and use.

Complexity: The complexity of partitioning is O(nlog(n)), a sort algorithm. Computing the reducts is NP-complete, but there are approximation algorithms (genetic, Johnson, heuristical search) that have proven very efficient.

Retractability: It is easy for the user to find out how the results were generated. This is because Rough Sets method can display the indiscernability classes it created and the distribution of the objects in these classes. The user can from this confirm the rules generated, and argument, in basis of the classes, why the rules are correct.

To experiment with the methods chosen, we selected the tools Rosetta, which is a Rough Set analysis tool and AutoClass, which is a Bayesian Networks classifier. The experiments were done on the Diabet Hospital Archives.

The Diabet Hospital Archives: The Diabet Hospital Archive is a large one, which has been developing for more than ten years. It contains data from all over Turkey

Most attributes in the archive are composed differently which for instance describe types of failures. Important information are usually provided. There are several missing attribute values in the archives for some patients.

2.2.2.4 The Tools

2.2.2.4.1 AutoClass - A Bayesian Networks Classifier

AutoClass is a tool for unsupervised classification using Bayesian methods to determine the optimal classes. AutoClass was developed at the NASA Ames Research Center.

In AutoClass, data must be represented in a data vector, with discrete nominal values or real scalar values. Each information vector is called a case, but we will still use the term object for each vector. The vectors can be stored in text- or binary files.

A class is a particular set of attribute (or parameter) values and their associated model. A model is either Single multinomial, Single normal or Multi normal. The first two models assume that attributes are conditionally independent, given a class; the probability that a class would have a particular value of any attribute depend only on the class and is independent of all other attribute values. The Multi normal model expresses mutual dependencies within the class.

Class membership is expressed probabilistically. The sum of probabilities for an object must always sum to one.

2.2.2.4.2 Rosetta - A Rough Set Toolkit for Analysis of Data

Rosetta is a toolkit for pattern recognition and Data Mining, within the framework of Rough Sets. It is developed as an cooperative effort between the Knowledge System Group at NTNU, Norway and the Logic Group at the University of Warsaw, Poland.

Rosetta runs on every platform, and consists of a kernel and a Graphical User Interface (GUI). The user interacts with the kernel through the GUI, but the kernel is an independent unit. The GUI in Rosetta relies on human interaction and human editing. Data are imported as a decision table from a text file.

The first line contains the attribute names, the second contains the attribute types; integers, real numbers (float) or strings. Then the objects are listed successively. Missing attributes are denoted Undefined.

Rosetta organizes the data in a project and displays relations between the various project items in project trees. Through these trees the user can monitor and manually edit the steps in the process.

Rosetta has also a possibility for testing the rules that have been generated. The rules are applied on a test set with known decision values. The result from the test is a confusion matrix, which is a matrix where the predicted decisions values are compared to the actual decision values. The steps done on the data are recorded together with the data, therefore reproduction of the results are easy to do.

2.2.3 Adaptive Neuro Fuzzy Inference System (ANFIS)

As a neural-fuzzy system, ANFIS is a combination of neural networks and fuzzy systems in such a way that neural networks or neural networks algorithms are used to determine parameters of fuzzy system. This means that the main intention of neural-fuzzy approach is to create or improve a fuzzy system automatically by means of neural network methods. Adaptive neuro fuzzy inference system basically has 5 layer architectures and each of the function is explained in detail below (Sojda 2007; Seising 2006):

Layer 1 Every mode in this layer is an adaptive node with a node function where x (or y) is the input to node I and $A_i\left(\text{or } B_{i-2}\right)$ is a linguistic label and O_i^1 is the membership grade of fuzzy set $\mathrm{A}\left(=A_1, A_2, B_1 \text{ or } B_2\right)$ and it specifies the degree to which the given input x (or y) satisfies the quantifier A. The membership function for A can be parameterized membership function as given in equation 1 or normally known as Bell function and $\{a_i, b_i, c_i\}$ is the parameter set

$$\mu_{Ai}(x) = \frac{1}{1+\left\{\left((x-ci)/ai\right)^2\right\}^{b_i}}, \qquad (2.19)$$

$$O_i^1 = \mu_{Ai}(x),\ \mathrm{i} = 1{,}2,$$
$$O_i^1 = \mu_{Bi-2}(y),\ \mathrm{i} = 3{,}4, \qquad (2.20)$$

Layer 2 Every node in this layer is a fixed node labeled M, whose output is the product of all the incoming signals each node output represents the firing strength of a rule.

$$O_i^2 = \omega_i = \mu_{Ai}(x)_{Bi}(y),\quad \mathrm{i} = 1{,}2, \qquad (2.21)$$

Layer 3 Every node in this layer is a fixed node labeled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths. Outputs of this layer are called normalized

$$O_i^3 = \varpi_i = \frac{\omega_i}{\omega_1 + \omega_2}, \quad \text{i} = 1{,}2, \tag{2.22}$$

Layer 4 Every node I in this layer is an adaptive node with a node function. Where wi is a normalized firing strength from layer 3 and $\{pi,\ qi,\ ri\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

$$O_i^4 = \varpi_i f_i = \varpi_i \left(p_i x + q_i y + r_i \right), \quad \text{i} = 1{,}2, \tag{2.23}$$

Layer 5 The single node in this layer is a fixed node labeled ∑, which computes the overall output as the summation of all incoming signals. Overall output:

$$O_i^4 = \sum_{i=1}^{2} \varpi_i f_i = \frac{\sum_{i=1}^{2} \omega_i f_i}{\omega_1 + \omega_2}, \tag{2.24}$$

For simplicity, we assume that the fuzzy inference system under consideration has two input x and y and one output z. For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is the following:

Rule 1: If x is A_1 and y is B_1, the $f_1 = p_1\ x + q_1\ y + r_1$,

Rule 2: If x is A_2 and y is B_2, the $f_2 = p_2\ x + q_2\ y + r_2$.

The corresponding equivalent ANFIS architecture is as shown in Figure 4.9, where nodes of the same layer have similar functions. ANFIS has hybrid learning capability which compromised of back propagation and least square method.

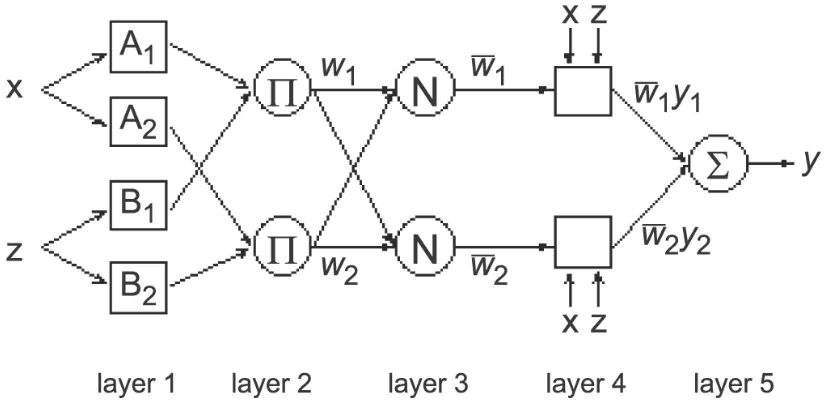


Figure 2.3 ANFIS architecture (Jang, 1993).

2.2.4 Multinomial Logistic Regression

Multinomial logistic regression is used when the dependent variable in question is nominal and consists of more than two categories. In our work, multinomial logistic regression would be appropriate, because we are trying to determine how factors to predict glucose level causing diabetes disease.

The multinomial logistic model assumes that data are case specific; that is, each independent variable has a single value for each case. The multinomial logistic model also assumes that the dependent variable cannot be perfectly predicted from the independent variables for any case.

$$\Pr\left(y_i = j\right) = \frac{\exp\left(X_i\beta_j\right)}{1+\sum_j^J \exp\left(X_i\beta_j\right)} \tag{2.25}$$

$$\Pr\left(y_i = 0\right) = \frac{1}{1+\sum_j^J \exp\left(X_i\beta_j\right)}, \tag{2.26}$$

According to multinomial logistic regression model, which is defined in (7) and (8), the ith is individual, y_i is the observed outcome and X_i is a vector off explanatory variables. The unknown parameters β_j are typically estimated by maximum likelihood (Keles & Keles 2006).

Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

In our thesis, some data mining methods on social-demographic data of the users were applied. Correction and actuality of the data is very important for data mining for diabetes.

2.2.5 ANFIS vs. MLR

Artificial Neural Networks (ANNs) and Fuzzy Logic (FL) have been increasingly in use in many engineering fields since their introduction as mathematical aids by McCulloch and Pitts, 1943, and Zadeh, 1965, respectively. Being branches of Artificial Intelligence (AI), both emulate the human way of using past experiences, adapting itself accordingly and generalizing. While the former have the capability of learning by means of parallel connected units, called neurons, which process inputs in accordance with their adaptable weights usually in a recursive manner for approximation; the latter can handle imperfect information through linguistic variables, which are arguments of their corresponding membership functions.

Although the fundamentals of ANNs and FL go back as early as 1940s and 1960s, respectively, significant advancements in applications took place around 1980s. After the introduction of back-propagation algorithm for training multi-layer networks by Rumelhart and McClelland, 1986, ANNs has found many applications in numerous inter-disciplinary areas (Patterson 1994; Rumelhart & McCelland 1986; McCelland & other 1986, pp.216-271). On the other hand, FL made a great advance in the mid 1970s with some successful results of laboratory experiments by Mamdani and Assilian (1975, pp.1-13). In 1985, Takagi and Sugeno (1985, pp.116-132) contributed FL with a new rule-based modeling technique. Operating with linguistic expressions, fuzzy logic can use the experiences of a human expert and also compensate for inadequate and uncertain knowledge about the system. On the other hand, ANNs have proven superior learning and generalizing capabilities even on completely unknown systems that can only be described by its input-output characteristics. By combining these features, more versatile and robust models, called “neuro-fuzzy” architectures have been developed (Culliere et al. 1995, pp.2009-2016).

In a control system the plant displaying nonlinearities has to be described accurately in order to design an effective controller. In obtaining the model, the designer has to follow one of two ways. The first one is using the knowledge of physics, chemistry, biology and the other sciences to describe an equation of motion with Newton’s laws, or electric circuits and motors with Ohm’s, Kirchhoff’ s or Lentz’s laws depending on the plant of interest. This is generally referred to as mathematical modeling. The second way requires the experimental data obtained by exciting the plant, and measuring its response. This is called system identification and is preferred in the cases where the plant or process involves extremely complex physical phenomena or exhibits strong nonlinearities.

Conventional control methods rely upon strong mathematical modeling, analysis, and synthesis. In case where mathematical models are available conventional control theory acts as a powerful tool for controlling even complex systems. On the other hand obtaining a mathematical model for a system can be rather complex and time consuming as it often requires some assumptions such as defining an operating point and doing linearization about that point and ignoring some system parameters, etc. This fact has recently led the researchers to exploit the neural and fuzzy techniques in modeling and control of complex systems.

Although fuzzy logic allows one to model (control) a system using human knowledge and experience with if-then rules, it is not always adequate on its own. This is also true for ANNs, which only deal with numbers rather than linguistic expressions. This deficiency can be overcome by combining the superior features of the two methods, as is performed in ANFIS architecture introduced by Jang (1993, pp.665-685). ANFIS architecture which was used in here as the controller of the dynamic system is generally encountered in the areas of function approximation, fault detection, medical diagnosis and control, (Gonzalez-Andujar et al. 2006, pp.115-123; Turner et al. 2006; Kim et al. 2006).

Adaptive Neuro Fuzzy Inference System was used, as an estimation method which has fuzzy input and output parameters. Then standard errors of ANFIS were benchmarked with Multinomial Logistic Regression as a non-linear regression method, and it’s seen that ANFIS is more efficient than Multinomial Logistic Regression with fuzzy diabetes dataset.

3. DATA MINING FINDINGS AND BENCHMARKING

3.1 Data Mining by WEKA Engine

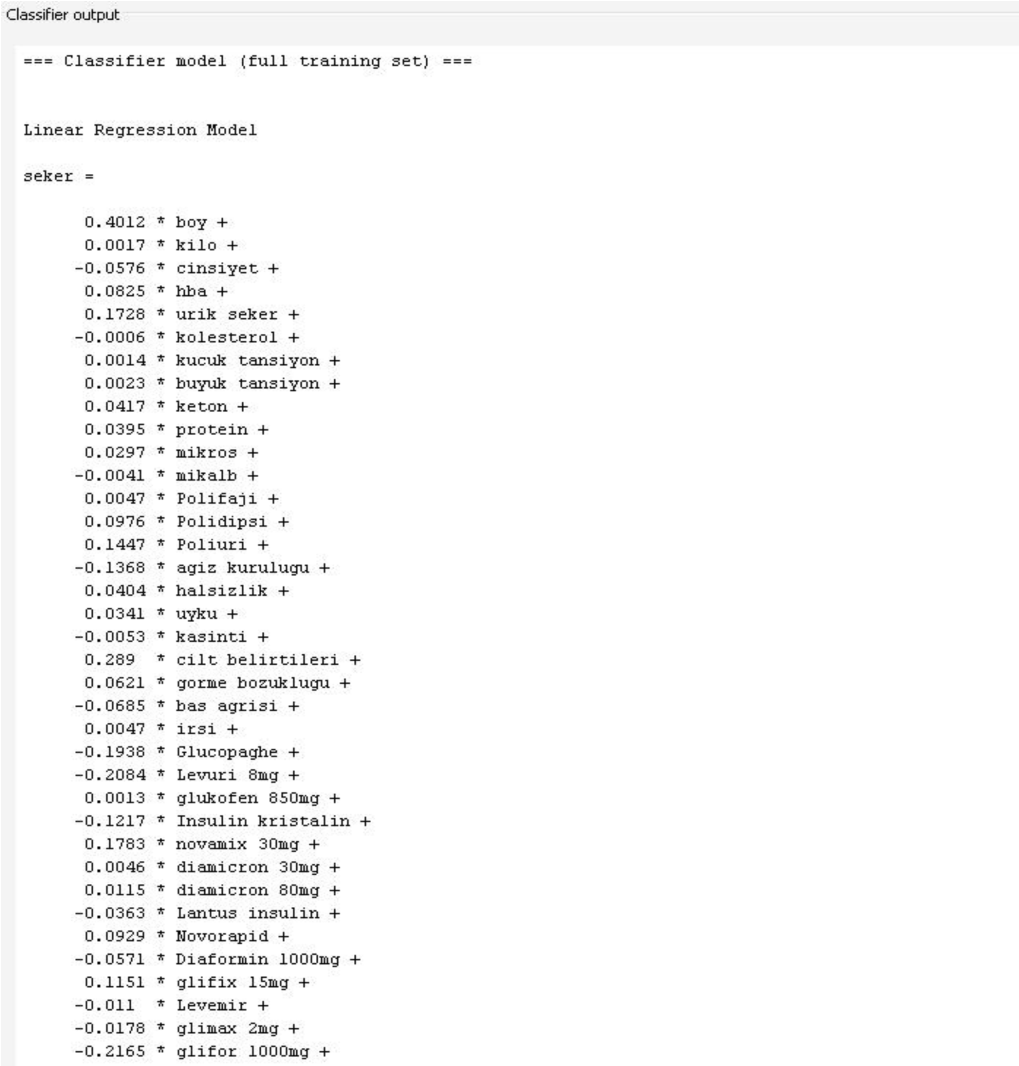
Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Weka is developed by the University of Waikato. In our thesis we used the Weka as data mining engine, and made a bridge between the Diabetes Expert System interface and Weka. No user needs to install Weka in his workstation, but it do already enough to be installed on server machine. In addition to this, the detail information is given in the section 2.

3.1.1 Preparation Phase of Dataset

Weka can directly reach the database and fetch the data from there. Or we need to prepare a dataset to be processed by Weka engine. We chose the second way, because we don't need the whole data (all columns of the table), and it's better to prepare an ARFF file, which is processed by Weka.

The Diabetes Expert System shows inclusion of the ARFF file that is prepared, after choosing some attributes of social-demographic data or all of them



```
=== Classifier model (full training set) ===


Linear Regression Model

seker =

       0.4012 * boy +
       0.0017 * kilo +
      -0.0576 * cinsiyet +
       0.0825 * hba +
       0.1728 * urik seker +
      -0.0006 * kolesterol +
       0.0014 * kucuk tansiyon +
       0.0023 * buyuk tansiyon +
       0.0417 * keton +
       0.0395 * protein +
       0.0297 * mikros +
      -0.0041 * mikalb +
       0.0047 * Polifaji +
       0.0976 * Polidipsi +
       0.1447 * Poliuri +
      -0.1368 * agiz kurulugu +
       0.0404 * halsizlik +
       0.0341 * uyku +
      -0.0053 * kasinti +
       0.289  * cilt belirtileri +
       0.0621 * gorme bozuklugu +
      -0.0685 * bas agrisi +
       0.0047 * irsi +
      -0.1938 * Glucopaghe +
      -0.2084 * Levuri 8mg +
       0.0013 * glukofen 850mg +
      -0.1217 * Insulin kristalin +
       0.1783 * novamix 30mg +
       0.0046 * diamicron 30mg +
       0.0115 * diamicron 80mg +
      -0.0363 * Lantus insulin +
       0.0929 * Novorapid +
      -0.0571 * Diaformin 1000mg +
       0.1151 * glifix 15mg +
      -0.011  * Levemir +
      -0.0178 * glimax 2mg +
      -0.2165 * glifor 1000mg +
```

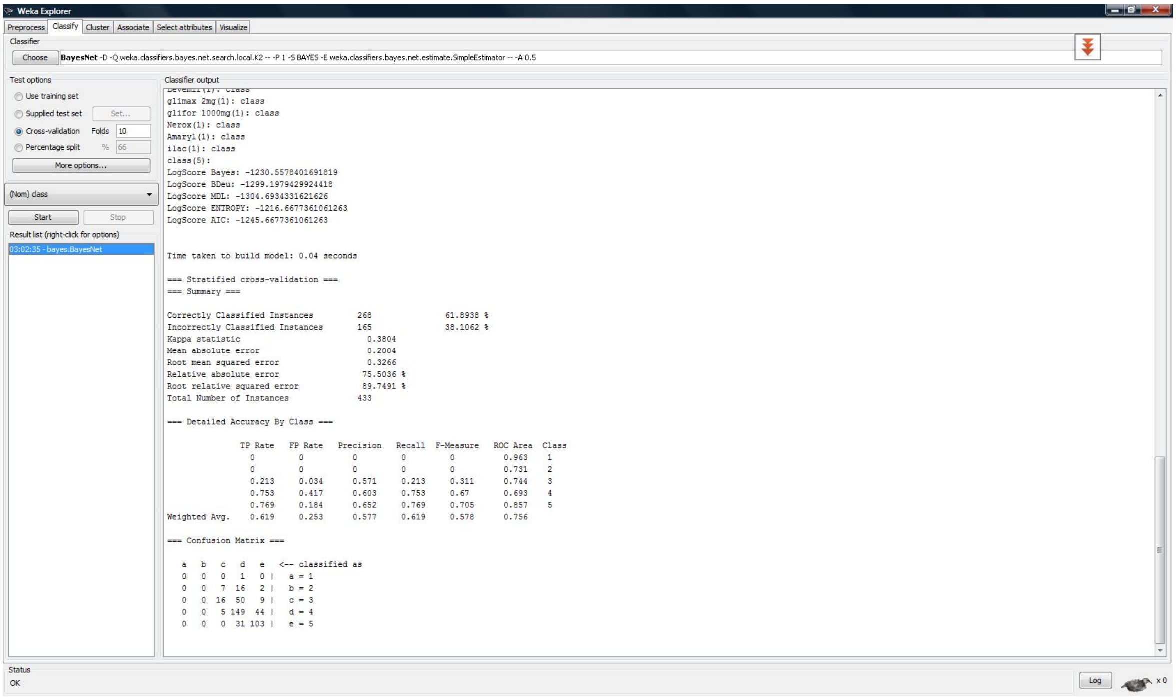
Figure 3.1 : Linear Regression Model

In Linear Regression the importance coefficients of the attributes van be seen.

3.1.2 Classification

3.1.2.1 Bayes Network Classifier

When we applied Bayes Network onto refreshed data, we got the results shown as below on Figure 3.2



BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

03:02:35 - bayes.BayesNet

```
Levemir(1): class
glimax 2mg(1): class
glifor 1000mg(1): class
Nerox(1): class
Amaryl(1): class
ilac(1): class
class(5):
LogScore Bayes: -1230.5578401691819
LogScore BDeu: -1299.1979429924418
LogScore MDL: -1304.6934331621626
LogScore ENTROPY: -1216.6677361061263
LogScore AIC: -1245.6677361061263


Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         268               61.8938 %
Incorrectly Classified Instances       165               38.1062 %
Kappa statistic                          0.3804
Mean absolute error                      0.2004
Root mean squared error                  0.3266
Relative absolute error                 75.5036 %
Root relative squared error             89.7491 %
Total Number of Instances              433

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0         0          0         0          0.963    1
                 0         0         0          0         0          0.731    2
                 0.213     0.034     0.571      0.213     0.311      0.744    3
                 0.753     0.417     0.603      0.753     0.67       0.693    4
                 0.769     0.184     0.652      0.769     0.705      0.857    5
Weighted Avg.    0.619     0.253     0.577      0.619     0.578      0.756

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
   0   0   0   1   0 |   a = 1
   0   0   7  16   2 |   b = 2
   0   0  16  50   9 |   c = 3
   0   0   5 149  44 |   d = 4
   0   0   0  31 103 |   e = 5
```

Status: OK

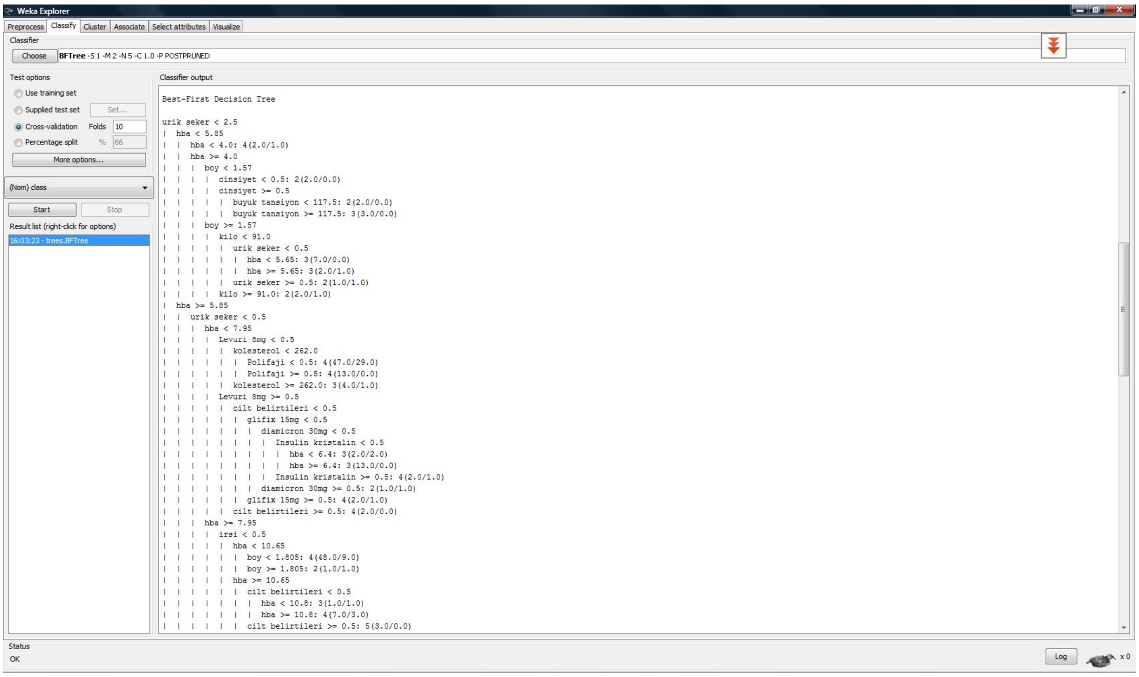
Figure 3.2 : Bayes network classifier

As seen correctly classified instances are low (61.8938) because there are so many attributes. Root mean squared error is 0.3266 and it shows the differences between values predicted by the model or the estimator. Confusion matrix shows us the data set is unbalanced and the data gathered in the attributes of 3,4,5. Log scores also given.

3.1.2.2 BF Tree Decision Tree

This is a module for building a best-first decision tree classifier. This class uses binary split for both nominal and numeric attributes. For missing values, the method of 'fractional' instances is used. This method used for pre-steps of bayes.

When we applied BF Tree onto refreshed data, we got the results shown as below on Figure 3.3.



Classifier: BFTree -S 1 -M 2 -N 5 -C 1.0 -P POSTPRUNED

Test options
○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Result list (right-click for options)
16:03:23 - trees.BFTree

Classifier output

```
Best-First Decision Tree

urik seker < 2.5
|  hba < 5.85
|  |  hba < 4.0: 4(2.0/1.0)
|  |  hba >= 4.0
|  |  |  boy < 1.57
|  |  |  |  cinsiyet < 0.5: 2(2.0/0.0)
|  |  |  |  cinsiyet >= 0.5
|  |  |  |  |  buyuk tansiyon < 117.5: 2(2.0/0.0)
|  |  |  |  |  buyuk tansiyon >= 117.5: 3(3.0/0.0)
|  |  |  boy >= 1.57
|  |  |  |  kilo < 91.0
|  |  |  |  |  urik seker < 0.5
|  |  |  |  |  |  hba < 5.65: 3(7.0/0.0)
|  |  |  |  |  |  hba >= 5.65: 3(2.0/1.0)
|  |  |  |  |  urik seker >= 0.5: 2(1.0/1.0)
|  |  |  |  kilo >= 91.0: 2(2.0/1.0)
|  hba >= 5.85
|  |  urik seker < 0.5
|  |  |  hba < 7.95
|  |  |  |  Levuri 8mg < 0.5
|  |  |  |  |  kolesterol < 262.0
|  |  |  |  |  |  Polifaji < 0.5: 4(47.0/29.0)
|  |  |  |  |  |  Polifaji >= 0.5: 4(13.0/0.0)
|  |  |  |  |  kolesterol >= 262.0: 3(4.0/1.0)
|  |  |  |  Levuri 8mg >= 0.5
|  |  |  |  |  cilt belirtileri < 0.5
|  |  |  |  |  |  glifix 15mg < 0.5
|  |  |  |  |  |  |  diamicron 30mg < 0.5
|  |  |  |  |  |  |  |  Insulin kristalin < 0.5
|  |  |  |  |  |  |  |  |  hba < 6.4: 3(2.0/2.0)
|  |  |  |  |  |  |  |  |  hba >= 6.4: 3(13.0/0.0)
|  |  |  |  |  |  |  |  Insulin kristalin >= 0.5: 4(2.0/1.0)
|  |  |  |  |  |  |  diamicron 30mg >= 0.5: 2(1.0/1.0)
|  |  |  |  |  |  glifix 15mg >= 0.5: 4(2.0/1.0)
|  |  |  |  |  cilt belirtileri >= 0.5: 4(2.0/0.0)
|  |  |  hba >= 7.95
|  |  |  |  irsi < 0.5
|  |  |  |  |  hba < 10.65
|  |  |  |  |  |  boy < 1.805: 4(48.0/9.0)
|  |  |  |  |  |  boy >= 1.805: 2(1.0/1.0)
|  |  |  |  |  hba >= 10.65
|  |  |  |  |  |  cilt belirtileri < 0.5
|  |  |  |  |  |  |  hba < 10.8: 3(1.0/1.0)
|  |  |  |  |  |  |  hba >= 10.8: 4(7.0/3.0)
|  |  |  |  |  |  cilt belirtileri >= 0.5: 5(3.0/0.0)
```

Status
OK

Figure 3.3 : Best-first decision tree

In this first decision tree, size is 93 and number of leaves are 47. This tree shows how weka determines the attributes to be classified. It shows the attributes range to decide and relations between them. In this tree correctly classified instances are 258 (%59.54) the reason of this low classification is there are so many attributes. Root mean squared error is 0.3402.

3.2 Rosetta Rough Set

3.2.1 Preprocessing

We manually preprocessed the data in MS-Notepad and through PERL scripts. This was done to remove unwanted space between the attribute values, and to insert attribute types and Undefined for missing values in the Rosetta input file.

The model used in AutoClass in the experiments to follow was Single multinomial. All attributes were treated as discrete values. The reason for this is that AutoClass expects a zero-point and a relative error for scalars. By using discrete values, we were able to make the input more similar to the input to Rosetta. The datasets were preprocessed in Rosetta and the splits used in Rosetta were exported using cut and pasted into MS-Notepad.

3.2.2 Experiment

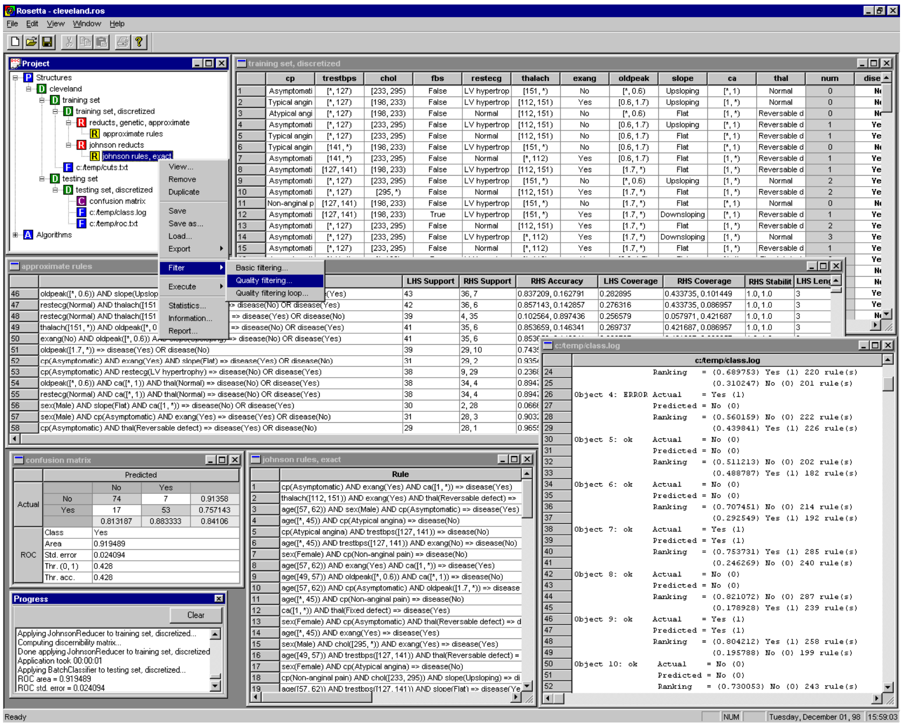
We manually read the archives and chose the attributes that we thought had some possible functional dependencies. The attributes, with explanation are listed in Table 2.3.

When we manually inspected the data we found that the values in attributes Glucopaghe and Insulin Lantus were mostly missing, so we removed them. We also removed three objects out of 470 that had most of the attribute values missing. table below shows the number of objects and attributes in the dataset of experiment.

Table 3.1: Facts on Experiment data set.

Name	Object	Attributes	Obj.w/miss val.
Rought set	470	18	156

The dataset was randomly split into a 30% training set, and a 70% test set.



training set, discretized

	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num	dise
1	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[151, *)	No	[*, 0.6)	Upsloping	[*, 1)	Normal	0	N
2	Typical angin	[*, 127)	[198, 233)	False	LV hypertrop	[112, 151)	Yes	[0.6, 1.7)	Upsloping	[1, *)	Normal	0	N
3	Atypical angi	[*, 127)	[198, 233)	False	Normal	[112, 151)	No	[*, 0.6)	Flat	[1, *)	Reversable d	0	N
4	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[112, 151)	No	[0.6, 1.7)	Upsloping	[*, 1)	Reversable d	1	Ye
5	Typical angin	[*, 127)	[233, 295)	False	Normal	[112, 151)	No	[0.6, 1.7)	Flat	[*, 1)	Reversable d	1	Ye
6	Typical angin	[141, *)	[198, 233)	False	LV hypertrop	[151, *)	No	[0.6, 1.7)	Flat	[*, 1)	Reversable d	0	N
7	Asymptomati	[141, *)	[233, 295)	False	Normal	[*, 112)	Yes	[0.6, 1.7)	Flat	[1, *)	Reversable d	1	Ye
8	Asymptomati	[127, 141)	[198, 233)	False	LV hypertrop	[112, 151)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	1	Ye
9	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[151, *)	No	[*, 0.6)	Upsloping	[1, *)	Normal	2	Ye
10	Asymptomati	[*, 127)	[295, *)	False	Normal	[112, 151)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	2	Ye
11	Non-anginal p	[127, 141)	[198, 233)	False	LV hypertrop	[151, *)	No	[1.7, *)	Flat	[*, 1)	Normal	0	N
12	Asymptomati	[127, 141)	[198, 233)	True	LV hypertrop	[151, *)	Yes	[1.7, *)	Downsloping	[*, 1)	Reversable d	1	Ye
13	Asymptomati	[*, 127)	[233, 295)	False	Normal	[112, 151)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	2	Ye
14	Asymptomati	[*, 127)	[233, 295)	False	LV hypertrop	[*, 112)	Yes	[1.7, *)	Downsloping	[1, *)	Normal	3	Ye
15	Asymptomati	[*, 127)	[233, 295)	False	Normal	[*, 112)	Yes	[1.7, *)	Flat	[1, *)	Reversable d	1	Ye

approximate rules

		LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage	RHS Stabilit	LHS Leng
46	oldpeak([*, 0.6)) AND slope(Upslop (Yes)	43	36, 7	0.837209, 0.162791	0.282895	0.433735, 0.101449	1.0, 1.0	3
47	restecg(Normal) AND thalach([151 => disease(No) OR disease(Yes)	42	36, 6	0.857143, 0.142857	0.276316	0.433735, 0.086957	1.0, 1.0	3
48	restecg(Normal) AND thalach([151 => disease(Yes) OR disease(No)	39	4, 35	0.102564, 0.897436	0.256579	0.057971, 0.421687	1.0, 1.0	3
49	thalach([151, *)) AND oldpeak([*, 0 => disease(No) OR disease(Yes)	41	35, 6	0.853659, 0.146341	0.269737	0.421687, 0.086957	1.0, 1.0	3
50	exang(No) AND oldpeak([*, 0.6)) A => disease(No) OR disease(Yes)	41	35, 6	0.8536				
51	oldpeak([1.7, *)) => disease(Yes) OR disease(No)	39	29, 10	0.7435				
52	cp(Asymptomatic) AND exang(Yes) AND slope(Flat) => disease(Yes) OR disease(No)	31	29, 2	0.9354				
53	cp(Asymptomatic) AND restecg(LV hypertrophy) => disease(No) OR disease(Yes)	38	9, 29	0.2368				
54	oldpeak([*, 0.6)) AND ca([*, 1)) AND thal(Normal) => disease(No) OR disease(Yes)	38	34, 4	0.8947				
55	restecg(Normal) AND ca([*, 1)) AND thal(Normal) => disease(No) OR disease(Yes)	38	34, 4	0.8947				
56	sex(Male) AND slope(Flat) AND ca([1, *)) => disease(No) OR disease(Yes)	30	2, 28	0.0666				
57	sex(Male) AND cp(Asymptomatic) AND exang(Yes) => disease(Yes) OR disease(No)	31	28, 3	0.9032				
58	cp(Asymptomatic) AND thal(Reversable defect) => disease(Yes) OR disease(No)	29	28, 1	0.9655				

confusion matrix

		Predicted		
		No	Yes	
Actual	No	74	7	0.91358
	Yes	17	53	0.757143
		0.813187	0.883333	0.84106
ROC	Class	Yes		
	Area	0.919489		
	Std. error	0.024094		
	Thr. (0, 1)	0.428		
	Thr. acc.	0.428		

```
Applying JohnsonReducer to training set, discretized...
Computing discernibility matrix...
Done applying JohnsonReducer to training set, discretized
Application took 00:00:01
Applying BatchClassifier to testing set, discretized...
ROC area = 0.919489
ROC std. error = 0.024094
```

johnson rules, exact

	Rule
1	cp(Asymptomatic) AND exang(Yes) AND ca([1, *)) => disease(Yes)
2	thalach([112, 151)) AND exang(Yes) AND thal(Reversable defect) =>
3	age([57, 62)) AND sex(Male) AND cp(Asymptomatic) => disease(Yes)
4	age([*, 45)) AND cp(Atypical angina) => disease(No)
5	cp(Atypical angina) AND trestbps([127, 141)) => disease(No)
6	age([*, 45)) AND trestbps([127, 141)) AND exang(No) => disease(No)
7	sex(Female) AND cp(Non-anginal pain) => disease(No)
8	age([57, 62)) AND exang(Yes) AND ca([1, *)) => disease(Yes)
9	age([49, 57)) AND oldpeak([*, 0.6)) AND ca([*, 1)) => disease(No)
10	age([57, 62)) AND cp(Asymptomatic) AND oldpeak([1.7, *)) => disease
11	age([*, 45)) AND cp(Non-anginal pain) => disease(No)
12	ca([1, *)) AND thal(Fixed defect) => disease(Yes)
13	sex(Female) AND cp(Asymptomatic) AND thal(Reversable defect) => d
14	age([*, 45)) AND exang(Yes) => disease(Yes)
15	sex(Male) AND chol([295, *)) AND exang(Yes) => disease(Yes)
16	age([49, 57)) AND trestbps([127, 141)) AND thal(Reversable defect) =
17	sex(Female) AND cp(Atypical angina) => disease(No)
18	cp(Non-anginal pain) AND chol([233, 295)) AND slope(Upsloping) => di
19	age([57, 62)) AND trestbps([127, 141)) AND slope(Flat) => disease(Ye

c:/temp/class.log

```
24                     Ranking   = (0.689753) Yes (1) 220 rule(s)
25                                 (0.310247) No (0) 201 rule(s)
26  Object 4: ERROR Actual    = Yes (1)
27                     Predicted = No (0)
28                     Ranking   = (0.560159) No (0) 222 rule(s)
29                                 (0.439841) Yes (1) 226 rule(s)
30  Object 5: ok     Actual    = No (0)
31                     Predicted = No (0)
32                     Ranking   = (0.511213) No (0) 202 rule(s)
33                                 (0.488787) Yes (1) 182 rule(s)
34  Object 6: ok     Actual    = No (0)
35                     Predicted = No (0)
36                     Ranking   = (0.707451) No (0) 214 rule(s)
37                                 (0.292549) Yes (1) 192 rule(s)
38  Object 7: ok     Actual    = Yes (1)
39                     Predicted = Yes (1)
40                     Ranking   = (0.753731) Yes (1) 285 rule(s)
41                                 (0.246269) No (0) 240 rule(s)
42  Object 8: ok     Actual    = No (0)
43                     Predicted = No (0)
44                     Ranking   = (0.821072) No (0) 287 rule(s)
45                                 (0.178928) Yes (1) 239 rule(s)
46  Object 9: ok     Actual    = Yes (1)
47                     Predicted = Yes (1)
48                     Ranking   = (0.804212) Yes (1) 258 rule(s)
49                                 (0.195788) No (0) 199 rule(s)
50  Object 10: ok     Actual    = No (0)
51                      Predicted = No (0)
52                      Ranking   = (0.730053) No (0) 243 rule(s)
```

Figure 3.4: Rosetta work screen.

Table 3.2: Rosetta confusion matrix.

			Prediction		
	Values	Decision1	Decision2	Decision3	Row Ratio
Actual	Decision1	310	77	83	0,664114
	Decision2	50	293	115	0,756655
	Decision3	22	15	430	0,951514
	Column Ratio	0,735485	0,94822	0,762121	0.77458

3.2.3 Results from Rosetta

We wanted to see epoch on train and check phase also see correctness. We chose the attribute Glucose level as decision attribute. Then we calculated dynamic reducts, both normal and object oriented, and generated rules. We tested the rules on the test set, and the correctness was around 91%. We then tried to remove four of the attributes that had missing attributes, but this did not improve the results.

3.2.4 Results from AutoClass- Bayesian Network

After 62 tries, AutoClass found 8 classes. The strength of each class is given table 3.4

Table 3.3: Strength and normalized weight for the classes.

Class	Strength	Norm Weight
0	-2.38e+01	0.201
1	-2.47e+01	0.175
2	-2.60e+01	0.160
3	-2.40e+01	0.133
4	-2.83e+01	0.106
5	-2.25e+01	0.088
6	-2.50e+01	0.085
7	-2.66e+01	0.048
8	-1.88e+01	0.011

3.3 Comparing Techniques

In this section, the performance of ANFIS was evaluated, by compared with the performance of Multinomial Logistic Regression and Rosetta-rough set.

After preparation of dataset explained in material and method section, ANFIS module was run for training and checking datasets before running ANFIS.

Matlab integration was used for ANFIS. The result of ANFIS is read from Matlab Logs although we used 430 instances for training and 90 ones for checking, ANFIS reached the results at just epoch 3.

After training and checking the data, estimation module by ANFIS could be run. The estimation module uses the rules found by Matlab engine, and estimates a glucose rate using input variables.

Same training and checking datasets were run by multinomial logistic regression (MLR) model in SPSS and Rough set of Rosetta. And then the results shown as Table 3.1 were reached.

Table 3.4 : Results of benchmarking

Method	**epoch**	**Standard Error**	**Error percentage**	**Data Type**
ANFIS	3	0.9095	**0.0418**	Train
ANFIS	3	0.8725	**0.0745**	Check
MLR	300	0.9083	**0.0417**	Train
MLR	90	0.8923	**0.1343**	Check
Rough set	48	0.9090	**0.0418**	Train
Rough set	77	0.8878	**0.1014**	Check

I would like to have your attention to 2 important points while evaluating the results. It's clear to see from Figure 3.5; standard errors of training datasets for all of the methods are very similar. However same parameter of checking datasets shows that multinomial logistic regression and rough set has much bigger standard error than ANFIS.

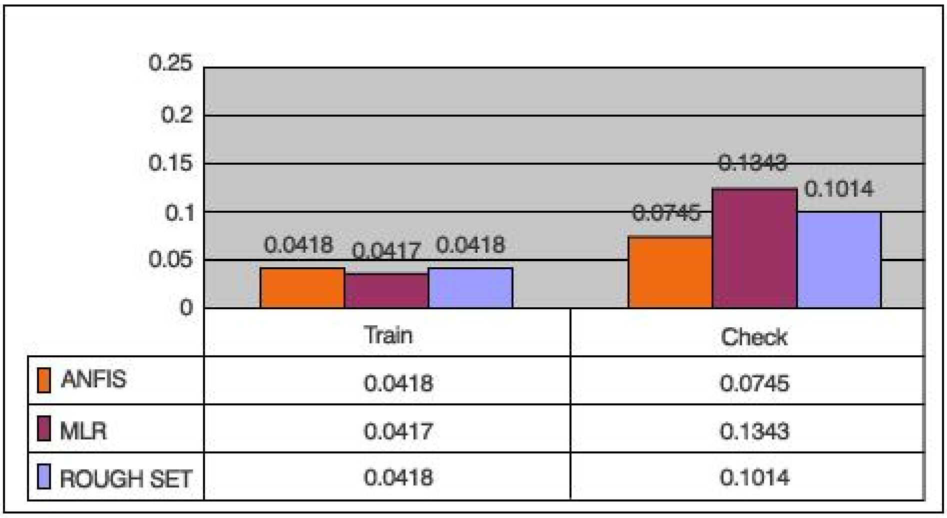


Figure 3.5 : Error percentages

Another important point is difference between learning durations of the methods. As seen from Table 4.4, the learning duration of ANFIS is shorter than multinomial logistic regression and rough set. ANFIS training could be completed at epoch 3; but MLR should evaluate the whole dataset and rough set should evaluate nearly %70 of the data set. Many medical tests have issues with **sensitivity** and **specificity**. Before dealing with the terms, it's better to explain what they mean.

Sensitivity: is the ability to correctly detect diabetes disease.

Specificity: is the ability to avoid calling Normal things as diabetes disease.

A perfect diabetes test would have 100 percent sensitivity and 100 percent specificity. It would positively identify all the true cases of disease, and it would never mislabel anything normal as disease. When a diabetes test is imperfect, the expert system should try to strike a balance between sensitivity and specificity. To do this, a chart might be plotted which is formed by sensitivity and 1-specificity on a graph, called a "ROC curve". ROC means Receiver Operator Characteristic. ROC curve is drawn in Figure 3.6 according to the Check Data of ANFIS and MLR.

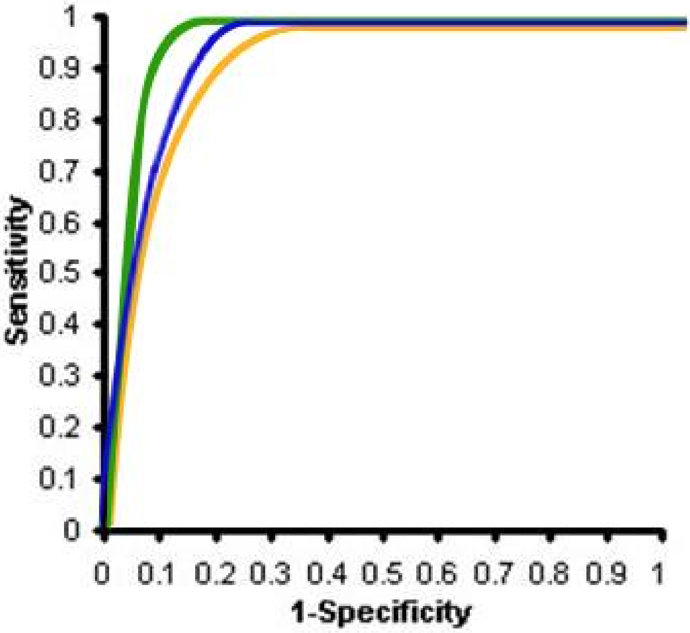


Figure 3.6 : ROC curve of check data

The **GREEN** line (ANFIS) is a better medical test than the **yellow** line (MLR) and blue line (Rough Set), because the curve of the yellow line and **blue** line comes closer to the **GREEN** line. To pick the best point along the ROC curve, it's generally looked for the shortest distance from the **yellow** line, to the **Green** line. In this case of the thesis, ANFIS gives **0.92** sensitivity (92 percent) and 0.11 false positive fraction (**86 percent** specificity), MLR gives **0.88** sensitivity (88 percent) and 0.14 false positive fraction (**88 percent** specificity) and Rough Set gives 0.91 sensitivity (91 percent) and 0.12 false positive fraction(**87 percent** specificity).

Since, ANFIS is the best method to segment diabetic patients. The best rule ANFIS created on data set is (with coefficients):

11 Data

0.11817 Gender

0.10258 Frame

0.14112 Cholestrol

0.171285 Hba%

0.09251 Tension

0.02358 Keton

0.09512 Protein

0.01254 Headache

0.11577 Polidipsi

0.06842 Poliuri

0.058905 Xerostomia

These coefficients show the importance of the attribute in finding the segment of the diabetic patient, the segmantion shown on the second section Table 2.2 as output, by this rule ANFIS can give the best results on diabetic patients segmentation. As seen the most important attribute is Hba%.

4. CONCLUSION

Some data mining techniques are benchmarked for segmenting diabetes patients. The main purpose of this thesis is to be reference on segmenting diabetic patients' data. Biomedicines may have benefits from this thesis by using the mentioned techniques. For this aim, I have made a benchmarking table for the use of the biomedicines.

Continuous values are initially designated in the dataset, and then converted to fuzzy values. Glucose rates (dependent variable) were made fuzzy, instead of binary. Binary values have high accuracy, but don't have enough information about diabetes risk. After preprocessing dataset, I run ANFIS method in Matlab, Multinomial Logistic Regression method in SPSS and Rough Set method on Rosetta. Table summarizes the results of benchmark we made between ANFIS, multinomial logistic regression and rough set methods.

It's found out that learning duration of ANFIS is much shorter than MLR's duration and Rough set's duration. When a more sophisticated system with a huge data is imagined, the use of ANFIS instead of multinomial logistic regression and Rough set method would be more useful to overcome faster the complexity of the problem.

In training of the data, ANFIS, MLR and Rough Set gave quite similar results with standard error. However, when the trained parameters were applied to checking data, standard error of ANFIS is smaller than that of MLR and also smaller then the Rough set. This shows that ANFIS is a better and faster learning method than multinomial logistic regression and Rough Set. Consequently it could be said, if we have a system which contains fuzzy inputs and output, ANFIS is better system than MLR and Rough Set for diabetes segmentation.

REFERENCES

Books

Agrawal R., & Srikant R., (1994), *Fast Algorithms for Mining Association Rules*, VLDB. Chile, 487-99, ISBN 1-55860-153-8

Chong, Q., Marwadi, A., Supekar, K. & Lee, Y., (2005*), Ontology Based Metadata Management in Medical Domains*, Published by School of Interdisciplinary Computing and Engineering University of Missouri – Kansas City

Dobroslawa, M., Grzymala-Busse, M., & Grzymala-Busse, W, (1995), *On The Usefulness Of Machine Learning Approach To Knowledge Acquisition, Department of Electrical Engineering and Computer Science*, University of Kansas, Lawrence, KS 66045, USA

Karagiannis, S., Dounis, A.I., Chalastras, T., Tiropanis, P. & Papachristos, D., (2006), *Design of Expert System for Search Allergy and Selection of the Skin Tests using CLIPS*, INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY VOLUME 3 NUMBER 2 ISSN, 1305-2403.

McCelland, J.L., Rumelhart, D.E. & PDP Research Group (Eds), (1986), *Parallel Distrubuted Processing: Exploratios in The Microstructeres of Cognition*, pp.216-271. Cambridge, MA: Bradford / MIT Press.

Patterson, D. W., *Artificial Neural Networks* – Theory and Applications, Prentice

Pop, D. & Negru, V., (2003), *An Extensible Environment for Expert System Development,* Department of Computer Science, University of the West from Timi_oara 4 V. Parvan Street, RO-1900 Timioara, Romania.

Quinlan, J.R., (1985), *Induction of Decision Trees, Centre for Advanced Computing Sciences,* New South Wales Institute of Technology, Sydney 2007, Australia

Articles

Ada (2002), The *Prevention or Delay of Type 2 Diabetes*, Diabetes Care, 25: 742-749.

Agrawal, R., Imielinski T., & Swami AN., (June 1993), *Mining Association Rules between Sets of Items in Large Databases.* SIGMOD, **22**(2):207-16.

Culliere, T., Titli, A., & Corrieu, J., (1995), *Neuro-fuzzy modelling of nonlinear systems for control purposes,* In Proc. IEEE INT. Conf. on Fuzzy Systems, pp 2009-2016, Yokohama.

Edoura-Gaena, R.B., Allais, I., Gros, J.B., & Trystram., G., (2006), *A decision support system to control the aeration of sponge finger batters*, Elsevier, Food Control **17**, 585–596.

Edoura-Gaena, R.B., Allais, I., Gros, J.B., & Trystram., G., (2006), *A decision support system to control the aeration of sponge finger batters"*, Elsevier, Food Control **17**, 585–596.

Edward R., & Omiecinski., (2003), *Alternative interest measures for mining associations in databases.* IEEE Transactions on Knowledge and Data Engineering, **15**(1):57-69.

Flach P.A., & N. Lachiche, (1999), *Confirmation-Guided Discovery of first-order rules with Tertius. Machine Learning,* **42**, 61-95.

Fleming, G., van der Merwe, M., & McFerren, G., (2007), *Fuzzy expert systems and GIS for cholera health risk prediction in southern Africa*, Elsevier, Environmental Modelling & Software **22**, 442 – 448.

Ghavazi S. N. & T. W. Liao (2004), *Medical data mining by fuzzy modeling with selected features,* Elsevier, Expert Systems with Applications.

Gonzalez-Andujar, J.L., Fernandez-Quintanilla, C., Izquierdo, J., & Urbano, J.M., (2006), *SIMCE: An expert system for seedling weed identification in cereals*, Elsevier, Computers and Electronics in Agriculture **54**, 115–123.

Guler, I., & Ubeyli, E. D., (2006), *Expert systems for time-varying biomedical signals using eigenvector methods,* Elsevier, Expert Systems with Applications.

Gutierrez-Estrada, J.C., De Pedro Sanz, E., Lopez-Luque, R,. & Pulido-Calvo, I., (2006), *SEDPA, an expert system for disease diagnosis in eel rearing systems*, Elsevier, Aquacultural Engineering **33**, 110–125.

Huang, M.J., & Chen, M.Y., (2007), *Integrated design of the intelligent web-based Chinese Medical Diagnostic System (CMDS) – Systematic development for digestive health,* Elsevier, Expert Systems with Applications **32**, 658–673.

Hwang, G.H., Chen, J.M., Hwang, G.J., & Chu, H.C, (2006), *A time scale-oriented approach for building medical expert systems,* Elsevier, Expert Systems with Applications **31**, 299–308.

Jang, J. (1993), *ANFIS: Adaptive-Network Based Fuzzy Inference System*. IEEE Trans. On Systems, Man and Cybernetics, Vol. **23**. No. 3 pp. 665-685.

Keles, A., & Keles, A., (2006), ESTDD: *Expert system for thyroid diseases diagnosis*, Elsevier, Expert Systems with Applications.

Kim, J.S., (2006*), Development of a user-friendly expert system for composite laminate design*, Elsevier, Composite Structures.

Kirk, E & J. Mistry,(2009), *Clinics in Sports Medicine*, Volume 28, Issue **3**, pp. 344 - 354

Mamdani, E.H., & Assilian, S., (1975), *An Experiment in Linguistic Synthesis with a Logic Controller,* Int. J. Man – Machine Studies **8**, pp. 1 – 13.

Mangalampalli, A., Mangalampalli, S. M., Chakravarthy, R., & Jain, A. K., (2006), *A neural network based clinical decision-support system for efficient diagnosis and fuzzy-based prescription of gynecological diseases using homoeopathic medicinal system,* Elsevier, Expert Systems with Applications **30**, 109–116.

Nammuni, K., Pickering, C., Modgil, S., Montgomery, A., Hammond, P., Wyatt, J.C., Altman, D.G., Dunlop, R., & Potts, H.W.W., (2004), *Design-a-trial: a rule-based decision support system for clinical trial design*, Elsevier, Knowledge-Based Systems **17**, 121–129.

Nauck, D., (1994), *Fuzzy neuro systems: An overview*; R., Gebhardt J., & R. Palm, R., (1994), *Fuzzy Systems in Computer Science*, pp 91-107, Vieweg, Braunschweig,.

Pawlak, Z. (2000), *Cardozo Law Review*, Intelligent Processing and Manufacturing of Materials, IPMM '99. Proceedings of the Second International Conference on. 08/1999; 1:37-44 vol.1.

Pawlak, Z. (2002), *Rough set theory for intelligent industrial applications*, Information sciences.

Piasecki, M., & Sas, J., (2005)*, Partial Parsing Method Applied to Rules Acquisition for Medical Expert System,* Computer Science Department of Wroclaw University of Technology ul. Wybrzeze Wyspianskiego **27**, 50-370, Wroclaw, Poland.

Polat, K. & Gunes, S., (2006), *An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease*, Elsevier, Digital Signal Processing.

Pontow, C., Dazinger, F. & Schubert, R., (2007), *A framework in prolog for computing structural relationship*, Elsevier, Data & Knowledge Engineering **62**, 308–326.

Prasad, R., Ranjan, K. R. & Sinha, A.K., (2006), *AMRAPALIKA: An expert system for the diagnosis of pests, diseases, and disorders in Indian mango*, Elsevier, Knowledge-Based Systems **19**, 9–21.

Seising, R., (2006), *From vagueness in medical thought to the foundations of fuzzy reasoning in medical diagnosis*, Elsevier, Artificial Intelligence in Medicine.

Shaalan, K., Rafea, M. & Rafea, A., (1998), *KROL: a knowledge presentation object language on top of Prolog, Pergamon*, Expert Systems with Applications **15**, 33 – 46.

Sojda, R.S., (2007), *Empirical evaluation of decision support systems: Needs, definitions, potential methods, and an example pertaining to waterfowl management,* Elsevier, Environmental Modelling & Software **22**, 269 - 277.

Staszewski, W.J., & Worden, K., (1997), *Classification of Faults in Gearboxes Preprocessing Algorithms and Neural Networks*, Neural Computing and Applications **5**(3): 160-183.

Takagi, S., & Sugeno, M, (1985), *Fuzzy identification of fuzzy systems and it'sapplication to modelling and control,* IEEE Trans. Systems Man Cybern., **15** pp 116-132.

Türkoğlu, İ (2004), *An intelligent diagnosis system based on principle component analysis and ANFIS for the hearth valve diseases,* Firat University, Department of Electronic and Computer Science,Elaziğ, Turkey.

Temurtaş, F. (2003), *A comparative study on thyroid disease diagnosis using neural networks* Sakarya University, Department of Computer Engineering

Turner, C., Bishay, H., Bastien, G., Peng, B. & Phillips, R.C., (2006), *Configuring policies in public health applications*, Elsevier, Expert Systems with Applications.

VITAE

İnayet Adalı was born in Ankara, Turkey on November 23, 1985. He was graduated from Ankara Özel Bilkent high school, in 2003 He received his B.Sc. degree in Computer Engineering from Bahçeşehir University in 2007. His main areas of interest are machine learning, programming, bio-medical and Artificial Intelligence.