INTELLIGENT HEART DISEASE PREDICTION SYSTEM USING NAÏVE BAYES

PROJECT REPORT

Submitted by

C. NARMADA

Reg. No: 0920108012

In partial fulfillment for the award of the degree

Of

MASTER OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOG

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 049

APRIL 2011

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 049

Department of Computer Science and Engineering

PROJECT WORK

APRIL 2011

This is to certify that the project entitled

INTELLIGENT HEART DISEASE PREDICTION SYSTEM USING NAÏVE BAYES

is the bonafide record of project work done by

C. NARMADA

Register No: 0920108012

of M.E. (Computer Science and Engineering) during the year 2010-2011.

Project Guide

Head of the Department

Submitted for the Project Viva-Voce examination held on 25.04.2011

DECLARATION

I affirm that the project work titled "**Intelligent Heart Disease Prediction System Using Naïve Bayes**" being submitted in partial fulfillment for the award of **M.E.** degree is the original work carried out by me. It has not formed the part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

C. Narmada

C. NARMADA

Register No: 0920108012

I certify that the declaration made above by the candidate is true

Ms. D. CHANDRAKALA, M.E., (PhD)

Assistant Professor

Department of Computer Science and Engineering,
Kumaraguru College of Technology,
(An Autonomous Institution)
Coimbatore - 641 049.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for enabling me to complete this project.

I express my profound gratitude to our *Chairman* **Padmabhusan Arutselvar Dr. N. Mahalingam, B.Sc., F.I.E.,** for giving this opportunity to pursue this course.

I would like to thank **Dr. S. Ramachandran, PhD.,** *Principal* for providing the necessary facilities to complete my thesis.

I take this opportunity to thank **Ms.P.Devaki, M.E., (PhD),** *Head of the Department,* Computer Science and Engineering, for his precious suggestions.

I thank all project committee members for their comments and advice during the reviews. Special thanks to **Mr.V.Subramani M.Tech,** *Associate Professor*, and **Ms.V.Vanitha, M.E., (PhD),** *Associate Professor*, Department of Computer science and Engineering, for arranging the brain storming project review sessions.

I register my hearty appreciation to my Guide **Ms. D. Chandrakala, M.E., (PhD),** *Associate Professor,* Department of Computer Science and Engineering, who is my thesis advisor. I thank her for her support, encouragement and ideas.

I would like to convey my honest thanks to all **Teaching** and **Non Teaching** staff members of the department for their support. I would like to thank all my classmates for extending technical support whenever I needed it.

I dedicate this project work to my **Parents,** without whose love, this work wouldn't have been possible.

TABLE OF CONTENTS

SNS COLLEGE OF TECHNOLOGY

An NBA Accredited,UGC Recognised & ISO 9001:2000 Certified Institution
(Approved by AICTE & Affiliated to Anna University of Technology,Coimbatore)
Coimbatore-641 035
Department of Computer Science & Engineering (UG&PG)

SNS INSTITUTIONS

Sixth National Conference on

Recent Trends in Advanced Computing

NCRTAC '10

Certificate

This is to certify that Dr./Prof./Mr./Mrs./Ms. C. NARMADA, II M.E (CSE) of Managuru College of Technology has a presented a technical paper entitled Intelligent Heart Disease Prediction System Using Data mining techniques in National Conference held on 6th October 2010.

Organizing Secretary | Convenor | Principal | Director

ஆய்வுச் சுருக்கம்

ஆரோக்கியம் தொடர்பான விவரங்களை மருத்துவ துறையில் அதிக அளவில் சேகரித்து வைத்திருந்தாலும் துரதிர்ஷ்டவசமாக நிலையான முடிவுகளை எடுப்பதற்கு பயன்படவில்லை. நோயின் தன்மையை கண்டுபிடிப்பதற்கான மாதிரிகளை கண்டறியவதேயில்லை. முன்னேறிய தரவுகள் கண்டறியும் தொழிற்நுட்பம் மட்டுமே இப்பிரச்சனையை தீர்க்க இயலும். இந்த ஆய்வின் மூலகாரணம் இதயம் சம்பந்தமான நோய்களை நேவிபயாஸ் பயன்படுத்தி முன்கூட்டியே அறிந்து கொள்ளும் முறையை உருவாக்குதல் ஆகும். இம்முறையை கணினி இணையதளத்தில் வினா தொடர்மூலம் அமுல்படுத்தலாம். பயன்படுத்துவோரின் பதில்களை வைத்து இதய நோய் கண்டறியப்படுகிறது. இதய நோய் சம்பந்தமான சிக்கலான கேள்விகளுக்கு பதிலளிக்கும். மேலும் மருத்துவர்களுக்கும் இதய நோயைக் கண்டறிவதற்கு உதவிகரமாக இருக்கும். இதன்மூலம் சரியான சிகிச்சை கிடைப்பதுடன் மருத்துவச் செலவும் குறைகிறது.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
HD	Heart Disease
EDA	Exploratory Data Analysis
OLAP	On-Line Analytical Processing
ROI	Return On Investment
CANFIS	CoActive Neuro-Fuzzy Inference System
ECG	Electrocardiogram
APC	Atrial Premature Contraction
FCM	Fuzzy C – Means
PCA	Principal Component Analysis
CHD	Coronary Heart Disease
KDD	Knowledge Discovery in Databases
IHDPS	Intelligent Heart Disease Prediction System
CRISP-DM	Cross Industry Standard Process for Data Mining
NB	Naïve Bayes

1. PROBLEM DEFINITION

A major challenge facing healthcare organizations (hospitals, medical centers) is the provision of quality services at affordable costs. Quality service implies diagnosing patients correctly and administering treatments that are effective. Poor clinical decisions can lead to disastrous consequences which are therefore unacceptable. Clinical decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. The integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and reduces diagnosis process.

Most hospitals today employ some sort of hospital information systems to manage their healthcare or patient data. Unfortunately, these data are rarely used to support clinical decision making. The main objective of this research is to develop a prototype Intelligent Heart Disease Prediction System with Naïve Bayes algorithm using historical heart disease databases to make intelligent clinical decisions which traditional decision support systems.

The cost of management of HD is a significant economic burden and so prevention of heart disease is very important step in the management. Prevention of HD can be approached in many ways including health promotion campaigns, specific protection strategies, life style modification programs, early detection and good control of risk factors and constant vigilance of emerging risk factors.

2. INTRODUCTION

2.1 INTRODUCTION TO DATA MINING [2, 3]

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, cut 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.

2.1.1 DATA, INFORMATION, AND KNOWLEDGE

DATA

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

- Operational or transactional data such as, sales, cost, inventory, payroll, and accounting
- Non-operational data, such as industry sales, forecast data, and macro economic data
- Meta data – data about the data itself, such as logical database design or data dictionary definitions

INFORMATION

The patterns, associations, or relationships among all this data can provide information. For example, analysis of retail point of sale transaction data can yield information on which products are selling and when.

KNOWLEDGE

Information can be converted into knowledge about historical patterns and future trends. For example, summary information on retail supermarket sales can be analyzed in light of promotional efforts to provide knowledge of consumer buying behavior. Thus, a manufacturer or retailer could determine which items are most susceptible to promotional efforts.

2.1.2 DATA WAREHOUSES

Dramatic advances in data capture, processing power, data transmission, and storage capabilities are enabling organizations to integrate their various databases into data warehouses. Data warehousing is defined as a process of centralized data management and retrieval. Data warehousing represents an ideal vision of maintaining a central repository of all organizational data. Centralization of data is needed to maximize user access and analysis. Dramatic technological advances are making this vision a reality for many companies. And, equally dramatic advances in data analysis software are allowing users to access this data freely. The data analysis software is what supports data mining.

2.1.3 WHAT CAN DATA MINING DO?

Data mining is primarily used today by companies with a strong consumer focus – retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

With data mining, a retailer could use point-of-sale records of customer purchases to send targeted promotions based on an individual's purchase history. By mining demographic data from comment or warranty cards, the retailer could develop products and promotions to appeal to specific customer segments.

For example, Blockbuster Entertainment mines its video rental history database to recommend rentals to individual customers. American Express can suggest products to its cardholders based on analysis of their monthly expenditures.

2.1.4 How does data mining work?

Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries.

Data mining consists of five major elements:

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.
- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table

2.2 Data Mining Techniques [4]

The ultimate goal of data mining is prediction. Predictive data mining is the most common type of data mining and one that has the most direct business applications. The process of data mining consists of three stages:

(1) Initial exploration
(2) Model building with validation/verification
(3) Deployment

2.2.1 Exploration

This stage usually starts with data preparation which may involve cleaning data, data transformations, and selecting subsets of records and – in case of data sets with large numbers of variables ("fields") performing some preliminary feature selection operations to bring the number of variables to a manageable range. Then, depending on the nature of the analytic problem, this first stage of the process of data mining may involve anywhere between a

simple choice of straightforward predictors for a regression model, to elaborate exploratory analyses using a wide variety of graphical and statistical methods, in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

2.2.2 Model Building and Validation

This stage involves considering various models and choosing the best one based on their predictive performance. There are a variety of techniques developed to achieve that goal – many of which are based on so-called "competitive evaluation of models". This means that applying different models to the same data set and then comparing their performance to choose the best. These techniques – which are often considered the core of predictive data mining – include: Bagging (voting, averaging), Boosting (to generate multiple models or classifiers (for prediction or classification), and to derive weights to combine the predictions from those models into a single prediction or predicted classification), Stacking and Meta-Learning (to combine the predictions from multiple models. It is particularly useful when the types of models included in the project are very different. In this context, this procedure is also referred to as Stacking (Stacked Generalization)).

2.2.3 Deployment

The final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

The concept of Data Mining is becoming increasingly popular as a business information management tool where it is expected to reveal knowledge structures that can guide decisions in conditions of limited certainty. Data Mining is still based on the conceptual principles of statistics including the traditional Exploratory Data Analysis (EDA) and modeling and it shares with them both some components of its general approaches and specific techniques.

However, an important general difference in the focus and purpose between Data Mining and the traditional Exploratory Data Analysis (EDA) is that Data Mining is more oriented towards applications than the basic nature of the underlying phenomena. In other words, Data Mining is relatively less concerned with identifying the specific relations

between the involved variables. For example, uncovering the nature of the underlying functions or the specific types of interactive, multivariate dependencies between variables are not the main goal of Data Mining. Instead, the focus is on producing a solution that can generate useful predictions. Therefore, Data Mining accepts among others, a "black box" approach to data exploration or knowledge discovery and uses not only the traditional Exploratory Data Analysis (EDA) techniques, but also uses techniques such as Neural Networks which can generate valid predictions but are not capable of identifying the specific nature of the interrelations between the variables on which the predictions are based.

2.3 AN ARCHITECTURE FOR DATA MINING [5]

To best apply these advanced techniques, they must be fully integrated with a data warehouse as well as flexible interactive business analysis tools. Many data mining tools currently operate outside of the warehouse, requiring extra steps for extracting, importing, and analyzing the data. Furthermore, when new insights require operational implementation, integration with the warehouse simplifies the application of results from data mining. The resulting analytic data warehouse can be applied to improve business processes throughout the organization, in areas such as promotional campaign management, fraud detection, new product rollout, and so on. Figure 2.1 illustrates architecture for advanced analysis in a large data warehouse.

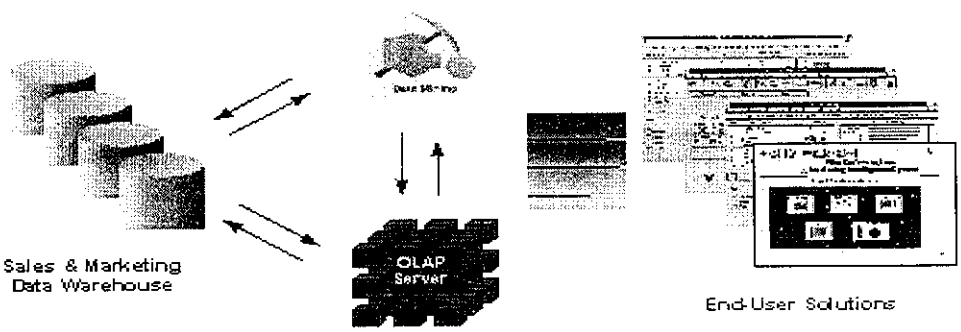


FIGURE 2.1 - INTEGRATED DATA MINING ARCHITECTURE

The ideal starting point is a data warehouse containing a combination of internal data tracking all customer contact coupled with external market data about competitor activity. Background information on potential customers also provides an excellent basis for prospecting. This warehouse can be implemented in a variety of relational database systems: Sybase, Oracle, Redbrick, and so on, and should be optimized for flexible and fast data access.

An OLAP (On-Line Analytical Processing) server enables a more sophisticated end-user business model to be applied when navigating the data warehouse. The multidimensional structures allow the user to analyze the data as they want to view their business – summarizing by product line, region, and other key perspectives of their business. The Data Mining Server must be integrated with the data warehouse and the OLAP server to embed ROI-focused business analysis directly into this infrastructure. An advanced, process-centric metadata template defines the data mining objectives for specific business issues like campaign management, prospecting, and promotion optimization. Integration with the data warehouse enables operational decisions to be directly implemented and tracked. As the warehouse grows with new decisions and results, the organization can continually mine the best practices and apply them to future decisions.

This design represents a fundamental shift from conventional decision support systems. Rather than simply delivering data to the end user through query and reporting software, the Advanced Analysis Server applies users' business models directly to the warehouse and returns a proactive analysis of the most relevant information. These results enhance the metadata in the OLAP Server by providing a dynamic metadata layer that represents a distilled view of the data. Reporting, visualization, and other analysis tools can then be applied to plan future actions and confirm the impact of those plans.

2.4 DATA MINING IN HEALTHCARE INDUSTRY [6]

Recently, the requirement of effective recognition of information- contextual data - non obvious and important for decision making, from a huge ensemble of data has been constantly rising. This is an interactive and iterative process consisting of numerous subtasks and decisions and is called as Knowledge Discovery from Data. The essential process of

Knowledge Discovery is the conversion of Intelligent and Effective Heart Attack Prediction System Using Naive Bayes in order to aid in decision making, referred to as Data Mining. Knowledge discovery in databases comprises of several distinct clearly exemplified processes.

The essential process is that of data mining; the one that assists the identification of concealed yet valuable knowledge from enormous databases. A broadly recognized formal definition of data mining is given as "Data mining is the non trivial extraction of implicit previously unknown and potentially useful information about data". Traditionally, the mined information is represented as a model of the semantic structure of the dataset. It might be possible to employ the model in the prediction and classification of new data.

A wide variety of areas including Marketing, Customer Relationship Management, Engineering, Medicine, Crime Analysis, Expert Prediction, Web Mining, and Mobile Computing, besides others utilize Data Mining. Numerous fields associated with medical services like prediction of effectiveness of surgical procedures, medical tests, medication, and the discovery of relationships among clinical and diagnosis data as well employ Data Mining methodologies.

Providing precious services at affordable costs is a major constraint encountered by the healthcare organizations (hospitals, medical centers). Valuable quality service denotes the accurate diagnosis of patients and providing efficient treatment. Poor clinical decisions may lead to disasters and hence are seldom entertained. Besides, it is essential that the hospitals decrease the cost of clinical test. Appropriate computer-based information and/or decision support systems can aid in achieving clinical tests at a reduced cost.

Owing to the accessibility of integrated information through enormous patient repositories, there is a swing in the insight of clinicians, patients and payers from qualitative visualization of clinical data to demanding a finer quantitative analysis of information with the assistance of all supporting clinical and imaging data. For example; now the physicians can evaluate diagnostic information of a variety of patients with identical conditions. Similarly, they can as well verify their findings with the conformity of physicians working on similar cases from all over the world.

Medical diagnosis is regarded as an important yet complicated task that needs to be executed accurately and efficiently. The automation of this system would be extremely advantageous. Regrettably all doctors do not possess expertise in every sub specialty and moreover there is a shortage of resource persons at certain places. Therefore, an automatic medical diagnosis system would probably be exceedingly beneficial by bringing all of them together.

Medical history data comprises of a number of tests essential to diagnose a particular disease. Clinical databases are elements of the domain where the procedure of data mining has develop into an inevitable aspect due to the gradual incline of medical and clinical research data. It is possible for the healthcare industries to gain advantage of Data mining by employing the same as an intelligent diagnostic tool. It is possible to acquire knowledge and information concerning a disease from the patient specific stored measurements as far as medical data is concerned. Therefore, data mining has developed into a vital domain in healthcare.

It is possible to predict the efficiency of medical treatments by building the data mining applications. Data mining can deliver an assessment of which courses of action prove effective by comparing and evaluating causes, symptoms, and courses of treatments. The real-life data mining applications are attractive since they provide data miners with varied set of problems, time and again. Working on heart disease patients databases is one kind of a real-life application. The detection of a disease from several factors or symptoms is a multi-layered problem and might lead to false assumptions frequently associated with erratic effects. Therefore it appears reasonable to try utilizing the knowledge and experience of several specialists collected in databases towards assisting the diagnosis process.

The researchers in the medical field identify and predict the diseases besides proffering effective care for patients with the aid of data mining techniques. The data mining techniques have been utilized by a wide variety of works in the literature to diagnose various diseases including: Diabetes, Hepatitis, Cancer, Heart diseases and the like. Information associated with the disease, prevailing in the form of electronic clinical records, treatment information, gene expressions, images and more; were employed in all these works.

The heart disease data warehouse consists of mixed attributes containing both the numerical and categorical data. These records are cleaned and filtered with the intention that the irrelevant data from the warehouse would be removed before mining process occurs. Subsequently the frequent patterns significant to heart disease diagnosis are mined from the extracted data. The significant weightage is calculated for each frequent pattern using the approach proposed. Then the patterns with significant weightage greater than a predefined threshold value are chosen. Afterwards, the Naïve Bayes is trained with the selected significant patterns in order to predict heart attack in an efficient manner.

3. LITERATURE SURVEY

3.1 EFFICIENT SUPANOVA KERNEL FOR HEART DISEASE DIAGNOSIS [7]

This paper presented a new heuristic for computing efficiently sparse kernel in SUPANOVA. On this data, 83.7% predictions were correct, exceeding the results obtained using the standard Support Vector Machine and equivalent kernels. It was assumed that there are N training data points in the form of vectors $xi=[xi1,xi\ 2,...,xi\ n]$, $i=1,2,...,N$. Each vector represents values of n features and has the corresponding output value, y1 they denoted the matrix containing these vectors (or training data points) as X and the vector of the corresponding output values as y.

The SUPANOVA method represents the solution as a sum of kernels that decompose functions of the order n into a sum of terms that are *unitary, 2- ary, ...,n-ary* order functions of the original arguments. Each function higher than first order uses a product of spline functions to represent its arguments. Hence, kernel function is replaced with a sum of kernels that measure similarity of argument vectors on a subset of features.

LIMITATION:

Elements with small magnitude are dropped to approximate the error.

3.2 INTELLIGENT HEART DISEASE PREDICTION SYSTEM USING CANFIS AND GENETIC ALGORITHM [8]

In Latha Parthiban et al. projected an approach on basis of CoActive Neuro-Fuzzy Inference System (CANFIS) for prediction of heart disease. The CANFIS model diagnosed the presence of disease by merging the neural network adaptive capabilities and the fuzzy logic qualitative approach and further integrating with genetic algorithm. On the basis of the training performances and classification

accuracies, the performances of the CANFIS model were evaluated. The CANFIS model is promising in the prediction of the heart disease as illustrated by the results.

3.3 Medical Data Classification Methods Based on Decision Tree and System Reconstruction Analysis [9]

This paper studies medical data classification methods, comparing decision tree and system reconstruction analysis as applied to heart disease medical data mining. The paper uses data from 1,723 coronary heart disease patients' cases. Each case contains about 71 attributes. The data come from a hospital clinic's observations and allow us to get a good classification of patients' status and behavior, from which we can determine the relationships among the factors. A data mining method to analyze the medical data is used. A system-reconstruction method is used to do data preprocessing and use decision-tree algorithms to do the classification. A comparison made on the classification correction rate on weighted and not weighted data, which is preprocessed by the system-reconstruction method. In this paper, first we introduce the system-reconstruction method and show how the coronary heart disease data are to be processed, and we discuss the theory and algorithms of decision trees, including, ID3, C4.5, CART, CHAID, and Exhausted-CHAID. We also apply these methods to medical data mining problems by designing some experiments to compare the correction rate, tree depth, and leaf number of weighted and not weighted data gotten by decision tree.

Limitation:

From the data – it concludes that data weighted by the system-reconstruction method can get a higher correction rate but will have little effect on the leaf number and tree depth of the decision tree.

3.4 Predicting Survival Causes After Out of Hospital Cardiac Arrest Using Data Mining Method [10].

The main objective of their work was to represent the relationship between variables to determine which variables were the most important for patient survival, with data mining methods. Other goals were firstly to build a decision tree to evaluate which were main factor

produce a real patient profile to determine they the best practice for these patients. One interest of the Bayesian network is the possibility of monitoring the propagation of the impact of an event in the network on other variables or events.

Analysis performed in three steps:

- Step 1: Learning step
- Step 2: Analysis of associations
- Step 3: Inference and prediction

For the learning step, they use the Taboo Order method to build the network. The graph of the variables showed that the probability of being alive after heart failure is directly associated with five variables: age sex, the initial cardiac rhythm, the origin of heart failure, and the type of specialized resuscitation employed. By monitoring the main node (Alive/death), it can infer conclusions about the patient profile. The patient profile observed after cardiac arrest. Forty nine point one six percent of the patients died, 6.04 % left the hospital and the 44.80 % represent non cardiac arrest or cardiac arrest where resuscitation was not possible. For each variable directly related with this node, the rhythm was equal to 1 asystole in 62.32 % of the cases, equal to 2 ventricular tachy arrhythmias FV/TV in 31.03 % and equals to 3 pulses less electrical activity in 6.65 % of the cases. Nineteen point six four percent of the patients were younger than 46 years and 87.37 % of the patients were not hospitalised in intensive care. The hidden Markov layer demonstrated that one node appears to be highly related to the main node.

LIMITATION:

The main limit of these tools is the necessity to have enough data to find regularity in the relationships.

3.5 A MULTI - CLASS HEARTBEAT CLASSIFIER EMPLOYING HYBRID FUZZY - NEURAL NETWORK [11]

Electrocardiogram (ECG) diagnosis is used to study the condition of the heart and

proposed a novel strategy for automatic heartbeat classification to palliate the mentioned problems. Ten types of heartbeats considered for automatic classification are Atrial Premature Contraction (APC), Fusion(F), Left Bundle Branch Block type I and type II (LBBBB 1& LBBBB II), Normal(N), Paced(p), Right Bundle Branch Block type I and type II (RBBBB I & RBBBB II) , Premature Ventricular Contraction type I. and type II (PVC I & PVC 11). Fuzzy C - Means clustering (FCM) is employed for feature extraction of the individual ECG cycles and these extracted features are then used for training multilayer perceptron. A detailed study undertaken to find the optimum number of clusters and optimal MLP configuration with the metric of overall percentage classification accuracy. The best FCM-MLP topology exhibited overall classification accuracy of 98.25°. This network was tested for performance in presence of additive white Gaussian noise and was found to be very robust. For comparison, a well-known method of Principal Component Analysis (PCA) was also experimented with. FCM-MLP performs better than PCA-MLP in classifying the correct heartbeats. The proposed scheme exhibited an average accuracy of more than 98%.

3.6 Combination Data Mining Methods with New Medical Data to Predicting Outcome of Coronary Heart Disease [12]

The prediction of survival of Coronary Heart Disease (CHD) has been a challenging research problem for medical society. The goal of this paper is to develop data mining algorithms for predicting survival of CHD patients based on 1000 cases .We carry out a clinical observation and a 6-month follow up to include 1000 CHD cases. The survival information of each case is obtained via follow up. Based on the data, we employed three popular data mining algorithms to develop the prediction models using the 502 cases. We also used 10-fold cross-validation methods 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 tress 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 us with an insight into the relative prediction ability of different data.

4. OBJECTIVE

The objectives of this system are

- To preprocess the dataset to make the mining process easier.
- To determine membership values for the attributes to train the model.
- The critical rules are determined which are used to train the model.
- To prove that the system produces consistent result.
- The system is designed to analyze the patient data and predict the heart disease type of that patient.
- The Heart disease data set is used for the system

5. SYSTEM METHODOLOGY

Knowledge Discovery in Databases (KDD) means the application of non-trivial procedures for identifying effective, coherent, potentially useful, and previously unknown patterns in large databases. The KDD process generally consists of the following three phases [13].

(1) **Pre-processing**: This consists of all the actions taken before the actual data analysis process starts. It may be performed on the data for the following reasons: solving data problems that may prevent us from performing any type of analysis on the data, understanding the nature of the data, performing a more meaningful data analysis, and extracting more meaningful knowledge from a given set of data.

(2) **Data-mining**: This involves applying specific algorithms for extracting patterns or rules from data sets in a particular representation.

(3) **Post-processing**: This translates discovered patterns into forms acceptable for human beings. It may also make possible visualization of extracted patterns.

The heart disease data warehouse contains the screening clinical data of heart patients. Initially, the data warehouse is preprocessed to make the mining process more efficient.

The frequent patterns are mined from the data, relevant to heart attack, using the Naïve Bayes algorithm. The significant weightage is calculated for all frequent patterns with the aid of the approach proposed. The frequent patterns with significant weightage greater than a predefined threshold are chosen. These chosen significant patterns can be used in the design and development of heart attack prediction system.

The system is divided into four major modules

- Analyzing the Data set
- Designing the IHDPS
- Naive Bayes Algorithm Implementation

5.1 SYSTEM ARCHITECTURE

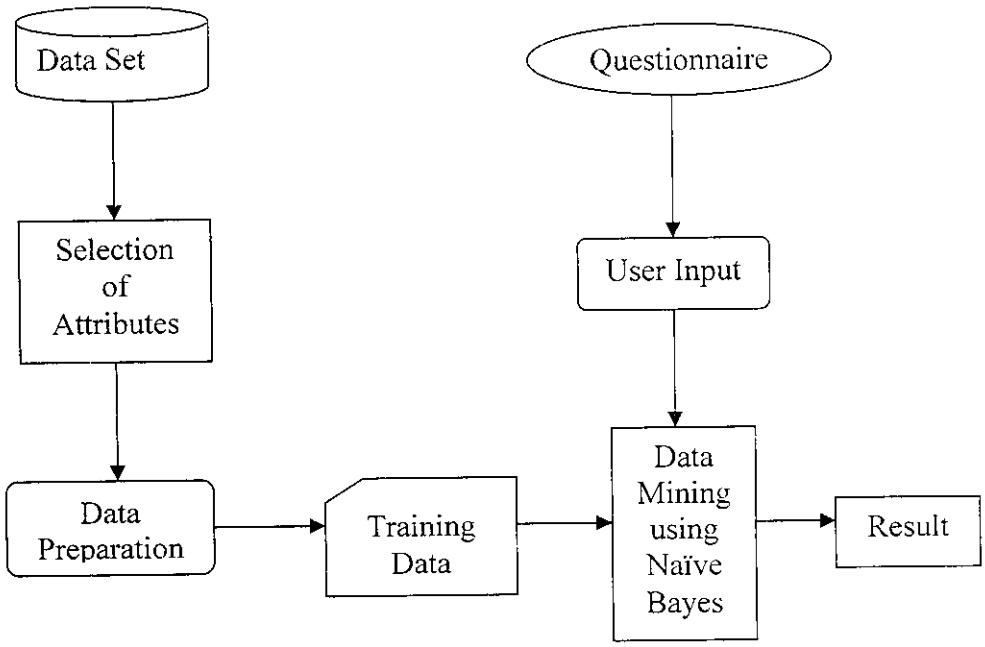


FIGURE 5.1: SYSTEM ARCHITECTURE

5.2 DATA PREPROCESSING [14]

Cleaning and filtering of the data might be necessarily carried out with respect to the data and data mining algorithm employed so as to avoid the creation of deceptive or inappropriate rules or patterns. The actions comprised in the pre-processing of a data set are the removal of duplicate records, normalizing the values used to represent information in the database, accounting for missing data points and removing unneeded data fields. In order for making the data appropriate for the mining process it needs to be transformed.

The raw data is changed into data sets with a few appropriate characteristics. Moreover it might be essential to combine the data so as to reduce the number of data sets

algorithm. In our approach, the heart disease data warehouse is refined by removing duplicate records and supplying missing values. Furthermore it is also transformed to a form appropriate for clustering.

5.3 HEART DISEASE PREDICTION SYSTEM USING NAIVE BAYES [15]

Naive Bayes or Bayes' Rule is the basis for many machine-learning and data mining methods. The rule (algorithm) is used to create models with predictive capabilities. It provides new ways of exploring and understanding data. It learns from the "evidence" by calculating the correlation between the target (i.e., dependent) and other (i.e., independent) variables.

The N*aive Bayesian classifier* uses the Naive Bayes formula to calculate the probability of each class given the values of all the attributes and assuming the conditional independence of the attributes. The attributes are usually defined by a human (especially in medical data), and are therefore relatively independent, as humans tend to think linearly. This is the reason why the Naive Bayesian formula often performs well on real-world problems.

Suppose if the data consist of fruits, described by their color and shape. Bayesian classifiers operate by saying "If there is a fruit that is red and round? Which type of fruit is it most likely to be? based on the observed data sample".

A difficulty arises when there are more than a few variables and classes –there is a requirement of enormous number of observations (records) to estimate these probabilities.

Naive Bayes classification gets around this problem by not requiring that have lots of observations for each possible combination of the variables. Rather, the variables are assumed to be independent of one another and, therefore the probability that a fruit that is red, round, firm, 3" in diameter, etc. will be an apple can be calculated from the independent probabilities that a fruit is red, that it is round, that it is firm, that it is 3" in diameter, etc.

In other words, Naïve Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variable. This assumption is called class

conditional independence. It is made to simplify the computation and in this sense considered to be "Naïve".

Studies comparing classification algorithms have found the Naïve Bayesian classifier to be comparable in performance with classification trees and with neural network classifiers. This algorithm also exhibits high accuracy and speed when applied to large databases.

5.3.1 BAYES THEOREM

Let X be the data record (case) whose class label is unknown. Let H be some hypothesis, such as "data record X belongs to a specified class C." For classification, to determine P (H|X) -- the probability that the hypothesis H holds, given the observed data record X.

P (H|X) is the posterior probability of H conditioned on X. For example, the probability that a fruit is an apple, given the condition that it is red and round. In contrast, P (H) is the prior probability, or apriori probability, of H. In this example P (H) is the probability that any given data record is an apple, regardless of how the data record looks. The posterior probability, P (H|X), is based on more information (such as background knowledge) than the prior probability, P (H), which is independent of X.

Similarly, P (X|H) is posterior probability of X conditioned on H. That is, it is the probability that X is red and round given that we know that it is true that X is an apple. P(X) is the prior probability of X, i.e., it is the probability that a data record from the set of fruits is red and round. Bayes theorem is useful in that it provides a way of calculating the posterior probability, P H|X), from P (H), P(X), and P (X|H). Bayes theorem is

$$P(H|X) = P(X|H)\, P(H) / P(X)$$

$$P(C|A_1 \ldots A_{|\mathcal{A}|}) = P(C) \prod_{A \in \mathcal{A}} Q_A \quad \text{where} \quad Q_A = \frac{P(A|C)}{P(A)} = \frac{P(C|A)}{P(C)}$$

A new instance is classified into the class with maximum calculated probability. For

$$P(C) = \frac{N(C)+1}{N+N(C)}$$

For estimations of conditional probabilities $P(C|A)$ the m-estimate was used:

$$P(C|A) = \frac{N(CA)+m \times Pa(C)}{N(A)+m} = \frac{N(CA)}{N(A)+m} + \frac{m \times Pa(C)}{N(A)+m}$$

The parameter m balances between the contributions of the relative frequency $N(CA)/N(A)$ and the prior probability $Pa(C)$. Both Laplace's law of succession and m-estimate are very useful, especially when estimating probabilities from small datasets. In our experiments, the parameter m was set to 2. This setting is usually used as default and, empirically, gives satisfactory results although with tuning better results might be expected.

5.3.2 ABOUT NAIVE BAYES

The Naive Bayes algorithm is based on conditional probabilities. It uses Bayes' Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values in the historical data.

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. If B represents the dependent event and A represents the prior event, Bayes' theorem can be stated as follows.

BAYES' THEOREM:

Prob (B given A) = Prob (A and B)/Prob (A)

To calculate the probability of B given A, the algorithm counts the number of cases where A and B occur together and divides it by the number of cases where A occurs alone.

5.3.3 ADVANTAGES OF NAIVE BAYES

The Naive Bayes algorithm affords fast, highly scalable model building and scoring. It scales linearly with the number of predictors and rows. The build process for Naive Bayes is parallelized. (Scoring can be parallelized irrespective of the algorithm.)

Naive Bayes can be used for both binary and multiclass classification problems. They have also exhibited high accuracy and speed when applied to large databases.

5.3.4 DATA PREPARATION FOR NAIVE BAYES

Automatic Data Preparation performs supervised binning for Naive Bayes. Supervised binning uses decision trees to create the optimal bin boundaries. Both categorical and numerical attributes are binned.

Naive Bayes handles missing values naturally as missing at random. The algorithm replaces sparse numerical data with zeros and sparse categorical data with zero vectors. Missing values in nested columns are interpreted as sparse. Missing values in columns with simple data types are interpreted as missing at random.

For the management of data preparation, Naive Bayes usually requires binning. Naive Bayes relies on counting techniques to calculate probabilities. Columns should be binned to reduce the cardinality as appropriate. Numerical data can be binned into ranges of values (for example, low, medium, and high), and categorical data can be binned into meta-classes (for example, regions instead of cities). Equi-width binning is not recommended, since outliers will cause most of the data to concentrate in a few bins, sometimes a single bin. As a result, the discriminating power of the algorithms will be significantly reduced.

5.4 METHODOLOGY [1]

IHDPS (INTELLIGENT HEART DISEASE PREDICTION SYSTEM) uses the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology to build the mining models. It consists of six major phases.

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment

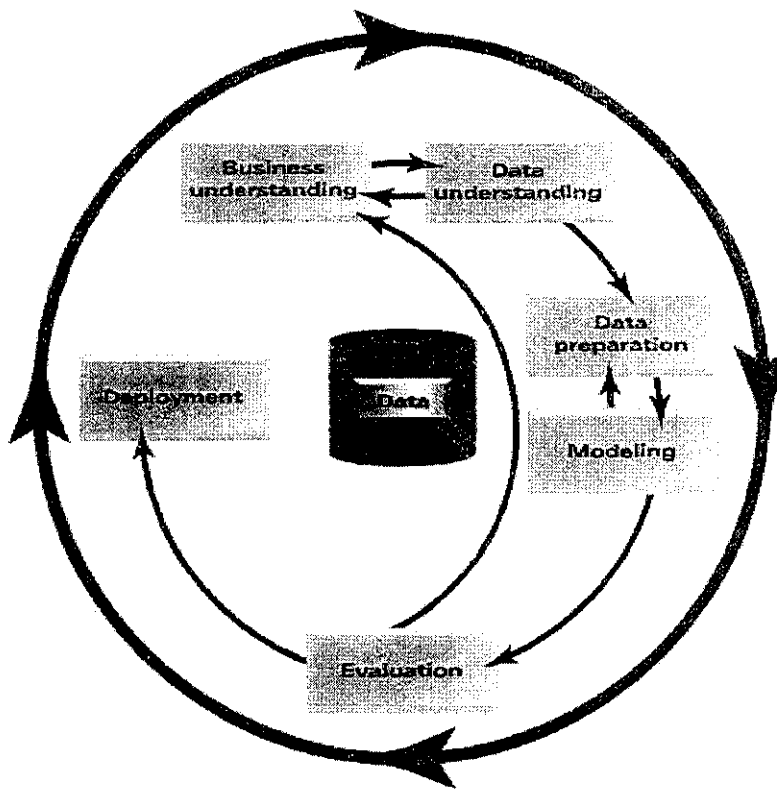


FIGURE 5.2: CRISP-DM METHODOLOGY

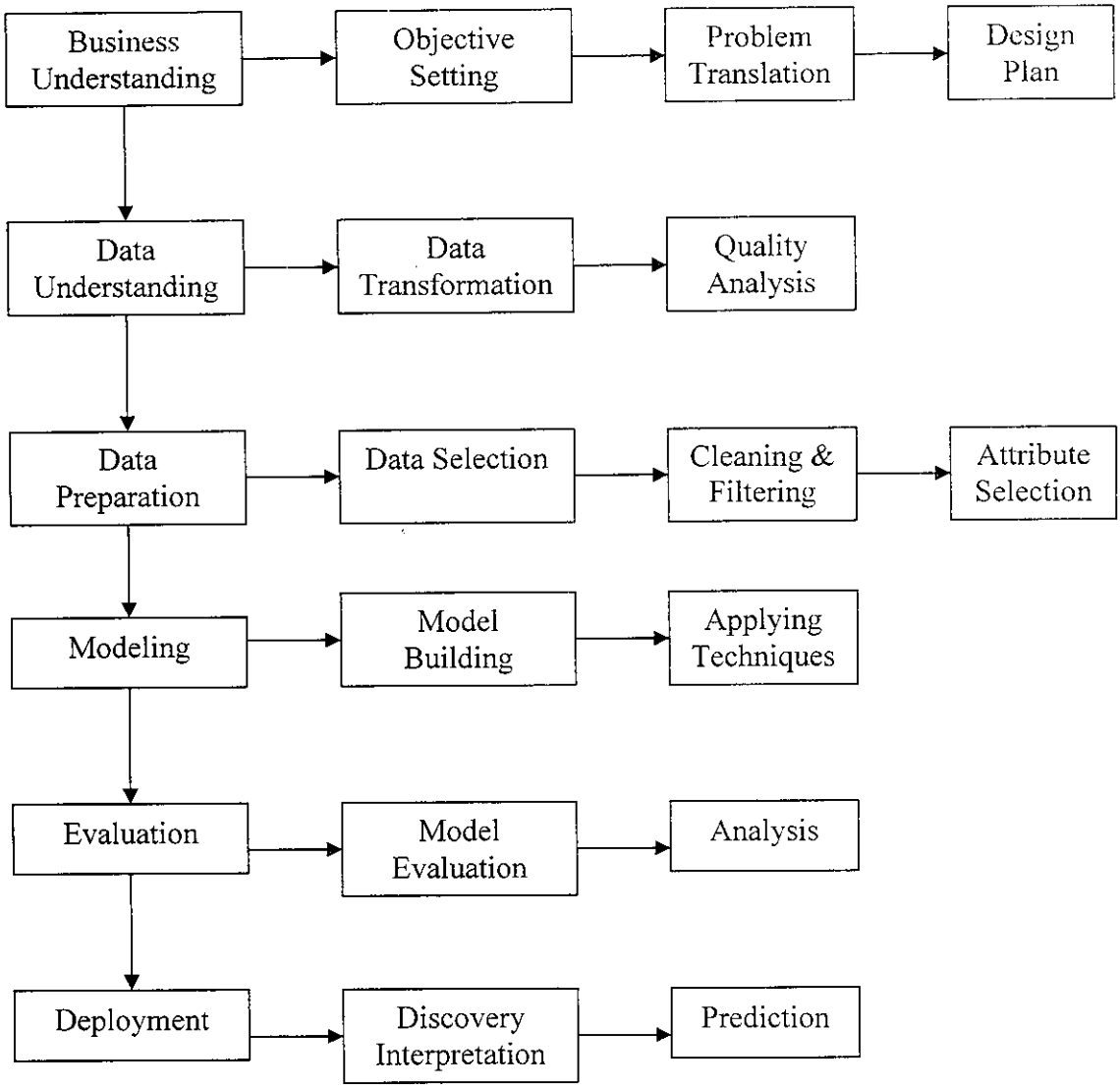


FIGURE 5.3 – STEPS IN CRISP METHODOLOGY

Business understanding phase focuses on understanding the objectives and requirements from a business perspective, converting this knowledge into a data mining problem definition, and designing a preliminary plan to achieve the objectives.

Data understanding phase uses the raw data and proceeds to understand the data, identify its quality, gain preliminary insights, and detect interesting subsets to form hypotheses for hidden information.

Data preparation phase constructs the final dataset that will be fed into the modeling tools. This includes table, record, and attribute selection as well as data cleaning and transformation.

The modeling phase selects and applies various techniques, and calibrates their parameters to optimal values.

The evaluation phase evaluates the model to ensure that it achieves the business objectives.

The deployment phase specifies the tasks that are needed to use the models.

5.5 DATA SOURCE [1]

A total of 909 records with 15 medical attributes (factors) were obtained from the Cleveland Heart Disease database. Figure 4.2 lists the attributes. The records were split equally into two datasets: training dataset (455 records) and testing dataset (454 records).To avoid bias; the records for each set were selected randomly. For the sake of consistency, only categorical attributes were used for all the three models. All the non-categorical medical attributes were transformed to categorical data.

The attribute "Diagnosis" was identified as the predictable attribute with value "1" for patients with heart disease and value "0" for patients with no heart disease. The attribute "PatientID" was used as the key; the rest are input attributes. It is assumed that problems such as missing data, inconsistent data, and duplicate data have all been resolved.

Predictable attribute

1. Diagnosis (value 0: < 50% diameter narrowing (no heart disease); value 1: > 50% diameter narrowing (has heart disease))

Key attribute

1. PatientID - Patient's identification number

Input attributes

1. Sex (value 1: Male; value 0: Female)
2. Chest Pain Type (value 1: typical type 1 angina, value 2: typical type angina, value 3: non-anginal pain; value 4: asymptomatic)
3. Fasting Blood Sugar (value 1: > 120 mg/dl; value 0: < 120 mg/dl)
4. Resteeg – resting electrographic results (value 0: normal; value 1: having ST-T wave abnormality; value 2: showing probable or definite left ventricular hypertrophy)
5. Exang – exercise induced angina (value 1: yes; value 0: no)
6. Slope – the slope of the peak exercise ST segment (value 1: unsloping; value 2: flat; value 3: downsloping)
7. CA – number of major vessels colored by flourosopy (value 0 – 3)
8. Thal (value 3: normal; value 6: fixed defect; value 7: reversible defect)
9. Trest Blood Pressure (mm Hg on admission to the hospital)
10. Serum Cholesterol (mg/dl)
11. Thalach – maximum heart rate achieved
12. Oldpeak – ST depression induced by exercise relative to rest
13. Age in Year

FIGURE 5.4 – DESCRIPTION OF ATTRIBUTES

6. SIMULATION RESULTS

6.1 EXPERIMENTAL RESULTS

This chapter is devoted to explain the simulation results because results of the project promise the effective operation of the model that has been designed. The classification of the patterns is a multi-stage system with data preprocessing module, parameter extracting and membership value calculation module, Rule generation module, weight calculation module and classification module which are clearly analyzed in this chapter.

The pattern classification system is tested on Intel Core 2 Duo (3846 MHz clock speed), 3 GB RAM desktop PC. For testing the validity of the proposed model the Cleveland database is selected from the UCI Machine Learning Repository [4]. The dataset is preprocessed for elimination of unknown values. The preprocessing modules consisting of various steps like cleaning, discretization attribute reduction and architecting. The proposed method is a Naïve Bayes algorithm.

6.1.1 DATA PREPROCESSING

Real world data is incomplete, inconsistent and noisy. Data preprocessing techniques can improve the quality of the data thereby helping to improve the accuracy and efficiency of the subsequent mining process. For getting quality results we need to preprocess the dataset first. A summary of the datasets taken from the UCI Machine learning repository is shown in Table 6.1. The database consisting of values for 76 attributes of a patient from that only the most important 15 attributes are selected.

Datasets	Number of instances	Number of attributes	Number of attributes chosen for classification	Number of classes	Missing values	Area
Cleveland	303	76	15	5	4	Heart Disease

TABLE 6.1 DATABASE SUMMARY

The preprocessing is done to convert the available data into the form that is required for the classification process. The datasets are preprocessed by available methods namely data cleaning, data discretization, data architecting and data reduction. From the Database selected to construct the decision making table are shown in Table 6.2.

Age	Sex	CPT	BP	Chol	FBS	Resteeg	Thalach	Exang	Old peak	Slope	CA	Thal	Height	Weight	Class	Dtype
63	Male	Angina	145	233	True	Hyp	150	Fal	2.3	Down	0.0	Fix	160	50	Buff	H
67	Male	Asympt	120	229	Fal	Hyp	129	True	2.6	Flat	2.0	Rev	170	45	Sick	S1
67	Male	Asympt	160	286	Fal	Hyp	108	True	1.5	Flat	3.0	Norm	150	70	Sick	S2
62	Fem	Asympt	140	268	Fal	Hyp	160	Fal	3.6	Down	2.0	Norm	172	85	Sick	S3

TABLE 6.2 CLEVELAND DATABASE OF HEART DISEASE DATA

6.1.1.1 DATA CLEANING

Among the data of 303 instances, the instance numbers 167,193,288 and 303 have null values for attribute 'CA'. The following are the various methods to fill null values.

1. **Ignore the tuple**

This is usually done when the class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.

2. **Fill in the missing value manually**

This approach is time consuming and may not be feasible given a large dataset with many missing value.

3. **Use global constant to fill in the missing value**

Replace all missing attribute values by the same constant such as a label like "unknown" or "-∞".

4. **Use attribute mean to fill in the missing value**

The mean value of the particular attribute having missing value can be calculated. Use this value to replace the missing value for that attribute.

5. **Use the most probable value to fill in the missing value**

This is determined with Bayesian formula.

Among these various methods available for filling the null values the last two methods are suitable for medical field and these two methods yield same result as shown in Table 6.3.

TRAINING SAMPLES	Number of CA values missing	Mean fill	Inference based (Bayesian formula)
153	2	0	.004
203	3	0	.002
303	4	0	.002

TABLE 6.3: FILLING MISSING VALUES FOR CA

The final preprocessed results are shown in Table 6.4 with sample records.

Age	Sex	CPT	BP	Chol	FBS	Resteeg	Thalach	Exang	Old peak	Slope	CA	Thal	Height	Weight	Class
63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.0	6.0	160	50	0
67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	150	70	1
37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	165	58	0
62.0	0.0	4.0	140.0	268.0	0.0	2.0	160.0	0.0	3.6	3.0	2.0	3.0	168	60	1

TABLE 6.4: PREPROCESSED DATASET

6.2 PERFORMANCE EVALUATION

Performance of the Naïve Bayes model with various membership values are found from confusion matrix and dataset are partitioned in various percentage for training and testing the Naïve Bayes. The actual classification are shown in table 6.14.The results of NB for different training and testing sets shown in Table 6.15.

Type	Data Set	Type H	Type S1	Type S2	Type S3	Type S4
Actual	303	164	55	36	35	13
Derived	271	156	49	25	32	9

TABLE 6.5: CLASSIFICATION OF HEART DISEASE

	True	False
True	**153**	**15**
False	**8**	**127**

TABLE 6.6: CONFUSION MATRIX OF PREPROCESSED DATASET

ACCURACY:

The accuracy is the proportion of true results (both true positives and true negatives) in the population.

$$\text{accuracy} = \frac{\text{number of true positives} + \text{number of true negatives}}{\text{numbers of true positives} + \text{false positives} + \text{false negatives} + \text{true negatives}}$$

ACCURACY = (153 + 127) / (153+15+8+127)

ACCURACY = 0.92409241

ACCURACY PERCENTAGE = 92.41%

ERROR:

ERROR = 1 – ACCURACY

ERROR = 1 - 0.92409241 = 0.0759076

ERROR PERCENTAGE = 7.6 %

Training Samples	Testing Samples	Accuracy %	Training Time in sec	Testing Time in sec
30 %	70%	85	0.122	0.31
50%	50%	88	0.225	0.29
60%	40%	90	0.229	0.22
80%	20%	92	0.357	0.14

TABLE 6.7: ACCURACY AND TIMING FOR DIFFERENT TRAINING DATASET

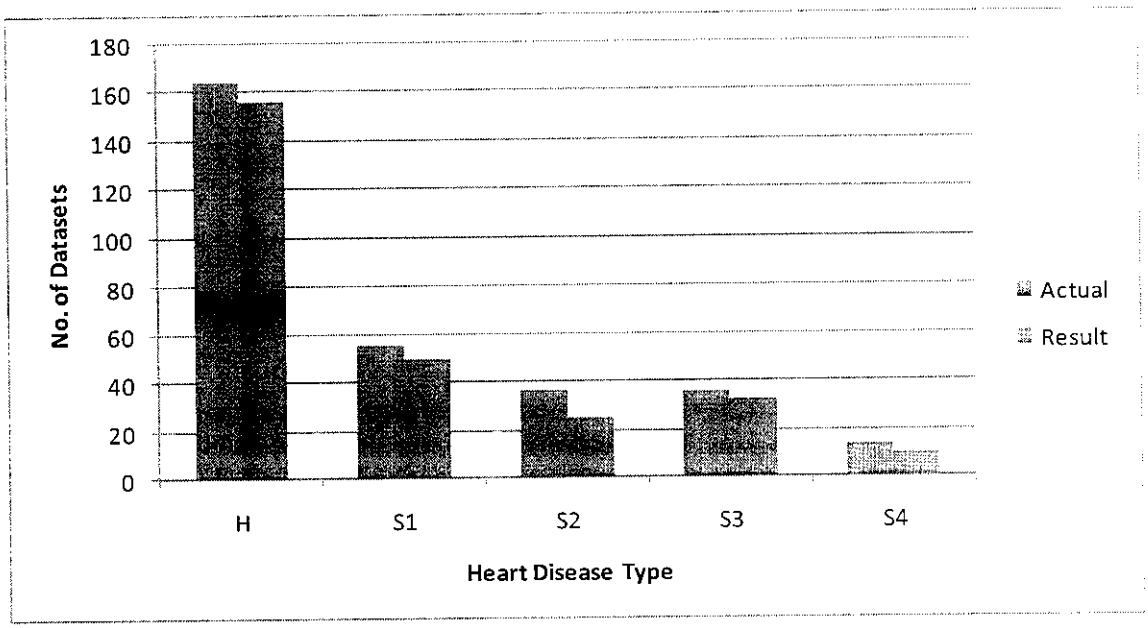


FIGURE 6.1: COMPARISON OF ACTUAL CASES VS. OBTAINED CASES THROUGH NAÏVE BAYES

6.3 CONCLUSION

In this project work, classification of various heart diseases by Naïve Bayes was proposed. Experiments conducted on the heart disease dataset illustrated that the proposed approach produces meaningful results and has reasonable efficiency. The results of the proposed model are consistent and hence encouraging.

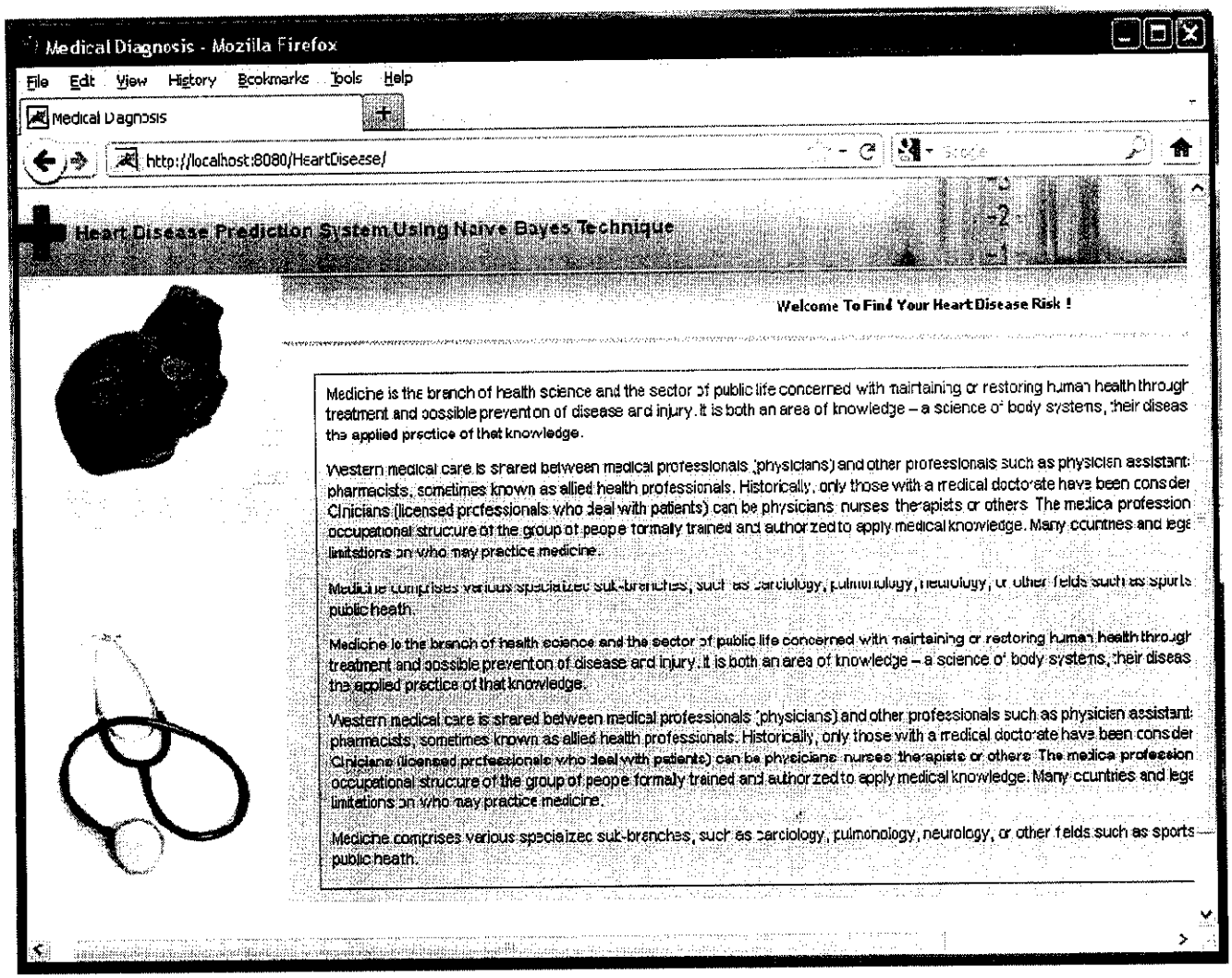
6.4 FUTURE SCOPE

This model can be further trained with more data set to get cent percent result and can be further developed for predicting unknown patient's record, further user interface can be developed for easy access. IHDPS can be further enhanced and expanded. For example, it can incorporate other medical attributes besides the 15 listed. It can also incorporate other data mining techniques, e.g., Time Series, Clustering and Association Rules. Continuous data can also be used instead of just categorical data. Another area is to use Text Mining to mine the vast amount of unstructured data available in healthcare databases. Another challenge would be to integrate data mining and text mining.

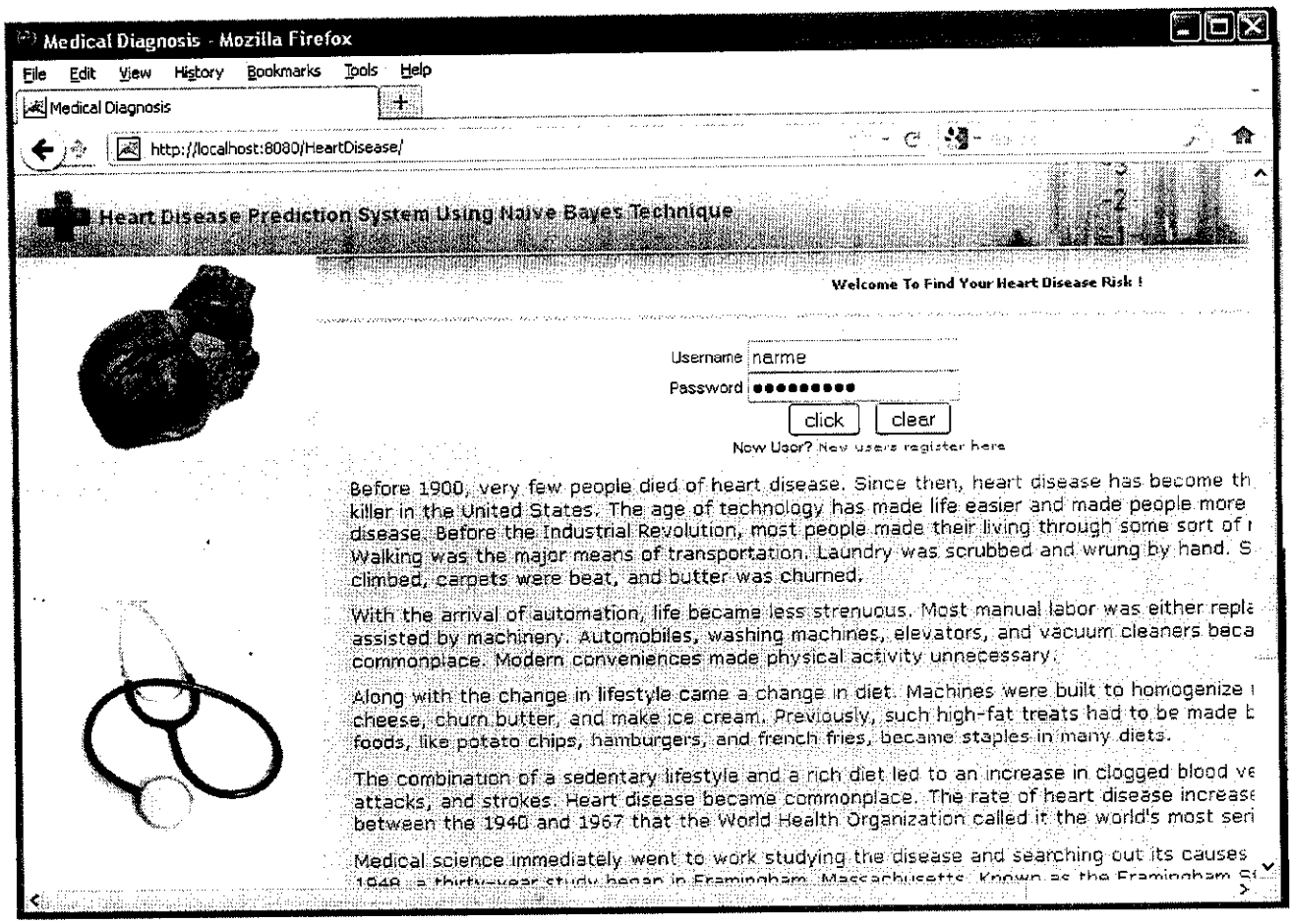
APPENDIX – I

SCREEEN SHOTS

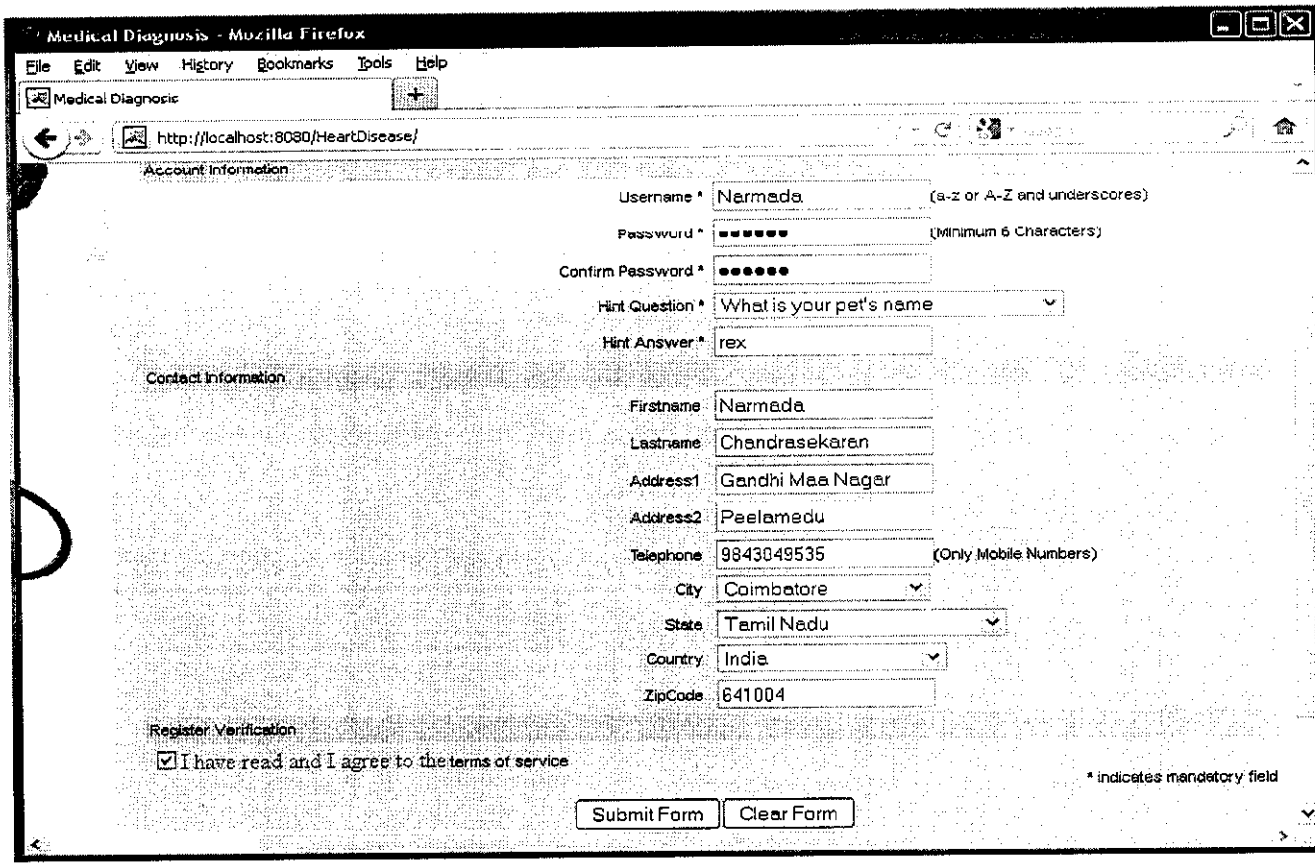
HOME PAGE OF THE INTELLIGENT HEART DISEASE PREDICTION SYSTEM



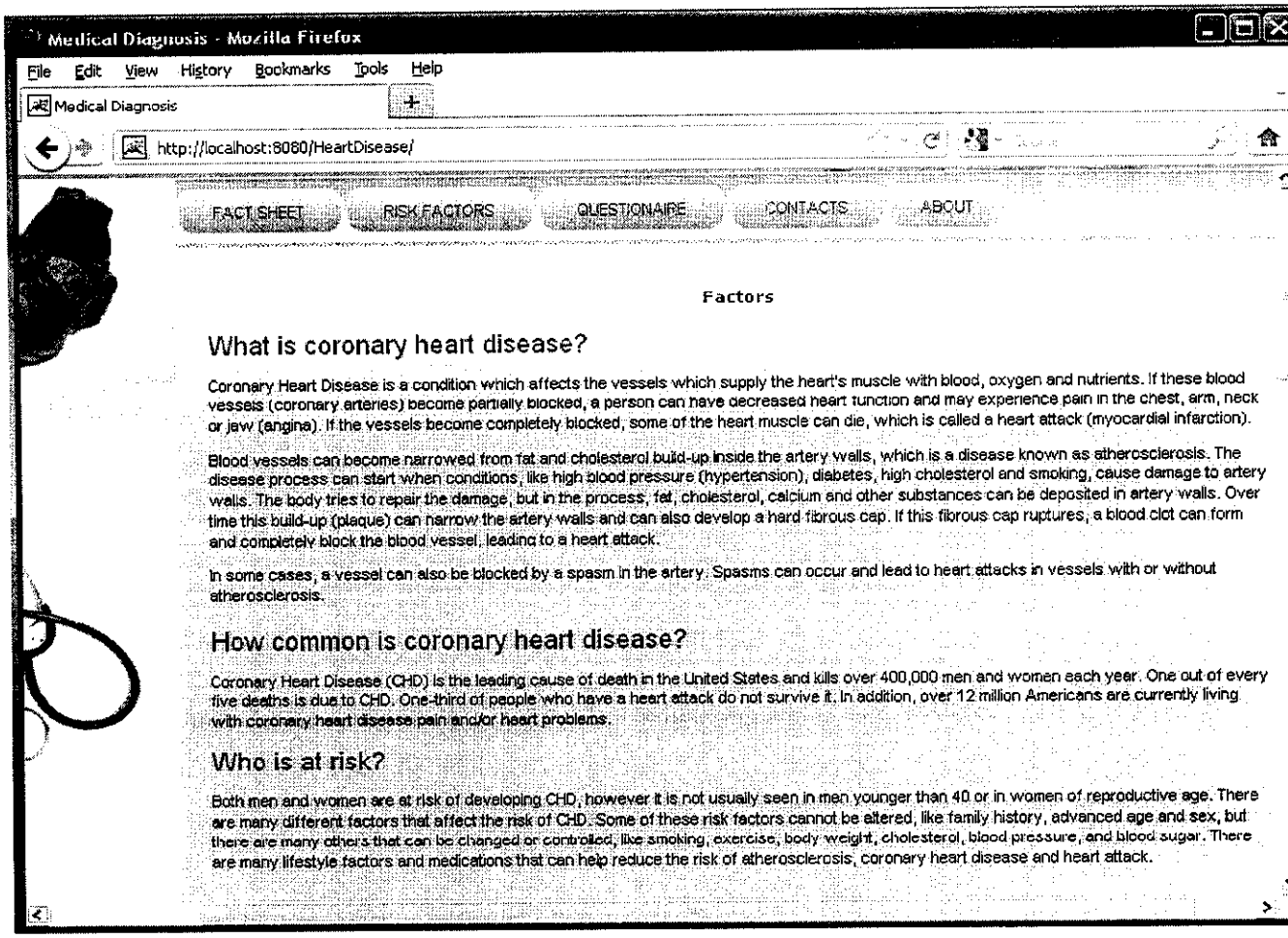
LOGIN PAGE OF THE IHDPS



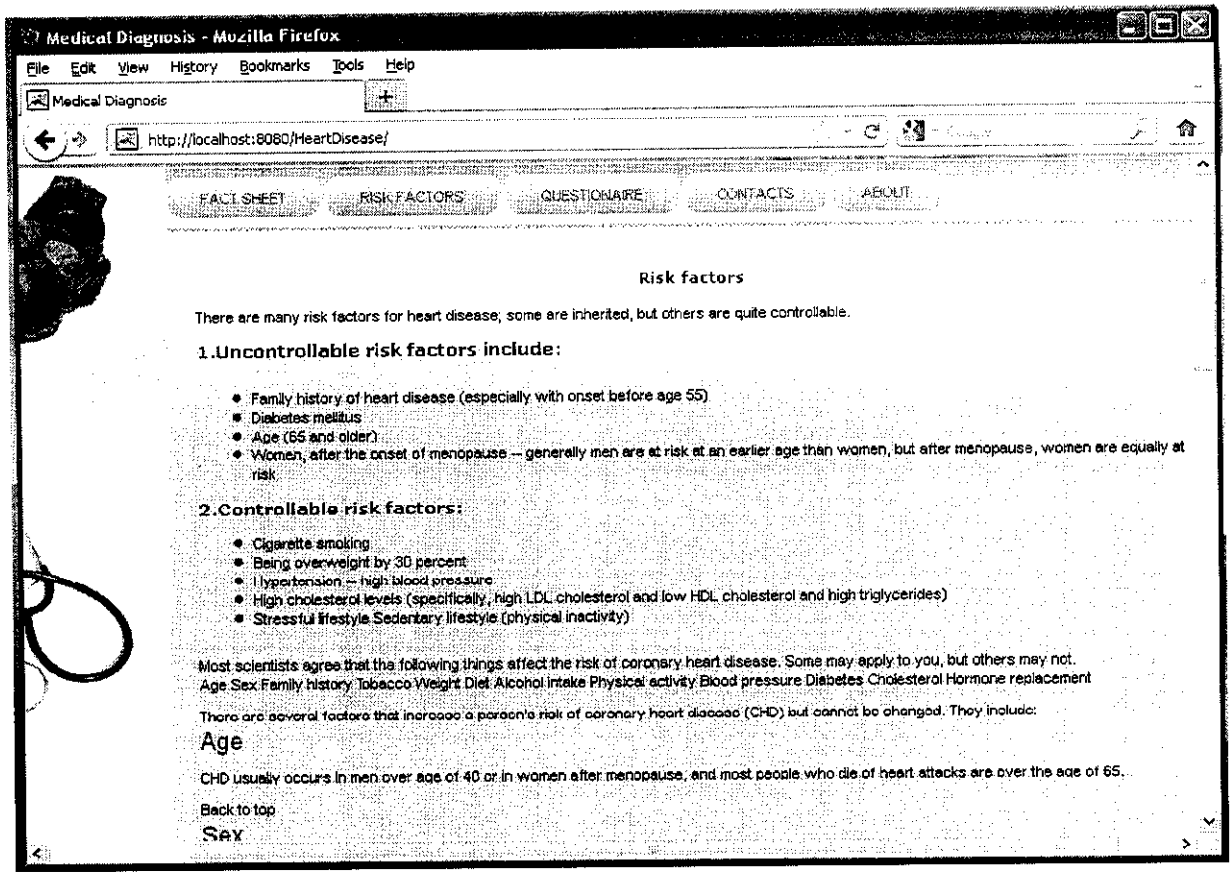
NEW USER REGISTRATION FORM OF IHDPS



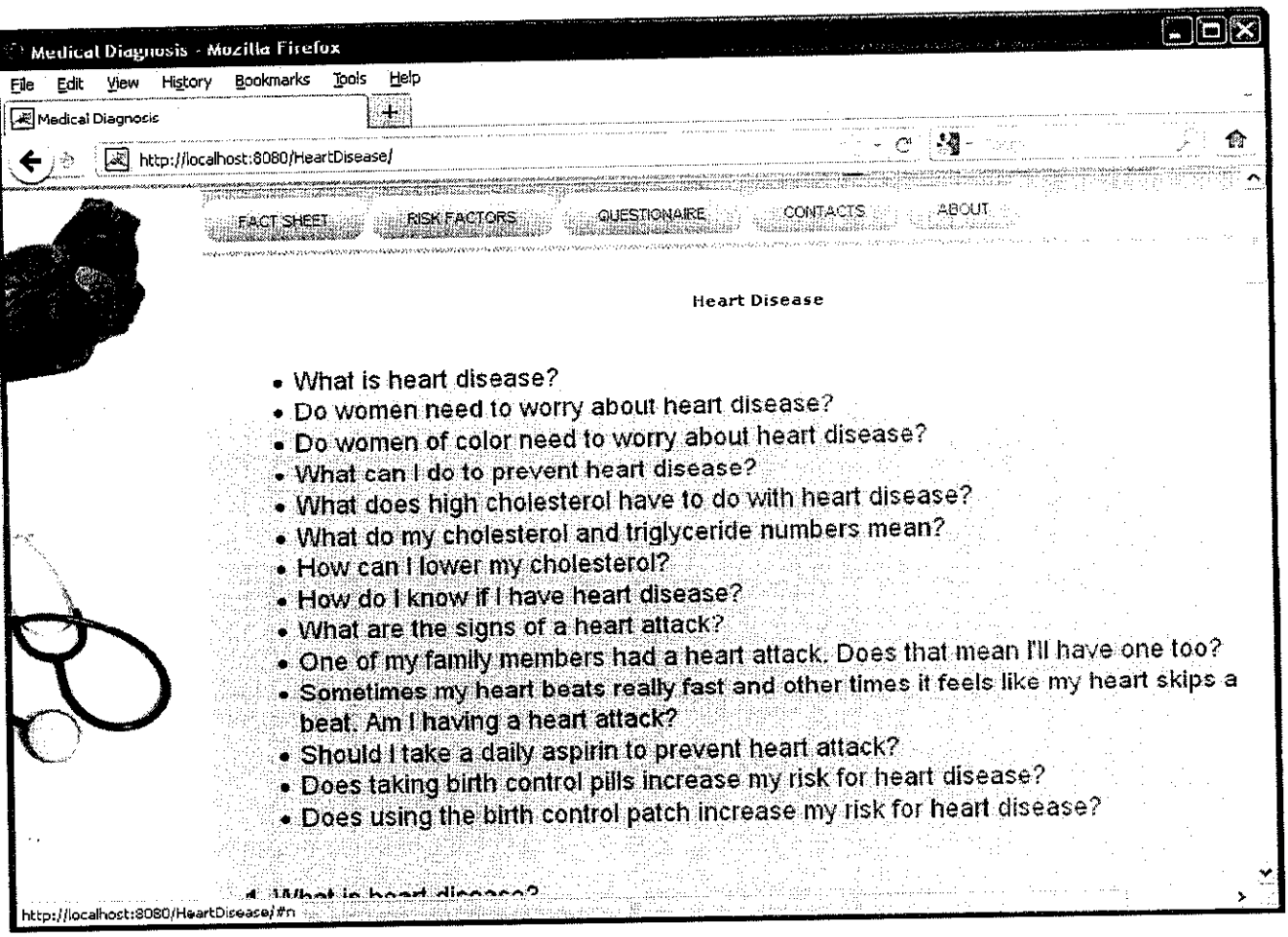
FACT SHEET OF HEART DISEASES



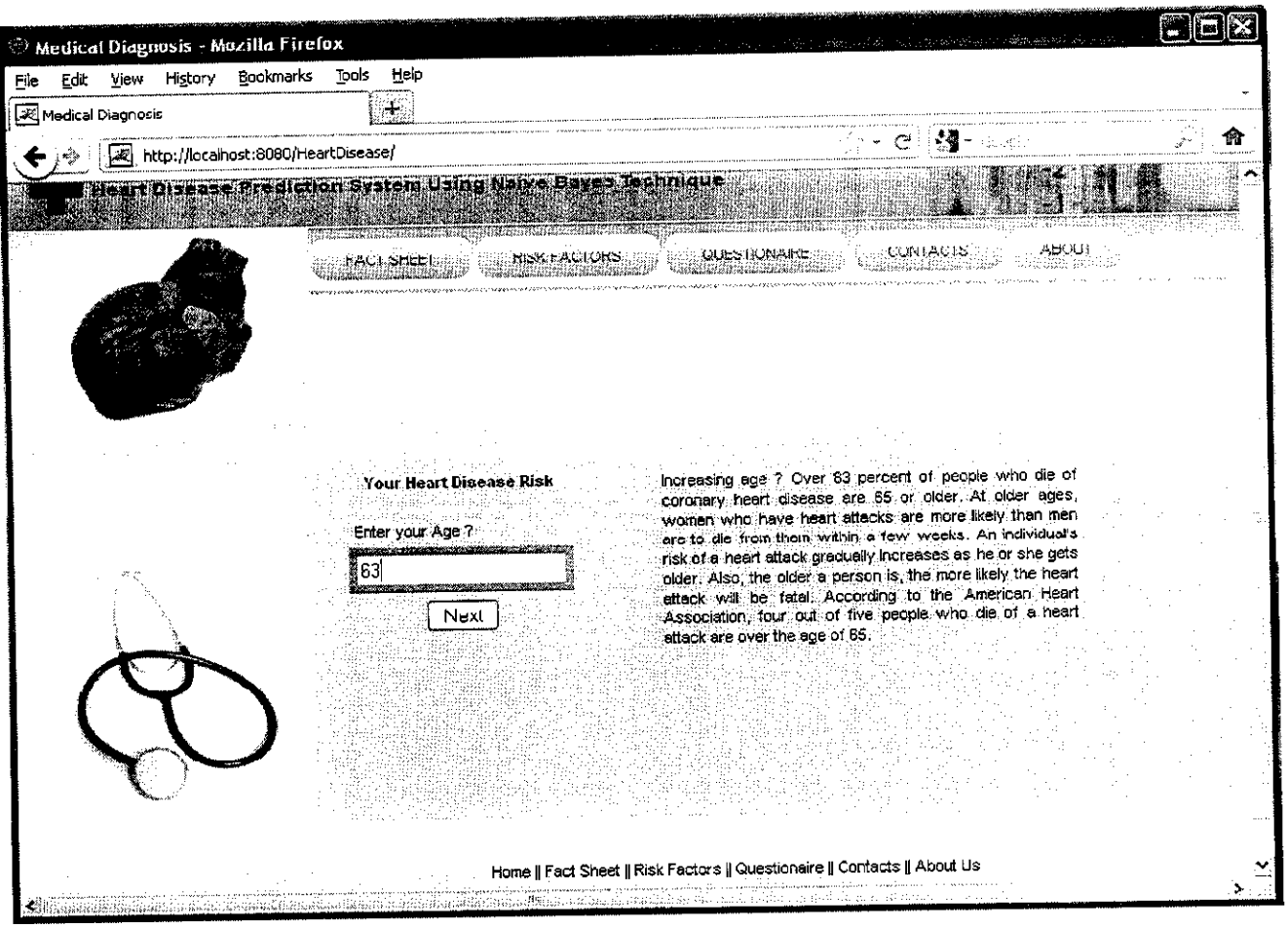
RISK FACTORS ABOUT THE HEART DISEASE

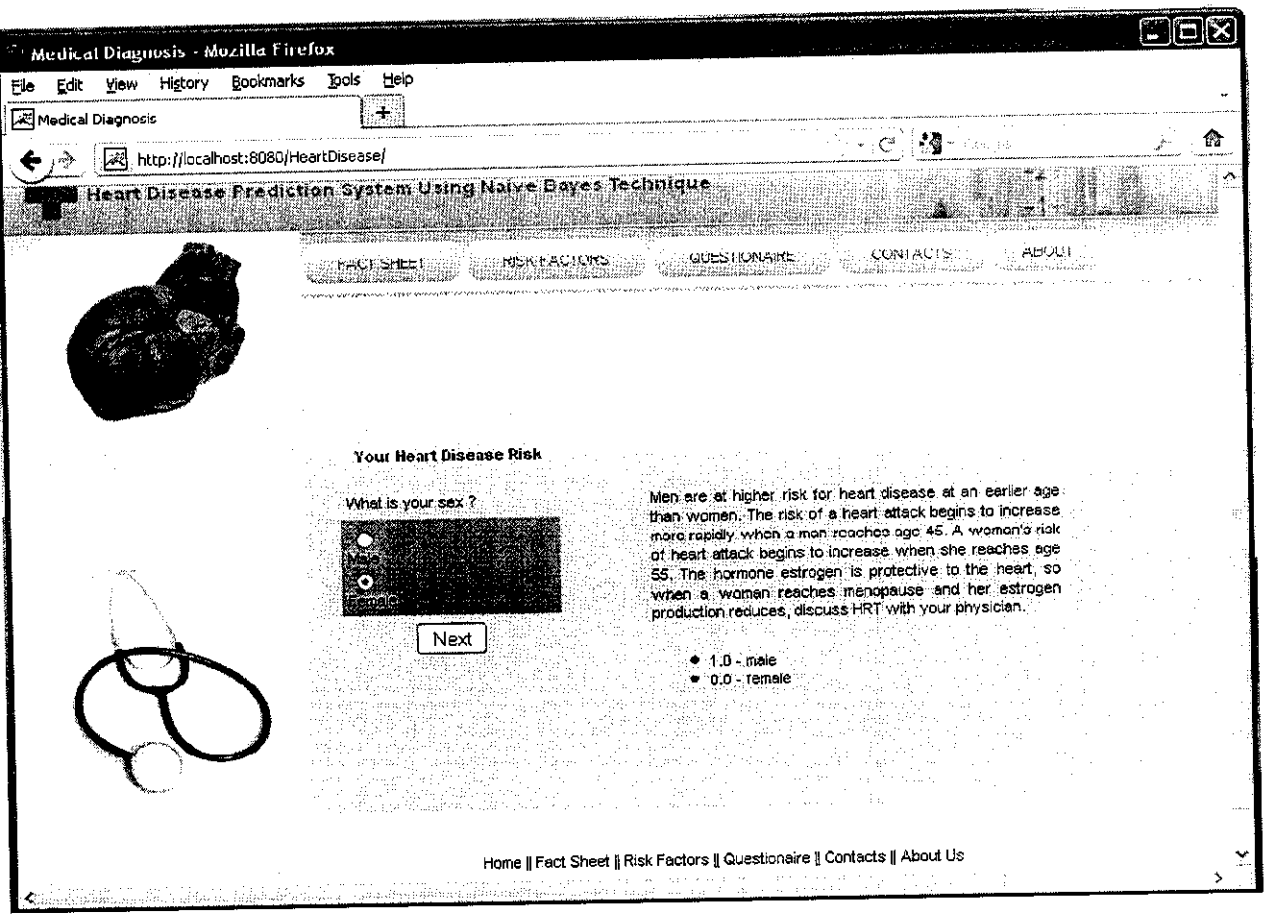


FAQ ABOUT HEART DISEASE

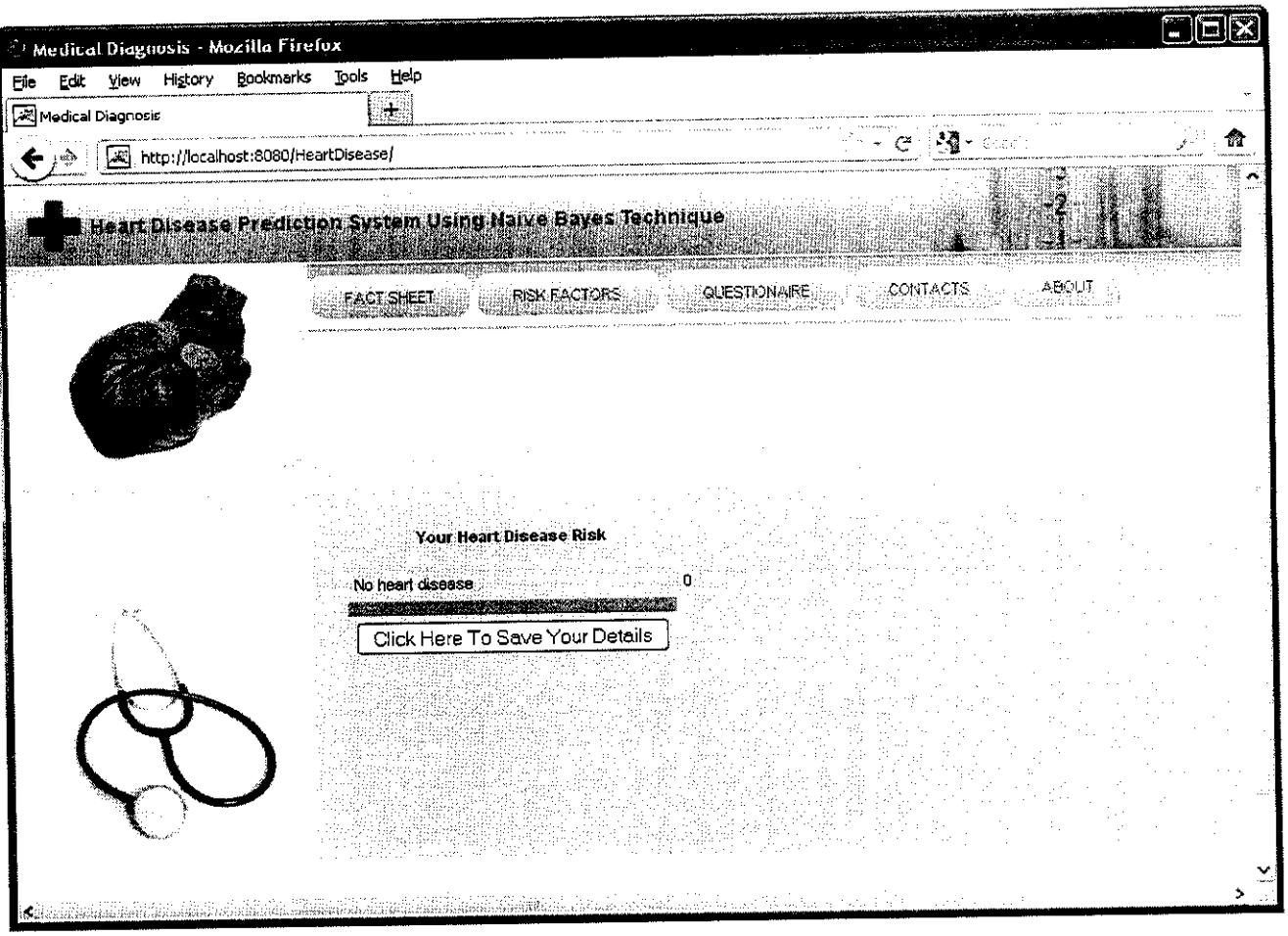


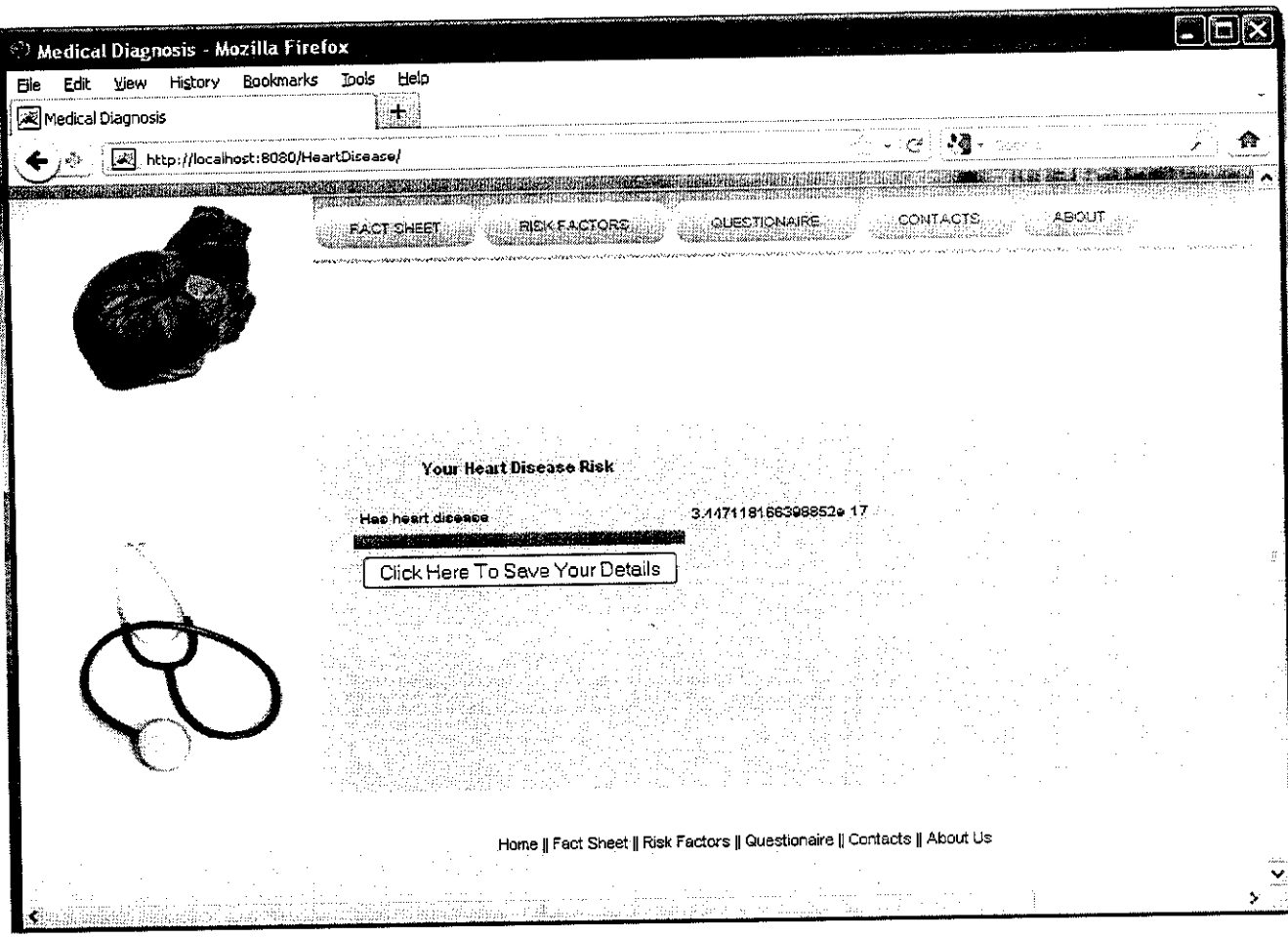
QUESTIONNAIRES OF THE IHDPS

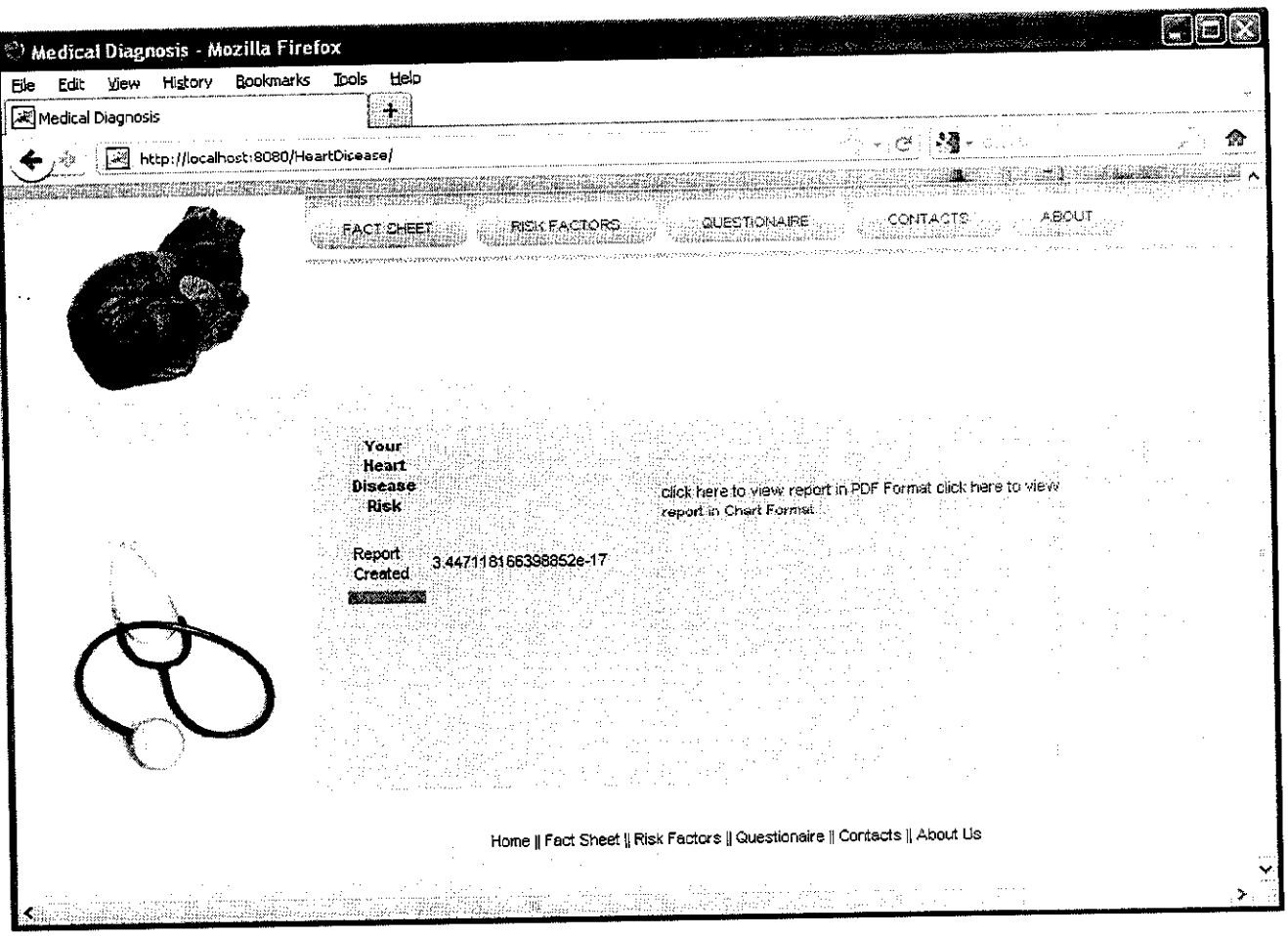



Medical Diagnosis - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Medical Diagnosis
http://localhost:8080/HeartDisease/
Heart Disease Prediction System Using Naive Bayes Technique
FACT SHEET
RISK FACTORS
QUESTIONAIRE
CONTACTS
ABOUT
Your Heart Disease Risk
What is your sex ?
Male
Female
Next
Men are at higher risk for heart disease at an earlier age than women. The risk of a heart attack begins to increase more rapidly when a man reaches age 45. A woman's risk of heart attack begins to increase when she reaches age 55. The hormone estrogen is protective to the heart, so when a woman reaches menopause and her estrogen production reduces, discuss HRT with your physician.
1.0 - male
0.0 - female
Home || Fact Sheet || Risk Factors || Questionaire || Contacts || About Us

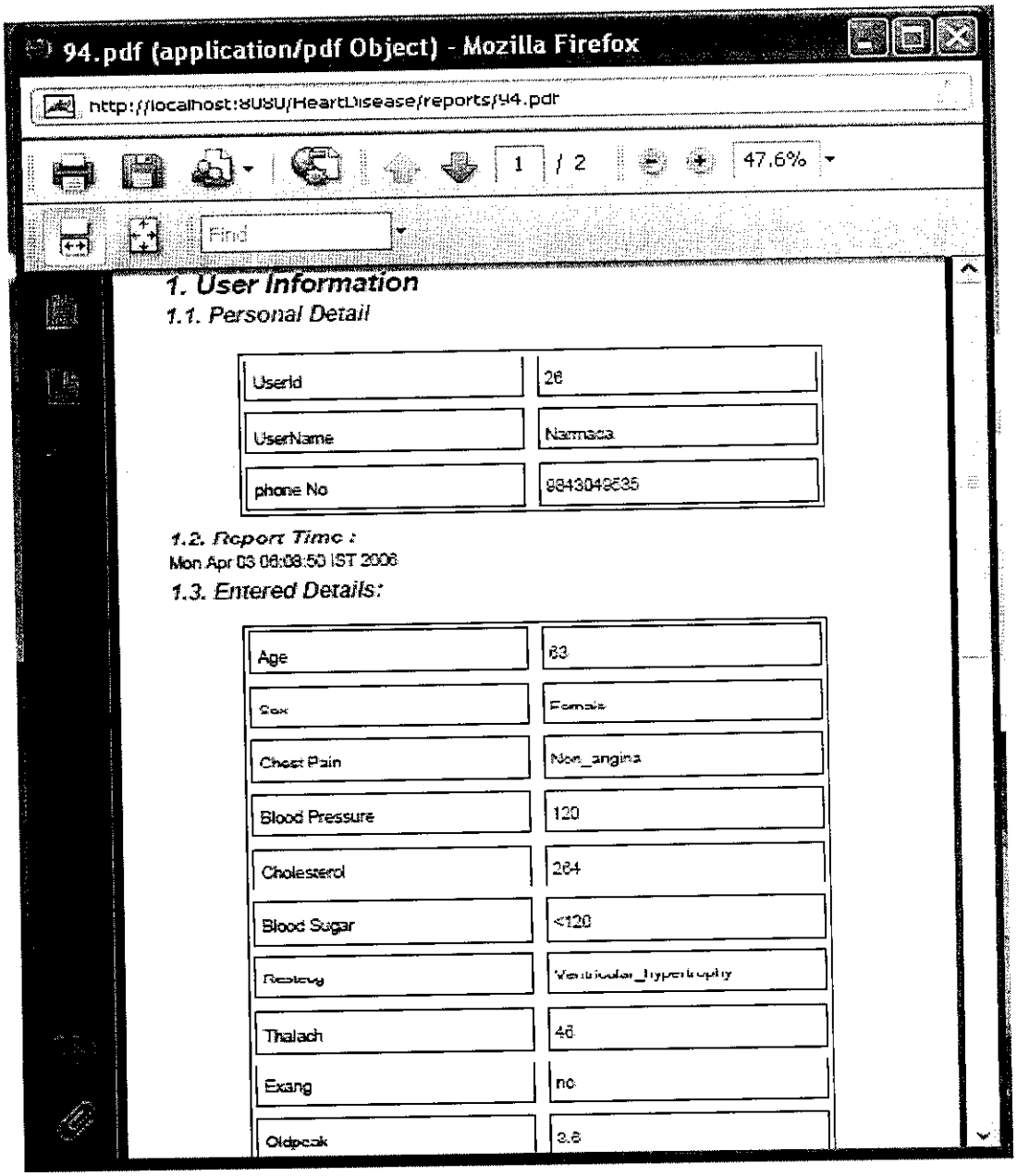
OUTPUT PAGE OF THE IHDPS



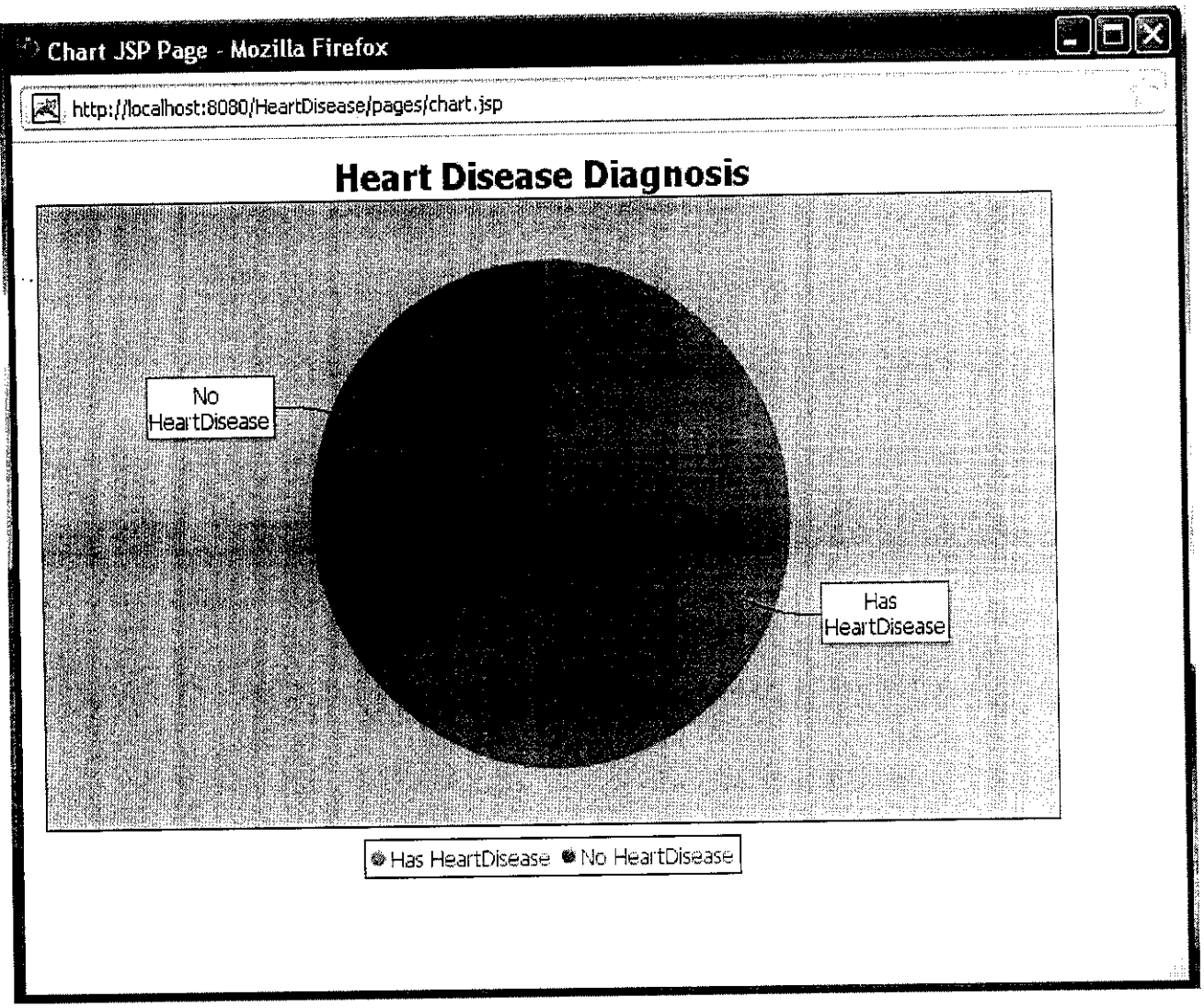

Medical Diagnosis - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Medical Diagnosis
http://localhost:8080/HeartDisease/
FACT SHEET
RISK FACTORS
QUESTIONAIRE
CONTACTS
ABOUT
Your Heart Disease Risk
Has heart disease
3.1471181663988528e 17
Click Here To Save Your Details
Home || Fact Sheet || Risk Factors || Questionaire || Contacts || About Us


Medical Diagnosis - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Medical Diagnosis
http://localhost:8080/HeartDisease/
FACT SHEET
RISK FACTORS
QUESTIONAIRE
CONTACTS
ABOUT
Your Heart Disease Risk
click here to view report in PDF Format click here to view report in Chart Format
Report Created
3.447118166398852e-17
Home || Fact Sheet || Risk Factors || Questionaire || Contacts || About Us

OUTPUT IN PDF FORMAT



OUTPUT IN CHART FORMAT



APPENDIX – II

SAMPLE CODING

Database.java

```
package com.heart.bean;

import java.io.*;
import java.util.*;
import java.sql.*;

public class Database implements Serializable
{
        private static String  jdbcDriver = "";
        private static String  dbURL = "";
        private static String  username = "";
        private static String  password = "";

        private Connection con;


        public Database() throws SQLException, ClassNotFoundException
        {
      ResourceBundle bundle = ResourceBundle.getBundle("/MessageResources");
                jdbcDriver = bundle.getString("jdbc.driver");
                dbURL =  bundle.getString("jdbc.url");
                username = bundle.getString("jdbc.user");
                password = bundle.getString("jdbc.password");
      Class.forName(jdbcDriver); //set Java database connectivity driver
      con = DriverManager.getConnection(dbURL, username, password);
        }

        public ResultSet executeQuery(String query)throws SQLException
        {
          PreparedStatement st  = con.prepareStatement(query);
     return st.executeQuery();
   }

        public int executeUpdate(String statement)throws SQLException
        {
                PreparedStatement st  = con.prepareStatement(statement);
     return st.executeUpdate();
        }

        public void close()
```

```
            try
        {
                    con.close();
            }
        catch (SQLException sqlException)
        {
                    sqlException.printStackTrace();
            con = null;
        }
    }
    protected void finalize()
    {
            close();
    }
}
```

DatabaseConnector.java

```
package com.heart.bean;

import java.sql.*;

public class DatabaseConnector {

        public static Connection getConnection() {
            Connection con = null;
            String driver = "com.mysql.jdbc.Driver";
            try {
               Class.forName(driver).newInstance();

            } catch (Exception e) {
               System.out.println("Failed to load mySQL driver.");
               return null;
            }
            try {
               con = DriverManager

 .getConnection("jdbc:mysql://localhost:3306/heartdisease","root","password");

            } catch (Exception e) {
               e.printStackTrace();
            }
            return con;
          }
 }
```

HeartBean.java

```
package com.heart.bean;

import java.util.*;

public class HeartBean {
   //private String name;
   private String age;
   private String sex;
   private String ChestPain;
   private String BloodPressure;
        private String Cholestrol;
   private String BloodSugar;
        private String Electrocardiographic;
   private String Thalach;
        private String Exang;
        private String OldPeak;
        private String Slope;
        private String CA;
        private String Thal;
    private String error;
    private Hashtable errors;

    public HeartBean() {
     //  name = "";
       age = "";
       sex = "";
       ChestPain = "";
       BloodPressure = "";
       Cholestrol = "";
       BloodSugar = "";
       Electrocardiographic = "";
       Thalach = "";
       Exang = "";
       OldPeak="";
       Slope = "";
       CA = "";
       Thal = "";
       error="";
       errors = new Hashtable();
     } /*public String getName() {
       return name;
           } public void setName(String name) {
       this.name = name;
           } **/public String getAge() {
       return age;
           } public void setAge(String age) {
```

```
    } public String getSex() {
  return sex;
    } public void setSex(String sex) {
  this.sex = sex;
    } public String getChestPain() {
  return ChestPain;
} public void setChestPain(String ChestPain) {
  this.ChestPain = ChestPain;
} public String getBloodPressure() {
  return BloodPressure;
} public void setBloodPressure(String BloodPressure) {
  this.BloodPressure = BloodPressure;
} public String getCholestrol() {
  return Cholestrol;
    } public void setCholestrol(String Cholestrol) {
  this.Cholestrol = Cholestrol;
    } public String getBloodSugar() {
  return BloodSugar;
    } public void setBloodSugar(String BloodSugar) {
  this.BloodSugar = BloodSugar;
    } public String getElectrocardiographic() {
  return Electrocardiographic;
    } public void setElectrocardiographic(String Electrocardiographic) {
  this.Electrocardiographic = Electrocardiographic;
    } public String getThalach() {
  return Thalach;
    } public void setThalach(String Thalach) {
  this.Thalach = Thalach;
    } public String getExang() {
  return Exang;
    } public void setExang(String Exang) {
  this.Exang = Exang;
    } public String getOldPeak() {
  return OldPeak;
    } public void setOldPeak(String OldPeak) {
  this.OldPeak = OldPeak;
    } public String getSlope() {
  return Slope;
    } public void setSlope(String Slope) {
  this.Slope = Slope;
    } public String getCA() {
  return CA;
    } public void setCA(String CA) {
  this.CA = CA;
    } public String getThal() {
  return Thal;
    } public void setThal(String Thal) {
  this.Thal = Thal;
    } public String getError() {
```

```
    } public String getErrors(String s) {
            String errorMsg =(String)errors.get(s.trim());
       return (errorMsg == null) ? "":errorMsg;
    } public void setErrors(String key, String msg) {

    } public void setError(String msg) {
       error=msg;
    } public boolean isValidate() {
  boolean allOk=true;
/*  if (name.equals("")) {
          errors.put("name","Name required");
          name="";
          allOk=false;
       }*/ if (age.equals("")) {
          errors.put("age","Age required");
          age="";
          allOk=false;
       } if (sex.equals("")) {
          errors.put("sex","Sex required");
          sex="";
          allOk=false;
       } if (ChestPain.equals("")) {
          errors.put("ChestPain","ChestPain required");
          ChestPain="";
          allOk=false;
       } if (BloodPressure.equals("")) {
          errors.put("BloodPressure","BloodPressure required");
          BloodPressure="";
          allOk=false;
       } if (Cholestrol.equals("")) {
          errors.put("Cholestrol","Cholestrol required");
          Cholestrol="";
          allOk=false;
       } if (BloodSugar.equals("")) {
          errors.put("BloodSugar","BloodSugar required");
          BloodSugar="";
          allOk=false;
       } if (Electrocardiographic.equals("")) {
          errors.put("Electrocardiographic","Electrocardiographic required");
          Electrocardiographic="";
          allOk=false;
       } if (Thalach.equals("")) {
          errors.put("Thalach","Thalach required");
          Thalach="";
          allOk=false;
       } if (Exang.equals("")) {
          errors.put("Exang","Exang required");
          Exang="";
          allOk=false;
```

```
            errors.put("OldPeak","OldPeak required");
            OldPeak="";
            allOk=false;
          } if (Slope.equals("")) {
            errors.put("Slope","Slope required");
            Slope="";
            allOk=false;
          } if (CA.equals("")) {
            errors.put("CA","CA required");
            CA="";
            allOk=false;
          } if (Thal.equals("")) {
            errors.put("Thal","Thal required");
            Thal="";
            allOk=false;
          }
      return allOk;
    }
}
```

Naïve Bayes:

Calculation.java

```
package com.heart.naivebayes;

import com.heart.naivebayes.Detection;
import com.heart.naivebayes.Prediction;

import java.io.*;
import java.util.*;

public class Calculation
{
  double pdfMale0,pdfMale1,pdfFemale0,pdfFemale1;
  double
pdfCPType10,pdfCPType11,pdfCPType0,pdfCPType1,pdfCPNa0,pdfCPNa1,pdfCPAy0,pdf
CPAy1;
  double pdfBSGt0,pdfBSGt1,pdfBSLt0,pdfBSLt1;
  double pdfResNL0,pdfResNL1,pdfResABN0,pdfResABN1,pdfResHy0,pdfResHy1;
  double pdfExY0,pdfExY1,pdfExN0,pdfExN1;
  double pdfSlopeUs0,pdfSlopeUs1,pdfSlopeFt0,pdfSlopeFt1,pdfSlopeDs0,pdfSlopeDs1;
  double
pdfCA0Bv0,pdfCA0Bv1,pdfCA1Bv0,pdfCA1Bvl,pdfCA2Bv0,pdfCA2Bvl,pdfCA3Bv0,pdfC
A3Bvl;
  double pdfTalNL0,pdfTalNL1,pdfTalFD0,pdfTalFD1,pdfTalRD0,pdfTalRD1;
```

```
double meanAgeD0,meanAgeD1;
double meanBpD0,meanBpD1;
double meanCholesterolD0,meanCholesterolD1;
double meanThalachD0,meanThalachD1;
double meanOldpeakD0,meanOldpeakD1;

double stdAgeD0,stdAgeD1;
double stdBloodPressureD0,stdBloodPressureD1;
double stdCholesterolD0,stdCholesterolD1;
double stdThalachD0,stdThalachD1;
double stdOldpeakD0,stdOldpeakD1;

Detection d2 = new Detection();
Prediction p2 = new Prediction();

        /**Nominal attributes Calculation**/
public double setPdfMale0()                          /**Sex PDF**/
{
 pdfMale0 = p2.findNominalPDF(d2.setMale0(),d2.setDiagnosis0());
 return pdfMale0;
}
public double setPdfMale1()
{
 pdfMale1 = p2.findNominalPDF(d2.setMale0(),d2.setDiagnosis1());
 return pdfMale1;
}
public double setPdfFemale0()
{
 pdfFemale0 = p2.findNominalPDF(d2.setFemale0(),d2.setDiagnosis0());
 return pdfFemale0;
}
public double setPdfFemale1()
{
 pdfFemale1 = p2.findNominalPDF(d2.setFemale1(),d2.setDiagnosis1());
 return pdfFemale1;
}

public double setPdfCPType10()                       /**ChestPain PDF**/
{
 pdfCPType10 = p2.findNominalPDF(d2.setCP10(),d2.setDiagnosis0());
 return pdfCPType10;
}
public double setPdfCPType11()
{
 pdfCPType11 = p2.findNominalPDF(d2.setCP11(),d2.setDiagnosis1());
 return pdfCPType11;
}
public double setPdfCPType0()
```

```
 pdfCPType0 = p2.findNominalPDF(d2.setCP20(),d2.setDiagnosis0());
 return pdfCPType0;
}
public double setPdfCPType1()
{
 pdfCPType1 = p2.findNominalPDF(d2.setCP21(),d2.setDiagnosis1());
 return pdfCPType1;
}
public double setPdfCPNa0()
{
 pdfCPNa0 = p2.findNominalPDF(d2.setCP30(),d2.setDiagnosis0());
 return pdfCPNa0;
}
public double setPdfCPNa1()
{
 pdfCPNa1 = p2.findNominalPDF(d2.setCP31(),d2.setDiagnosis1());
 return pdfCPNa1;
}
public double setPdfCPAy0()
{
 pdfCPAy0 = p2.findNominalPDF(d2.setCP40(),d2.setDiagnosis0());
 return pdfCPAy0;
}
public double setPdfCPAy1()
{
 pdfCPAy1 = p2.findNominalPDF(d2.setCP41(),d2.setDiagnosis1());
 return pdfCPAy1;
}

public double setPdfBSGt0()                    /**BloodSugar PDF**/
{
 pdfBSGt0 = p2.findNominalPDF(d2.setBSG0(),d2.setDiagnosis0());
 return pdfBSGt0;
}
public double setPdfBSGt1()
{
 pdfBSGt1 = p2.findNominalPDF(d2.setBSG1(),d2.setDiagnosis1());
 return pdfBSGt1;
}
public double setPdfBSLt0()
{
 pdfBSLt0 = p2.findNominalPDF(d2.setBSL0(),d2.setDiagnosis0());
 return pdfBSLt0;
}
public double setPdfBSLt1()
{
 pdfBSLt1 = p2.findNominalPDF(d2.setBSL1(),d2.setDiagnosis1());
 return pdfBSLt1;
}
```

```
public double setPdfResNL0()                    /**Restang PDF**/
{
 pdfResNL0 = p2.findNominalPDF(d2.setRestecgNL0(),d2.setDiagnosis0());
 return pdfResNL0;
}
public double setPdfResNL1()
{
 pdfResNL1 = p2.findNominalPDF(d2.setRestecgNL1(),d2.setDiagnosis1());
 return pdfResNL1;
}
public double setPdfResABN0()
{
 pdfResABN0 = p2.findNominalPDF(d2.setRestecgABN0(),d2.setDiagnosis0());
 return pdfResABN0;
}
public double setPdfResABN1()
{
 pdfResABN1 = p2.findNominalPDF(d2.setRestecgABN1(),d2.setDiagnosis1());
 return pdfResABN1;
}
public double setPdfResHy0()
{
 pdfResHy0 = p2.findNominalPDF(d2.setRestecgHY0(),d2.setDiagnosis0());
 return pdfResHy0;
}
public double setPdfResHy1()
{
 pdfResHy1 = p2.findNominalPDF(d2.setRestecgHY1(),d2.setDiagnosis1());
 return pdfResHy1;
}

public double setPdfExY0()                       /**Exang PDF**/
{
 pdfExY0 = p2.findNominalPDF(d2.setExangY0(),d2.setDiagnosis0());
 return pdfExY0;
}
public double setPdfExY1()
{
 pdfExY1 = p2.findNominalPDF(d2.setExangY1(),d2.setDiagnosis1());
 return pdfExY1;
}
public double setPdfExN0()
{
 pdfExN0 = p2.findNominalPDF(d2.setExangN0(),d2.setDiagnosis0());
 return pdfExN0;
}
public double setPdfExN1()
{
 pdfExN1 = p2.findNominalPDF(d2.setExangN1(),d2.setDiagnosis1());
```

```
}

public double setPdfSlopeUs0()                    /**Slope PDF**/
{
 pdfSlopeUs0 = p2.findNominalPDF(d2.setSlopeUS0(),d2.setDiagnosis0());
 return pdfSlopeUs0;
}
public double setPdfSlopeUs1()
{
 pdfSlopeUs1 = p2.findNominalPDF(d2.setSlopeUS1(),d2.setDiagnosis1());
 return pdfSlopeUs1;
}
public double setPdfSlopeFt0()
{
 pdfSlopeFt0 = p2.findNominalPDF(d2.setSlopeF0(),d2.setDiagnosis0());
 return pdfSlopeFt0;
}
public double setPdfSlopeFt1()
{
 pdfSlopeFt1 = p2.findNominalPDF(d2.setSlopeF0(),d2.setDiagnosis1());
 return pdfSlopeFt1;
}
public double setPdfSlopeDs0()
{
 pdfSlopeDs0 = p2.findNominalPDF(d2.setSlopeDS0(),d2.setDiagnosis0());
 return pdfSlopeDs0;
}
public double setPdfSlopeDs1()
{
 pdfSlopeDs1 = p2.findNominalPDF(d2.setSlopeDS1(),d2.setDiagnosis1());
 return pdfSlopeDs1;
}

public double setPdfCA0Bv0()                       /**CA PDF**/
{
 pdfCA0Bv0 = p2.findNominalPDF(d2.setCA0BV0(),d2.setDiagnosis0());
 return pdfCA0Bv0;
}
public double setPdfCA0Bv1()
{
 pdfCA0Bv1 = p2.findNominalPDF(d2.setCA0BV1(),d2.setDiagnosis1());
 return pdfCA0Bv1;
}
public double setPdfCA1Bv0()
{
 pdfCA1Bv0 = p2.findNominalPDF(d2.setCA1BV0(),d2.setDiagnosis0());
 return pdfCA1Bv0;
}
public double setPdfCA1Bvl()
```

```
 pdfCA1Bvl = p2.findNominalPDF(d2.setCA1BV1(),d2.setDiagnosis1());
 return pdfCA1Bvl;
}
public double setPdfCA2Bv0()
{
 pdfCA2Bv0 = p2.findNominalPDF(d2.setCA2BV0(),d2.setDiagnosis0());
 return pdfCA2Bv0;
}
public double setPdfCA2Bvl()
{
 pdfCA2Bvl = p2.findNominalPDF(d2.setCA2BV1(),d2.setDiagnosis1());
 return pdfCA2Bvl;
}
public double setPdfCA3Bv0()
{
 pdfCA3Bv0 = p2.findNominalPDF(d2.setCA3BV0(),d2.setDiagnosis0());
 return pdfCA3Bv0;
}
public double setPdfCA3Bvl()
{
 pdfCA3Bvl = p2.findNominalPDF(d2.setCA3BV1(),d2.setDiagnosis1());
 return pdfCA3Bvl;
}

public double setPdfTalNL0()                    /**Thal PDF**/
{
 pdfTalNL0 = p2.findNominalPDF(d2.setThalNL0(),d2.setDiagnosis0());
 return pdfTalNL0;
}
public double setPdfTalNL1()
{
 pdfTalNL1 = p2.findNominalPDF(d2.setThalNL1(),d2.setDiagnosis1());
 return pdfTalNL1;
}
public double setPdfTalFD0()
{
 pdfTalFD0 = p2.findNominalPDF(d2.setThalFD0(),d2.setDiagnosis0());
 return pdfTalFD0;
}
public double setPdfTalFD1()
{
 pdfTalFD1 = p2.findNominalPDF(d2.setThalFD1(),d2.setDiagnosis1());
 return pdfTalFD1;
}
public double setPdfTalRD0()
{
 pdfTalRD0 = p2.findNominalPDF(d2.setThalRD0(),d2.setDiagnosis0());
 return pdfTalRD0;
}
```

```
{
 pdfTalRD1 = p2.findNominalPDF(d2.setThalRD1(),d2.setDiagnosis1());
 return pdfTalRD1;
}

public double setPdfDiagnosis0()                    /**Diagnosis PDF**/
{
 pdfDiagnosis0 = p2.findNominalPDF(d2.setDiagnosis0(),297);
 return pdfDiagnosis0;
}
public double setPdfDiagnosis1()
{
 pdfDiagnosis1 = p2.findNominalPDF(d2.setDiagnosis1(),297);
 return pdfDiagnosis1;
}
          /**Numerical attributes Calculation**/

/** ---  To Find Numerical Mean Values  --- **/

 public double setMeanAgeD0(){                          /**Age Mean**/
  meanAgeD0= p2.findMean(d2.setAgeD0());
System.out.println("Mean[Age(0)] : "+meanAgeD0);
  return meanAgeD0;
  }
  public double setMeanAgeD1(){
   meanAgeD1= p2.findMean(d2.setAgeD1());
System.out.println("Mean[Age(1)] : "+meanAgeD1);
   return meanAgeD1;
   }

  public double setMeanBloodPressureD0(){          /**BloodPressure Mean**/
   meanBpD0= p2.findMean(d2.setBloodPressureD0());
//System.out.println("Mean[BloodPressure(0)] : "+meanBpD0);
   return meanBpD0;
   }
   public double setMeanBloodPressureD1(){
    meanBpD1= p2.findMean(d2.setBloodPressureD1());
//System.out.println("Mean[BloodPressure(1)] : "+meanBpD1);
    return meanBpD1;
    }

   public double setMeanCholesterolD0(){           /**Cholesterol Mean**/
    meanCholesterolD0= p2.findMean(d2.setCholesterolD0());
//System.out.println("Mean[Cholesterol(0)] : "+meanCholesterolD0);
    return meanCholesterolD0;
    }
    public double setMeanCholesterolD1(){
     meanCholesterolD1= p2.findMean(d2.setCholesterolD1());
//System.out.println("Mean[Cholesterol(1)] : "+meanCholesterolD1);
```

```
}

public double setMeanThalachD0(){                    /**Thalach Mean**/
 meanThalachD0= p2.findMean(d2.setThalachD0());
//System.out.println("Mean[Thalach(0)] : "+meanThalachD0);
 return meanThalachD0;
}
public double setMeanThalachD1(){
 meanThalachD1= p2.findMean(d2.setThalachD1());
//System.out.println("Mean[Thalach(1)] : "+meanThalachD1);
 return meanThalachD1;
}

public double setMeanOldpeakD0(){                    /**Oldpeak Mean**/
 meanOldpeakD0= p2.findMean(d2.setOldpeakD0());
//System.out.println("Mean[Oldpeak(0)] : "+meanOldpeakD0);
 return meanOldpeakD0;
}
public double setMeanOldpeakD1(){
 meanOldpeakD1= p2.findMean(d2.setOldpeakD1());
//System.out.println("Mean[Oldpeak(1)] : "+meanOldpeakD1);
 return meanOldpeakD1;
}

/** ----  To Find Numerical Standard Deviation Values  --- **/

public double setStdAgeD0(){                         /**Age STD**/
   stdAgeD0 = p2.findStandardDeviation(d2.setAgeD0(),setMeanAgeD0());
//System.out.println("stdAgeD0 : "+stdAgeD0);
   return stdAgeD0;
}
public double setStdAgeD1(){
   stdAgeD1 = p2.findStandardDeviation(d2.setAgeD1(),setMeanAgeD1());
//System.out.println("stdAgeD1 : "+stdAgeD1);
   return stdAgeD1;
}

public double setStdBloodPressureD0() {              /**BloodPressure STD**/
   stdBloodPressureD0 =
p2.findStandardDeviation(d2.setBloodPressureD0(),setMeanBloodPressureD0());
//System.out.println("stdBloodPressureD0 : "+stdBloodPressureD0);
   return stdBloodPressureD0;
}
public double setStdBloodPressureD1() {
   stdBloodPressureD1 =
p2.findStandardDeviation(d2.setBloodPressureD1(),setMeanBloodPressureD1());
//System.out.println("stdBloodPressureD1 : "+stdBloodPressureD1);
   return stdBloodPressureD1;
}
```

```
 public double setStdCholesterolD0(){                    /**Cholesterol STD**/
    stdCholesterolD0 =
p2.findStandardDeviation(d2.setCholesterolD0(),setMeanCholesterolD0());
//System.out.println("stdCholesterolD0 : "+stdCholesterolD0);
    return stdCholesterolD0;
  }
  public double setStdCholesterolD1(){
    stdCholesterolD1 =
p2.findStandardDeviation(d2.setCholesterolD1(),setMeanCholesterolD1());
//System.out.println("stdCholesterolD1 : "+stdCholesterolD1);
    return stdCholesterolD1;
  }

  public double setStdThalachD0(){                    /**Thalach STD**/
    stdThalachD0 = p2.findStandardDeviation(d2.setThalachD0(),setMeanThalachD0());
//System.out.println("stdThalachD0 : "+stdThalachD0);
    return stdThalachD0;
  }
  public double setStdThalachD1(){
    stdThalachD1 = p2.findStandardDeviation(d2.setThalachD1(),setMeanThalachD1());
//System.out.println("stdThalachD1 : "+stdThalachD1);
    return stdThalachD1;
  }

  public double setStdOldpeakD0(){                    /**Oldpeak STD**/
    stdOldpeakD0 = p2.findStandardDeviation(d2.setOldpeakD0(),setMeanOldpeakD0());
//System.out.println("stdOldpeakD0 : "+stdOldpeakD0);
    return stdAgeD0;
  }
  public double setStdOldpeakD1(){
    stdOldpeakD1 = p2.findStandardDeviation(d2.setOldpeakD1(),setMeanOldpeakD1());
//System.out.println("stdOldpeakD1 : "+stdOldpeakD1);
    return stdOldpeakD1;
  }
}
```

Detection.java

```
package com.heart.naivebayes;

import java.io.*;
import java.util.*;
import java.lang.Math;

public class Detection
{
 static Hashtable map = new Hashtable();
```

```
static final Integer ONE = new Integer(1);
static ArrayList Age = new ArrayList();
  static ArrayList AgeD0 = new ArrayList();
  static ArrayList AgeD1 = new ArrayList();
static ArrayList Sex = new ArrayList();
static ArrayList ChestPain = new ArrayList();
static ArrayList BloodPressure = new ArrayList();
  static ArrayList BloodPressureD0 = new ArrayList();
  static ArrayList BloodPressureD1 = new ArrayList();
static ArrayList Cholesterol = new ArrayList();
  static ArrayList CholesterolD0 = new ArrayList();
  static ArrayList CholesterolD1 = new ArrayList();
static ArrayList BloodSugar = new ArrayList();
static ArrayList Restecg = new ArrayList();
static ArrayList Thalach = new ArrayList();
  static ArrayList ThalachD0 = new ArrayList();
  static ArrayList ThalachD1 = new ArrayList();
static ArrayList Exang = new ArrayList();
static ArrayList Oldpeak = new ArrayList();
  static ArrayList OldpeakD0 = new ArrayList();
  static ArrayList OldpeakD1 = new ArrayList();
static ArrayList Slope = new ArrayList();
static ArrayList CA = new ArrayList();
static ArrayList Thal = new ArrayList();
static ArrayList Height = new ArrayList();
static ArrayList Weight = new ArrayList();
static ArrayList Diagnosis = new ArrayList();

 static int k,l,m,n;
 static int a=0,b=0,c=0,d=0;
 private int maleD0,femaleD0 = 0,maleD1 = 0,femaleD1 = 0;
 private int type1_anginaD0 = 0,type1_anginaD1 = 0,type_anginaD0 = 0,type_anginaD1 =
0,non_anginaD0 =0,non_anginaD1 =0,asymptomaticD0 =0,asymptomaticD1 =0;
 private int gt_120D0 =0,gt_120D1 =0,lt_120D0 =0,lt_120D1 =0;
 private int normalD0 =0,normalD1 =0,abnormalityD0 =0,abnormalityD1 =0,hypertrophyD0
=0,hypertrophyD1 =0;
 private int yesD0 =0,yesD1 =0,noD0 =0,noD1 =0;
 private int unslopingD0 =0,unslopingD1 =0,flatD0 =0,flatD1 =0,downslopingD0
=0,downslopingD1 =0;
 private int blood_vessel0D0 = 0,blood_vessel1D0 = 0,blood_vessel2D0 =
0,blood_vessel3D0 = 0, blood_vessel0D1 = 0,blood_vessel1D1 = 0,blood_vessel2D1 =
0,blood_vessel3D1 =0;
 private int normal2D0 = 0,fixed_defectD0 = 0,reversible_defectD0 = 0,normal2D1 =
0,fixed_defectD1 = 0,reversible_defectD1 = 0;
 private int result0=0,result1 = 0;

 public static void read()                    /**Read Dataset Values**/
 {
   try
```

```
    FileReader fr = new FileReader("C:/Program Files/Apache Software Foundation/Tomcat
5.5/webapps/HeartDisease/WEB-INF/dataset/HeartDataset.dat");
    BufferedReader br = new BufferedReader(fr);
    String line="";
    clearArrayList();
    while (line != null)
    {
      processLine(line, map);
      line = br.readLine();
      k=0;
      l=0;
      m=0;
      n=0;
    }
  }
  catch(Exception e2)
  {
    System.out.println(e2.toString());
  }
}

public static void processLine(String line, Map map)     /**Add datas in ArrayList**/
{
 StringTokenizer st = new StringTokenizer(line);

 while (st.hasMoreTokens())
 {
  String one = st.nextToken();
  if(k==0){
   Age.add(one);
   k=1;
  } else if(k==1) {
   Cholesterol.add(one);
   k=2;
  } else if(k==2) {
   Exang.add(one);
   k=3;
  } else {
   Thal.add(one);
  }
  String two = st.nextToken();
  if(l==0) {
   Sex.add(two);
   l=1;
  } else if(l==1) {
   BloodSugar.add(two);
   l=2;
  } else if(l==2) {
   Oldpeak.add(two);
```

```
  } else {
   Height.add(two);
  }
  String three = st.nextToken();
  if(m==0) {
   ChestPain.add(three);
   m=1;
  } else if(m==1) {
   Restecg.add(three);
   m=2;
  } else if(m==2) {
   Slope.add(three);
   m=3;
  } else {
   Weight.add(three);
  }
  String four = st.nextToken();
  if(n==0) {
   BloodPressure.add(four);
   n=1;
  } else if(n==1) {
   Thalach.add(four);
   n=2;
  } else if(n==2) {
   CA.add(four);
   n=3;
  } else {
   Diagnosis.add(four);
  }
 }
}

public void findAge() {                    /**Add datas into Age ArrayList**/
 for(int i=0;i<Diagnosis.size();i++)
 {
    String aa = (String)Diagnosis.get(i);
    String str = (String)Age.get(i);
    if(aa.equals("0"))
    {
     AgeD0.add(str);
    }
    else
    {
     AgeD1.add(str);
    }
 }
}
public ArrayList setAgeD0()
{
```

```
  return AgeD0;
}
public ArrayList setAgeD1()
{
  findAge();
  return AgeD1;
}

public void findSex() {                    /**Add datas into Sex ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
     String aa = (String)Diagnosis.get(i);
     if(aa.equals("0"))
     {
        String str = (String)Sex.get(i);
        if(str.equals("1.0"))
        {
          maleD0=maleD0+1;
        }
        else
        {
          femaleD0++;
        }
     }
     else if(aa.equals("1"))
     {
        String str = (String)Sex.get(i);
        if(str.equals("1.0"))
        {
          maleD1++;
        }
        else
        {
          femaleD1++;
        }
     }
   }
 }
 public int setMale0()
 {
  findSex();
  return maleD0;
 }
 public int setMale1()
 {
  findSex();
  return maleD1;
 }
 public int setFemale0()
```

```
 findSex();
 return femaleD0;
}
public int setFemale1()
{
 findSex();
 return femaleD1;
}

public void findChestPain() {        /**Add datas into ChestPain ArrayList**/
 for(int i=0;i<Diagnosis.size();i++)
 {
    String aa = (String)Diagnosis.get(i);
    if(aa.equals("0"))
    {
       String str = (String)ChestPain.get(i);
       if(str.equals("1.0"))
       {
         type1_anginaD0++;
       }
       else if(str.equals("2.0"))
       {
         type_anginaD0++;
       }
       else if(str.equals("3.0"))
       {
         non_anginaD0++;
       }
       else {
        asymptomaticD0++;
       }
    }
    else
    {
       String str = (String)ChestPain.get(i);
       if(str.equals("1.0"))
       {
         type1_anginaD1++;
       }
       else if(str.equals("2.0"))
       {
         type_anginaD1++;
       }
       else if(str.equals("3.0"))
       {
         non_anginaD1++;
       }
       else {
        asymptomaticD1++;
```

```
    }
  }
}
public int setCP10()
{
 findChestPain();
 return type1_anginaD0;
}
public int setCP11()
{
 findChestPain();
 return type1_anginaD1;
}
public int setCP20()
{
 findChestPain();
 return type_anginaD0;
}
public int setCP21()
{
 findChestPain();
 return type_anginaD1;
}
public int setCP30()
{
 findChestPain();
 return non_anginaD0;
}
public int setCP31()
{
 findChestPain();
 return non_anginaD1;
}
public int setCP40()
{
 findChestPain();
 return asymptomaticD0;
}
public int setCP41()
{
 findChestPain();
 return asymptomaticD1;
}

public void findBloodPressure() { /**Add datas into BloodPressure ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
     String aa = (String)Diagnosis.get(i);
     String str = (String)BloodPressure.get(i);
```

```
    {
     BloodPressureD0.add(str);
    }
    else
    {
     BloodPressureD1.add(str);
    }
  }
}
public ArrayList setBloodPressureD0()
{
  findBloodPressure();
  return BloodPressureD0;
}
public ArrayList setBloodPressureD1()
{
  findBloodPressure();
  return BloodPressureD1;
}

public void findCholesterol() {    /**Add datas into Cholesterol ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
     String aa = (String)Diagnosis.get(i);
     String str = (String)Cholesterol.get(i);
     if(aa.equals("0"))
     {
      CholesterolD0.add(str);
     }
     else
     {
      CholesterolD1.add(str);
     }
  }
}
public ArrayList setCholesterolD0()
{
  findCholesterol();
  return CholesterolD0;
}
public ArrayList setCholesterolD1()
{
  findCholesterol();
  return CholesterolD1;
}

public void findBloodSugar() {     /**Add datas into BloodSugar ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
```

```
    if(aa.equals("0"))
    {
       String str = (String)BloodSugar.get(i);
       if(str.equals("1.0"))
       {
         gt_120D0++;
       }
       else
       {
         lt_120D0++;
       }
     }
     else if(aa.equals("1"))
     {
       String str = (String)BloodSugar.get(i);
       if(str.equals("1.0"))
       {
         gt_120D1++;
       }
       else
       {
         lt_120D1++;
       }
     }
   }
 }
 public int setBSG0()
 {
  findBloodSugar();
  return gt_120D0;
 }
 public int setBSG1()
 {
  findBloodSugar();
  return gt_120D1;
 }
 public int setBSL0()
 {
  findBloodSugar();
  return lt_120D0;
 }
 public int setBSL1()
 {
  findBloodSugar();
  return lt_120D1;
 }

 public void findRestecg() {          /**Add datas into Restecg ArrayList**/
   for(int i=0;i<Diagnosis.size();i++)
```

```
  String aa = (String)Diagnosis.get(i);
  if(aa.equals("0"))
  {
     String str = (String)Restecg.get(i);
     if(str.equals("0.0"))
     {
       normalD0++;
     }
     else if(str.equals("1.0"))
     {
       abnormalityD0++;
     }
     else if(str.equals("2.0"))
     {
       hypertrophyD0++;
     }
   }
   else
   {
     String str = (String)Restecg.get(i);
     if(str.equals("0.0"))
     {
       normalD1++;
     }
     else if(str.equals("1.0"))
     {
       abnormalityD1++;
     }
     else if(str.equals("2.0"))
     {
       hypertrophyD1++;
     }
   }
  }
}
public int setRestecgNL0()
{
 findRestecg();
 return normalD0;
}
public int setRestecgNL1()
{
 findRestecg();
 return normalD1;
}
public int setRestecgABN0()
{
 findRestecg();
 return abnormalityD0;
```

```
public int setRestecgABN1()
{
 findRestecg();
 return abnormalityD1;
}
public int setRestecgHY0()
{
 findRestecg();
 return hypertrophyD0;
}
public int setRestecgHY1()
{
 findRestecg();
 return hypertrophyD1;
}

 public void findThalach() {          /**Add datas into Thalach ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
     String aa = (String)Diagnosis.get(i);
     String str = (String)Thalach.get(i);
     if(aa.equals("0"))
     {
      ThalachD0.add(str);
     }
     else
     {
      ThalachD1.add(str);
     }
   }
 }
 public ArrayList setThalachD0()
 {
   findThalach();
   return ThalachD0;
 }
 public ArrayList setThalachD1()
 {
   findThalach();
   return ThalachD1;
 }


 public void findExang() {            /**Add datas into Exang ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
     String aa = (String)Diagnosis.get(i);
     if(aa.equals("0"))
     {
```

```
      if(str.equals("1.0"))
      {
        yesD0++;
      }
      else
      {
        noD0++;
      }
    }
    else
    {
      String str = (String)Exang.get(i);
      if(str.equals("1.0"))
      {
        yesD1++;
      }
      else
      {
        noD1++;
      }
    }
  }
}
public int setExangY0()
{
 findExang();
 return yesD0;
}
public int setExangY1()
{
 findExang();
 return yesD1;
}
public int setExangN0()
{
 findExang();
 return noD0;
}
public int setExangN1()
{
 findExang();
 return noD1;
}

public void findOldpeak() {          /**Add datas into Oldpeak ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
    String aa = (String)Diagnosis.get(i);
    String str = (String)Oldpeak.get(i);
```

```
    {
     OldpeakD0.add(str);
    }
    else
    {
     OldpeakD1.add(str);
    }
  }
}
public ArrayList setOldpeakD0()
{
  findOldpeak();
  return OldpeakD0;
}
public ArrayList setOldpeakD1()
{
  findOldpeak();
  return OldpeakD1;
}


public void findSlope() {              /**Add datas into Slope ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
    String aa = (String)Diagnosis.get(i);
    if(aa.equals("0"))
    {
      String str = (String)Slope.get(i);
      if(str.equals("1.0"))
      {
        unslopingD0++;
      }
      else if(str.equals("2.0"))
      {
        flatD0++;
      }
      else if(str.equals("3.0"))
      {
        downslopingD0++;
      }
    }
    else
    {
      String str = (String)Slope.get(i);
      if(str.equals("1.0"))
      {
        unslopingD1++;
      }
      else if(str.equals("2.0"))
```

```
            flatD1++;
          }
          else if(str.equals("3.0"))
          {
            downslopingD1++;
          }
        }
      }
    }

public int setSlopeUS0()
{
 findSlope();
 return unslopingD0;
}
public int setSlopeUS1()
{
 findSlope();
 return unslopingD1;
}
public int setSlopeF0()
{
 findSlope();
 return flatD0;
}
public int setSlopeF1()
{
  findSlope();
 return flatD1;
}
public int setSlopeDS0()
{
 findSlope();
 return downslopingD0;
}
public int setSlopeDS1()
{
 findSlope();
 return downslopingD1;
}

public void findCA() {                    /**Add datas into CA ArrayList**/
  for(int i=0;i<Diagnosis.size();i++)
  {
  String aa = (String)Diagnosis.get(i);
  if(aa.equals("0"))
      {

    String str = (String)CA.get(i);
```

```
      blood_vessel0D0++;
    } else if(str.equals("1.0")) {
      blood_vessel1D0++;
    } else if(str.equals("2.0")) {
      blood_vessel2D0++;
    } else if(str.equals("3.0")){
      blood_vessel3D0++;
    }
  } else
  {

    String str = (String)CA.get(i);
    if(str.equals("0.0")) {
      blood_vessel0D1++;
    } else if(str.equals("1.0")) {
      blood_vessel1D1++;
    } else if(str.equals("2.0")) {
      blood_vessel2D1++;
    } else if(str.equals("3.0")){
      blood_vessel3D1++;
    }

    }
   }
}
public int setCA0BV0()
{
 findCA();
 return blood_vessel0D0;
}
public int setCA0BV1()
{
 findCA();
 return blood_vessel0D1;
}
public int setCA1BV0()
{
 findCA();
 return blood_vessel1D0;
}
public int setCA1BV1()
{
 findCA();
 return blood_vessel1D1;
}
public int setCA2BV0()
{
 findCA();
 return blood_vessel2D0;
```

```
public int setCA2BV1()
{
 findCA();
 return blood_vessel2D1;
}
public int setCA3BV0()
{
 findCA();
 return blood_vessel3D0;
}
public int setCA3BV1()
{
 findCA();
 return blood_vessel3D1;
}

public void findThal() {                /**Add datas into Thal ArrayList**/
for(int i=0;i<Diagnosis.size();i++)
{
  String aa = (String)Diagnosis.get(i);
  if(aa.equals("0"))
  {
    String str = (String)Thal.get(i);
    if(str.equals("3.0")) {
       normal2D0++;
    } else if(str.equals("6.0")) {
       fixed_defectD0++;
    } else if(str.equals("7.0")) {
       reversible_defectD0++;
    }
  }else
  {
    String str = (String)Thal.get(i);
    if(str.equals("3.0")) {
       normal2D1++;
    } else if(str.equals("6.0")) {
       fixed_defectD1++;
    } else if(str.equals("7.0")) {
       reversible_defectD1++;
    }
  }
}
}
public int setThalNL0()
{
 findThal();
 return normal2D0;
}
public int setThalNL1()
```

```
findThal();
return normal2D1;
}
public int setThalFD0()
{
findThal();
return fixed_defectD0;
}
public int setThalFD1()
{
findThal();
return fixed_defectD1;
}
public int setThalRD0()
{
findThal();
return reversible_defectD0;
}
public int setThalRD1()
{
findThal();
return reversible_defectD1;
}


public ArrayList findHeight() {        /**Add datas into Height ArrayList**/
   return Height;
}

public ArrayList findWeight() {        /**Add datas into Weight ArrayList**/
   return Weight;
}

public void findDiagnosis() {        /**Add datas into Diagnosis ArrayList**/
for(int i=0;i<Diagnosis.size();i++)
  {
    String str = (String)Diagnosis.get(i);
    if(str.equals("0")) {
       result0++;
    } else {
       result1++;
    }
  }
}

public int setDiagnosis0()
{
  findDiagnosis();
  return result0;
```

```
public int setDiagnosis1()
{
 findDiagnosis();
 return result1;
}

public static void clearArrayList(){        /**Clear all ArrayList values**/
  Age.clear();
   AgeD0.clear();
   AgeD1.clear();
  Sex.clear();
  ChestPain.clear();
  BloodSugar.clear();
  BloodPressure.clear();
   BloodPressureD0.clear();                    /** --- Numerical --- **/
   BloodPressureD1.clear();                          //--- (1)Age
  Cholesterol.clear();                       //--- (4)BloodPressure
   CholesterolD0.clear();                      //--- (5)Cholesterol
   CholesterolD1.clear();                       //--- (8)Thalach
  Restecg.clear();                           //--- (10)Oldpeak
  Thalach.clear();
   ThalachD0.clear();
   ThalachD1.clear();
  Exang.clear();                           /** --- Nominal --- **/
  Oldpeak.clear();                                //--- (2)Sex
   OldpeakD0.clear();                           //--- (3)ChestPain
   OldpeakD1.clear();                           //---(6)BloodSugar
  Slope.clear();                              //---(7)Restang
  CA.clear();                                  //---(9)Exang
  Thal.clear();                               //--- (11)Slope
  Height.clear();                                 //--- (12)CA
  Weight.clear();                                 //--- (13)Thal
  Diagnosis.clear();                          //--- (16)Diagnosis
 }
}
```

Prediction.java

```
package com.heart.naivebayes;

import java.io.*;
import java.util.*;

public class Prediction
{
```

```
double b1=0.0d;
double pdf2;
 public double findMean(ArrayList value) {
   double mean = 0.0;
   double sum = 0.0;
   for(int i=0;i<value.size();i++)
   {
     String no= (String)value.get(i);
     double num = Double.valueOf(no);
     sum = sum + num;
         }
   mean = sum / value.size();
   return mean;
 }

 public double findNominalPDF(int a,int b) {
   a1=Double.valueOf(a);
   b1=Double.valueOf(b);
   pdf2 = a1/b1;
   return pdf2;
 }

//System.out.println("A : "+a);
//System.out.println("B : "+b);
//System.out.println("PDF2 : "+pdf2);

 public double findStandardDeviation(ArrayList value, double mean) {
    double sum = 0.0;
    double sd = 0.0;
    int length =  value.size();
    for (int i=0;i< length ; i++)
          {
            String no= (String)value.get(i);
      double num =  Double.valueOf(no);
      sum = sum + ((num - mean)*(num - mean));
    }
    sd = Math.sqrt(sum / (length -1));
    return sd;
  }

  public double findNumericalPDF(double value ,double mean , double sd) {
    double pdf = 0.0;
    pdf = (1/(sd * (Math.sqrt( 2* Math.PI)))) * (Math.pow(Math.E, -((Math.pow((value -
mean),2))/(2 * Math.pow(sd,2)))));
    return pdf;
  }
 }
```

REFERENCES

[1] Sellappan Palaniappan, Rafiah Awang "Intelligent Heart Disease Prediction System Using Data Mining Techniques" *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 No.8, August 2008

[2] http://www.statsoft.com/textbook/stdatmin.html

[3] http://www.mathworks.com

[4] Arun and K.Pujari, "Data mining Techniques", University Press, First Edition, 2001.

[5] "Data Mining" [Online]
Available : http://en.wikipedia.org/wiki/Data_mining

[6] Shantakumar B.Patil, Y.S.Kumaraswamy "Intelligent and Effective Heart Attack Prediction System Using Data Mining and Artificial Neural Network" *European Journal of Scientific Research, ISSN 1450-216X* Vol. 31 No.4 (2009), © Euro Journals Publishing, Inc. 2009, [Online].
Available: http://www.eurojournals.com/ejsr.htm

[7] Boleslaw Szymanski, Long Han, Mark Embrechts, Alexander Ross, Karsten Sternickel, Lijuan Zhu, "Using Efficient Supanova Kernel For Heart Disease Diagnosis", proc. NNIE 06, intelligent engineering systems through artificial neural networks, Vol. 16, pp:305-310, 2006.

[8] Latha Parthiban and R.Subramanian, "Intelligent Heart Disease Prediction System Using CANFIS and Genetic Algorithm", *International Journal of Biological, Biomedical and Medical Sciences;* 3, 2008

[9] Tzung-I Tang, Gang Zheng, Yalou Huang, Guangfu Shu, Pengtao Wang, "A Comparative Study of Medical Data Classification Methods Based on Decision Tree and System Reconstruction Analysis", IEMS, Vol. 4, No. 1, pp. 102-108, June 2005.