2.INTRODUCTION

Today’s era ia witnessing an alarmig ascent in the incidences of numerous life style disorders like Heart disease, Hypertension, Obesity, etc. Among these, Heart disease is on top of the list which is the leading cause of death in most of the countries. It includes diseases of heart muscles, valves, conduction system, heart attack & others. Myocardial infarction or heart attack is the major one among all other types of heart diseases. Heart diseases are seen in all the classes of people in recent times, in contrast to previous days when it was disease of rich class people.

Heart disease is even being highlighted as a silent killer which leads to death of a person without obvious symptoms. This nature of the disease is the cause of growing anxiety about the disease & its consequences. Hence continued efforts are being done to predict the possibility of this deadly disease in prior. So that various tools & techniques are regularly being experimented to suit the present day health needs. Data mining techniques can be a boon in this regard.

Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analyzing to extract the desired data we can arrive at a conclusion. This technique can be very well adapted to the do the prediction of heart disease. As the well known quote says “Prevention is better than cure”, early prediction & its control can be helpful to prevent & decrease the death rates due to heart disease

3.LITERATUREREVIEW

The paper by Vanishree. K developed asystem for diagnosis of congenital heartdisease using decision support system. Itused Back propagation Neural Network withMLP. It was based on the data – signs of theheart disease, symptoms & result ofevaluation obtained from the patient. Itshowed 90 % accurate results [3].The study done by Kharya highlighted thefact that artificial neural network is thefrequently used technique for prediction inthe medical field. The paper alsodemonstrated the merits & de merits of themachine learning techniques like Decisiontree, Naïve Bayes, Neural network &SVM[4].

C.D. Katsis et al study devised ways usingCorrelation Feature Selection (CFS)procedure & an Artificial ImmuneRecognition System (AIRS) classifier todiagnose reast cancer. Data for the studywas collected from 53 patients among the4726 cases. Biopsy was taken in all thepatients & it was kept as standard parameterto validate the methodology. The featuresalong with the biopsy result were used forthe analysis in 53 patients. Adoption ofSVM technique resulted in 70.00 + 6.33%accurate result [5].

4.PROBLEM IDENTIFICATION&OBJECTIVES

The heart is important organ or part of our body. Life is itself dependent on efficient working of heart. If operation of heart is not proper, it will affect the other body parts of human such as brain, kidney etc. It is nothing more than a pump, which pumps blood through the body. If circulation of blood in body is inefficient the organs like brain suffer and if heart stops working altogether, death occurs within minutes. Life is completely dependent on efficient working of the heart. The term Heart disease refers to disease of heart & blood vessel system within it.

Data mining is a new powerful technology which is of high interest in computer world. It is a sub field of computer science that uses already existing data in different databases to transform it into new researches and results. It makes use of Artificial Intelligence, machine learning and database management to extract new patterns from large data sets and the knowledge associated with these patterns. The actual task is to extract data by automatic or semi-automatic means. The different parameters included in data mining includes clustering, forecasting, path analysis and predictive analysis.

Data mining has many applications in the fields of telecommunication industry, financial data analysis biological data analysis and much more. With the growing research in the field of health informatics a lot of data is being produced. The analysis of such a large amount of data is very hard and requires excessive knowledge. E-healthcare applies data mining and telecommunication techniques for health diagnosis. E-health was primarily used for patient data analysis and disease diagnosis at various levels. Since the knowledge of its vast use more and more attention has been paid to this field from clinical data analysis to record management of users.

The system is first taught with various symptoms related to herat disease associated with each system. User gives the knowledge of symptoms he/she is dealing with. The machine processes these symptoms and provides the results. With advancement of technology more and more smart systems are being designed with better data mining technologies to give the most accurate results that could be associated with the disease. It then processes users symptoms to predict either the user is at risk or normal.

Objective:

It is the application of computing and communication technologies to optimize health information processing by collection, storage, effective retrieval (in due time and place).The proposed system is mainly used by the all the people where confidentiality and integrity of the data has utmost importance. Computer assisted information retrieval may help support quality decision making and to avoid human error. Imagine a doctor who has to examine 5 patient records; he or she will go through them with ease. If the number of records grows with a time constraint, it is almost certain that the accuracy with which the doctor delivers the results will not be as high as the ones obtained when he had only five records to be analyzed.

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

5.SYSTEM METHODOLOGY

5.1 Process Model:

The Heart disease based on the data set values stored by using the MSQL database as backend server.

Using this training data set values it is possible to predict the presence or absence of Heart disease in that particular patient record. Here classification techniques like Naïve Bayes & Decision tree algorithm were utilized. It is a web based application. In the front end .NET framework which acts as client. This paper even compared the accuracy of results of prediction obtained between the two algorithms.

A. Data base server

The data set was collected from UCI laboratory. It contains 13 attributes which include sex, serum cholesterol level, resting ECG etc. This data set was stored using MSQL database. It acts as a DB server

(backend server). Connection was established between DB server & client (front end). Queries are posed on the server using this DB & each individual data is retrieved.

B. Data preprocessing

In this step data is taken out from UCI repository in a recognized format. Missing fields are evacuated in this process & thus the data is transformed. Mean is entered in place of missing attributes.

C. Classsification

In data mining mainly two methods are used, supervised & unsupervised learning. Supervised learning uses a training set inorder to learn model parameters. But no such training set like k – means custering isutilised in unsupervised learning.Data mining has got two most frequent modeling goals– classification &prediction.Classification model classifies discrete, unordered values or data.

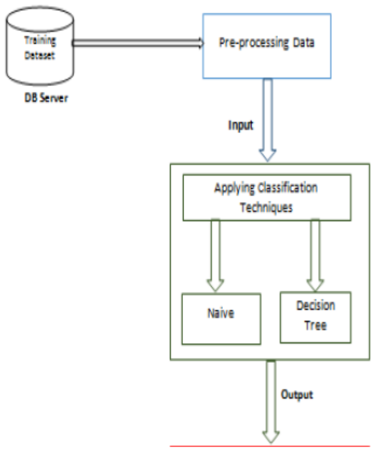
In this prediction process, the classification

techniques utilized are,

1)Decision tree

2)Naïve Bayes

5.2 Architecture:-



Flow Chart:-

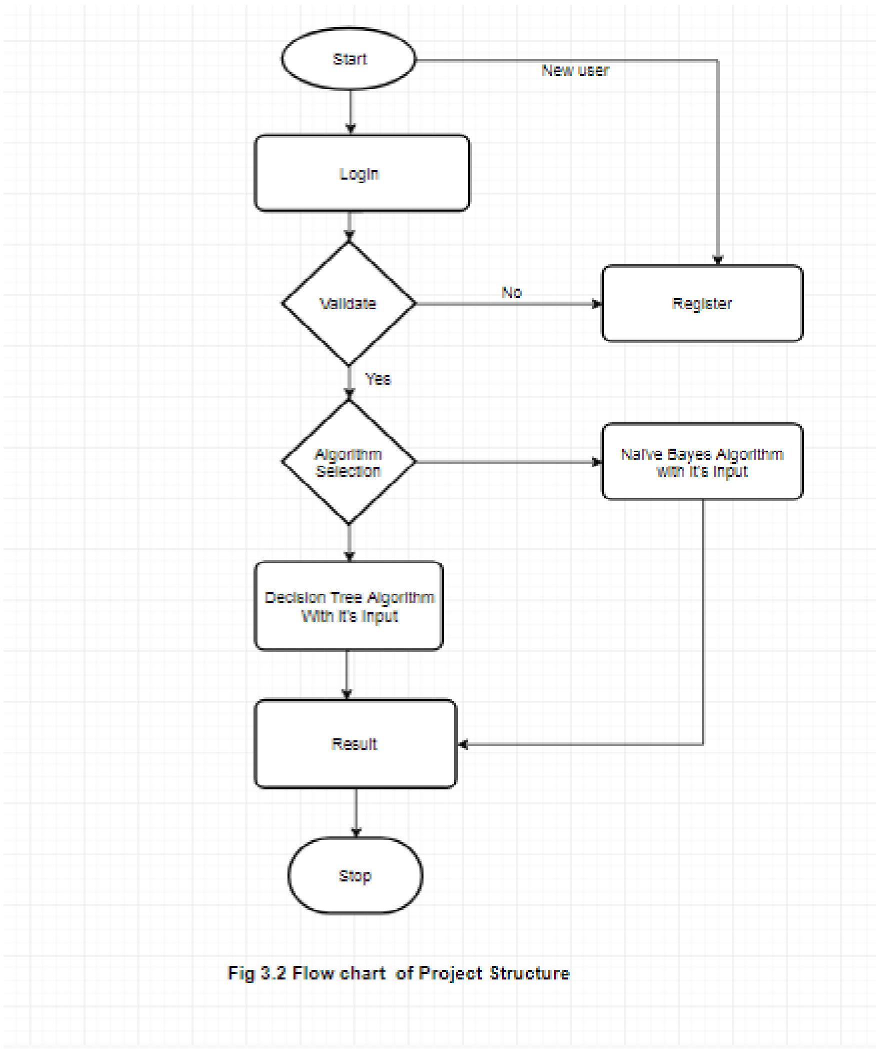

Start
New user
Login
Validate
No
Register
Yes
Algorithm Selection
Naive Bayes Algorithm with it's Input
Decision Tree Algorithm With it's Input
Result
Stop

Fig 3.2 Flow chart of Project Structure

6.OVERVIEW OFTECHNOLOGY

6.1 Data Mining:

Data Mining is an analytic process designed to explore data (usually large amounts of data - typically business or market related - also known as "big data") in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction - and 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) the initial exploration,

(2) model building or pattern identification with validation/verification, and

(3) deployment (i.e., the application of the model to new data in order to generate predictions)

Stage 1: Exploration.

This stage usually starts with data preparation which may involve cleaning data, data transformations, 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 (depending on the statistical methods which are being considered). 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 (see *Exploratory Data Analysis (EDA)*) 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.

Stage 2: Model building and validation.

This stage involves considering various models and choosing the best one based on their predictive performance (i.e., explaining the variability in question and producing stable results across samples). This may sound like a simple operation, but in fact, it sometimes involves a very elaborate process. There are a variety of techniques developed to achieve that goal - many of which are based on so-called "competitive evaluation of models," that is, 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, Stacking (Stacked Generalizations), and Meta-Learning.

Stage 3: Deployment.

That 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.

6.2 Data Mining Techniques

1.Classification:

This analysis is used to retrieve important and relevant information about data, and metadata. This data mining method helps to classify data in different classes.

2. Clustering:

Clustering analysis is a data mining technique to identify data that are like each other. This process helps to understand the differences and similarities between the data.

3. Regression:

Regression analysis is the data mining method of identifying and analyzing the relationship between variables. It is used to identify the likelihood of a specific variable, given the presence of other variables.

4. Association Rules:

This data mining technique helps to find the association between two or more Items. It discovers a hidden pattern in the data set.

5. Outer detection:

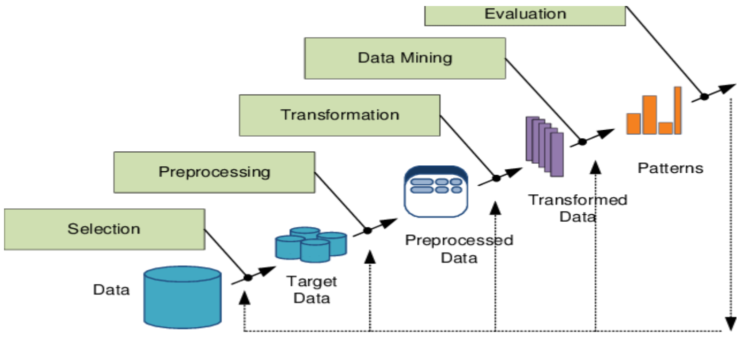
This type of data mining technique refers to observation of data items in the dataset which do not match an expected pattern or expected behavior. This technique can be used in a variety of domains, such as intrusion, detection, fraud or fault detection, etc. Outer detection is also called Outlier Analysis or Outlier mining.

6. Sequential Patterns:

This data mining technique helps to discover or identify similar patterns or trends in transaction data for certain period.

7. Prediction:

Prediction has used a combination of the other data mining techniques like trends, sequential patterns, clustering, classification, etc. It analyzes past events or instances in a right sequence for predicting a future event.



Benefits of Data Mining:

Data mining technique helps companies to get knowledge-based information.

Data mining helps organizations to make the profitable adjustments in operation and production.

The data mining is a cost-effective and efficient solution compared to other statistical data applications.

Data mining helps with the decision-making process.

Facilitates automated prediction of trends and behaviors as well as automated discovery of hidden patterns.

It can be implemented in new systems as well as existing platforms

It is the speedy process which makes it easy for the users to analyze huge amount of data in less time.

Challenges of Implementation of Data mine:

Skilled Experts are needed to formulate the data mining queries.

Overfitting: Due to small size training database, a model may not fit future states.

Data mining needs large databases which sometimes are difficult to manage

Business practices may need to be modified to determine to use the information uncovered.

If the data set is not diverse, data mining results may not be accurate.Integration information needed from heterogeneous databases and global information systems could be complex.

6.3 Algorithms:

6.3.1 Naïve Bayes:

Bayesian classification is based on Baye’s theorem, described next. Studies comparing classification algorithms have found a simple Bayesian classifier known as the naive Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called classconditional independence. It is made to simplify the computations involved and, in this sense, is considered “naive.

Bayes’ Theorem Bayes’ theorem is named after Thomas Bayes, a nonconformist English clergyman who did early work in probability and decision theory during the 18th century. Let X be a data tuple. In Bayesian terms, X is considered “evidence.” As usual, it is described by measurements made on a set of n attributes. Let H be some hypothesis such as that the data tuple X belongs to a specified class C. For classification problems, we want to determine P(H|X), the probability that the hypothesis H holds given the “evidence” or observed data tuple X. In other words, we are looking for the probability that tuple X belongs to class C, given that we know the attribute description of X.

P(H|X) is the posterior probability, or **a posteriori probability**, of H conditioned on X. For example, suppose our world of data tuples is confined to customers described by the attributes age and income, respectively, and that X is a 35-year-old customer with an income of $40,000. Suppose that H is the hypothesis that our customer will buy a computer. Then P(H|X) reflects the probability that customer X will buy a computer given that we know the customer’s age and income.

In contrast, P(H) is the prior probability, or **a priori probability**, of H. For our example, this is the probability that any given customer will buy a computer, regardless of age, income, or any other information, for that matter. The posterior probability, P(H|X), is based on more information (e.g., customer information) than the prior probability, P(H), which is independent of X.

Similarly, P(X|H) is the posterior probability of X conditioned on H. P(X) is the prior probability of X.

“How are these probabilities estimated?” P(H), P(X|H), and P(X) may be estimated from the given data, as we shall see next. Bayes’ theorem is useful in that it provides a way of calculating the posterior probability, P(H|X), from P(H), P(X|H), and P(X). Bayes’ theorem is P(H|X) = P(X|H)P(H) P(X) .

Naïve bayesClassification

The naïvebayes classifier, or simple Bayesian classifier, works as follows:

1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X = (x1, x2,..., xn), depicting n measurements made on the tuple from n attributes, respectively, A1, A2,..., An.

2. Suppose that there are m classes, C1, C2,..., Cm. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve bayes classifier predicts that tuple X belongs to the class Ci if and only if P(Ci |X) > P(Cj |X) for 1 ≤ j ≤ m,j 6= i. Thus, we maximize P(Ci |X). The class Ci for which P(Ci |X) is maximized is called the maximum posteriori hypothesis. By Bayes’ theorem

P(Ci |X) = P(X|Ci)P(Ci) /P(X)

3. As P(X) is constant for all classes, only P(X|Ci)P(Ci) needs to be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, P(C1) = P(C2) = ··· = P(Cm), and we would therefore maximize P(X|Ci). Otherwise, we maximize P(X|Ci)P(Ci). Note that the class prior probabilities may be estimated by P(Ci) = |Ci,D|/|D|, where |Ci,D| is the number of training tuples of class Ci in D.

4. Given data sets with many attributes, it would be extremely computationally expensive to compute P(X|Ci). To reduce computation in evaluating P(X|Ci), the na¨ıve assumption of class-conditional independence is made. This presumes that the attributes’ values are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes).

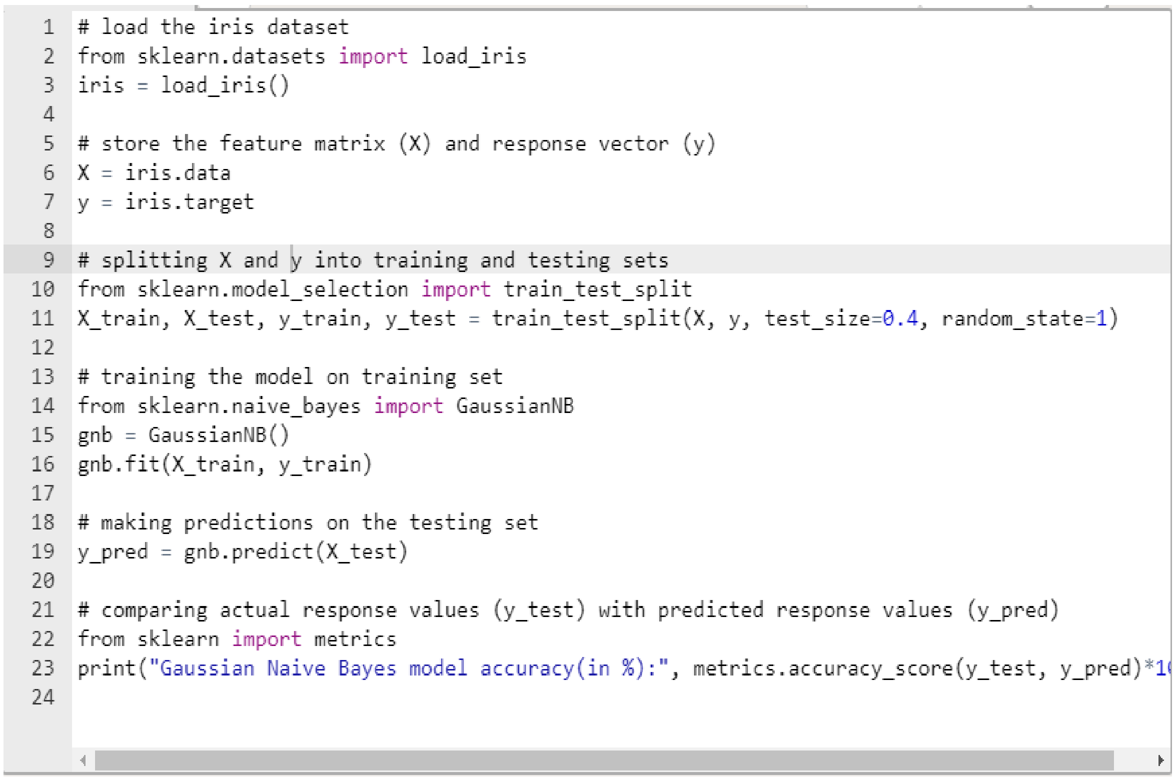
Thus,

P(X|Ci) = (k=1 to n)π P(xk |Ci) = P(x1|Ci) × P(x2|Ci) × ··· × P(xn|Ci).

We can easily estimate the probabilities P(x1|Ci), P(x2|Ci),..., P(xn|Ci) from the training tuples. Recall that here xk refers to the value of attribute Ak for tuple X. For each attribute, we look at whether the attribute is categorical or continuous-valued. For instance, to compute P(X|Ci), we consider the following:

(a) If Ak is categorical, then P(xk |Ci) is the number of tuples of class Ci in D having the value xk for Ak , divided by |Ci,D|, the number of tuples of class Ci in D.

(b) If Ak is continuous-valued, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ



```
# load the iris dataset
from sklearn.datasets import load_iris
iris = load_iris()

# store the feature matrix (X) and response vector (y)
X = iris.data
y = iris.target

# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)

# training the model on training set
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)

# making predictions on the testing set
y_pred = gnb.predict(X_test)

# comparing actual response values (y_test) with predicted response values (y_pred)
from sklearn import metrics
print("Gaussian Naive Bayes model accuracy(in %):", metrics.accuracy_score(y_test, y_pred)*1
```

6.3.2 Decision Tree Induction

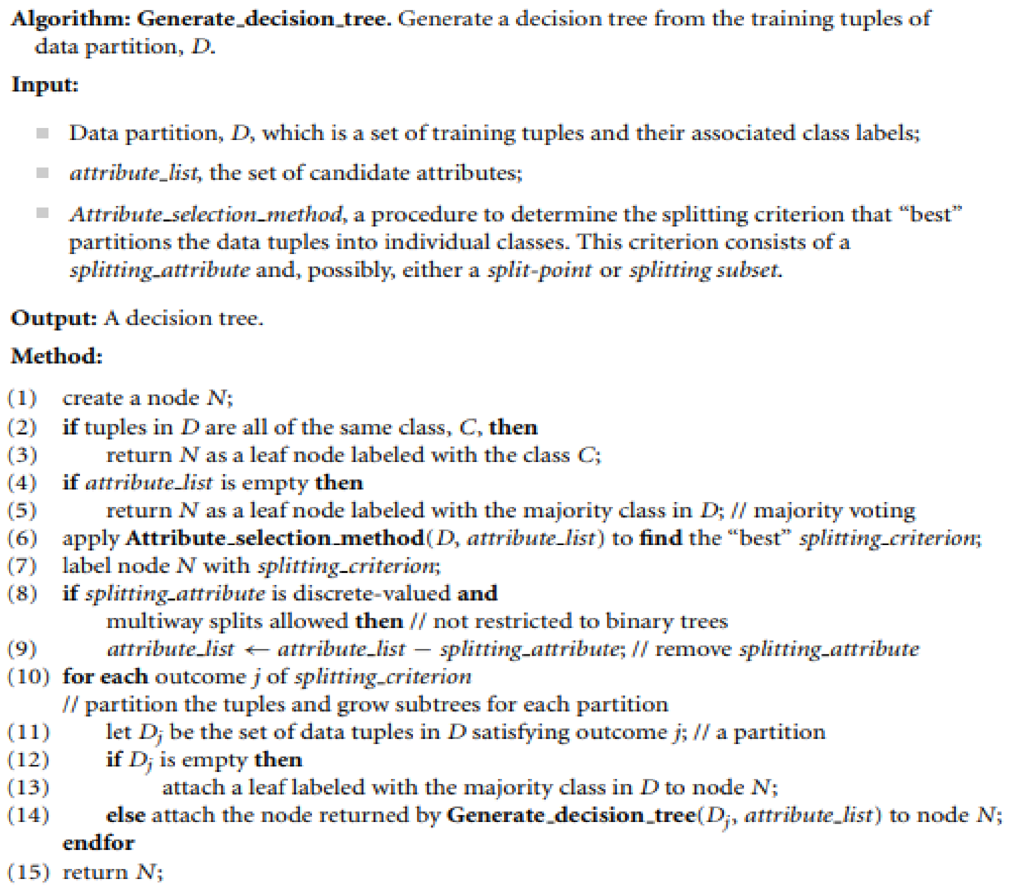
Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules.

The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle multidimensional data. Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans. The learning and classification steps of decision tree induction are simple and fast. In general, decision tree classifiers have good accuracy. However, successful use may depend on the data at hand. Decision tree induction algorithms have been used for classification in many application areas such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.Decisiontrees are the basis of several commercial rule induction systems.

During tree construction, attribute selection measures are used to select the attribute that best partitions the tuples into distinct classes. When decision trees are built, many of the branches may reflect noise or outliers in the training data. Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data. Scalability issues for the induction of decision trees.

During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). This work expanded on earlier work on concept learning systems, described by E. B. Hunt, J. Marin, and P. T. Stone. Quinlan later presented C4.5 (a successor of ID3), which became a benchmark to which newer supervised learning algorithms are often compared. In 1984, a group of statisticians (L. Breiman, J. Friedman, R. Olshen, and C. Stone) published the book Classification and Regression Trees (CART), which described the generation of binary decision trees. ID3 and CART were invented independently of one another at around the same time, yet follow a similar approach for learning decision trees from training tuples. These two cornerstone algorithms spawned a flurry of work on decision tree induction.

ID3, C4.5, and CART adopt a greedy (i.e., nonbacktracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. Most algorithms for decision tree induction also follow a top-down approach, which starts with a training set of tuples and their associated class labels. The training set is recursively partitioned into smaller subsets as the tree is being built. A basic decision tree algorithm is summarized in Figure 8.3. At first glance, the algorithm may appear long, but fear not! It is quite straightforward.



Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition, *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split-point* or *splitting subset*.

Output: A decision tree.

Method:

```
(1)  create a node N;
(2)  if tuples in D are all of the same class, C, then
(3)      return N as a leaf node labeled with the class C;
(4)  if attribute_list is empty then
(5)      return N as a leaf node labeled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  if splitting_attribute is discrete-valued and
         multiway splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let D_j be the set of data tuples in D satisfying outcome j; // a partition
(12)     if D_j is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(D_j, attribute_list) to node N;
     endfor
(15) return N;
```

The algorithm is called with three parameters: D, attribute list, and Attribute selection method. We refer to D as a data partition. Initially, it is the complete set of training tuples and their associated class labels. The parameter attribute list is a list of attributes describing the tuples. Attribute selection method specifies a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class. This procedure employs an attribute selection measure such as information gain or the Gini index. Whether the tree is strictly binary is generally driven by the attribute selection measure. Some attribute selection measures, such as the Gini index, enforce the resulting tree to be binary. Others, like information gain, do not, therein allowing multiway splits (i.e., two or more branches to be grown from a node).
The tree starts as a single node, N, representing the training tuples in D (step 1).

1. A is **discrete-valued**: In this case, the outcomes of the test at node N correspond directly to the known values of A. A branch is created for each known value, aj , of A and labeled with that value. Partition Dj is the subset of class-labeled tuples in D having value aj of A. Because all the tuples in a.given partition have the same value for A, A need not be considered in any future partitioning of the tuples.

2. A is **continuous-valued**: In this case, the test at node N has two possible outcomes, corresponding to the conditions A ≤ split point and A > split point, respectively, where split point is the split-point returned by Attribute selection method as part of the splitting criterion. (In practice, the split-point, a, is often taken as the midpoint of two known adjacent values of A and therefore may not actually be a preexisting value of A from the training data.) Two branches are grown from N and labeled according to the previous outcomes (Figure 8.4b). The tuples are partitioned such that D1 holds the subset of class-labeled tuples in D for which A ≤ split point, while D2 holds the rest.

3. A is **discrete-valued and a binary tree** must be produced (as dictated by the attribute selection measure or algorithm being used): The test at node N is of the form "A ∈SA?," where SA is the splitting subset for A, returned by Attribute selection method as part of the splitting criterion. It is a subset of the known values of A. If a given tuple has value aj of A and if aj∈ SA, then the test at node N is satisfied. Two branches are grown from N (Figure 8.4c). By convention, the left branch out of N is labeled yes so that D1 corresponds to the subset of class-labeled tuples in D that satisfy the test. The right branch out of N is labeled no so that D2 corresponds to the subset of class-labeled tuples from D that do not satisfy the test.

6.4 Java Technology:-

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

6.4.1 Java Application's

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

Applets: The set of conventions used by applets.

Networking: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.

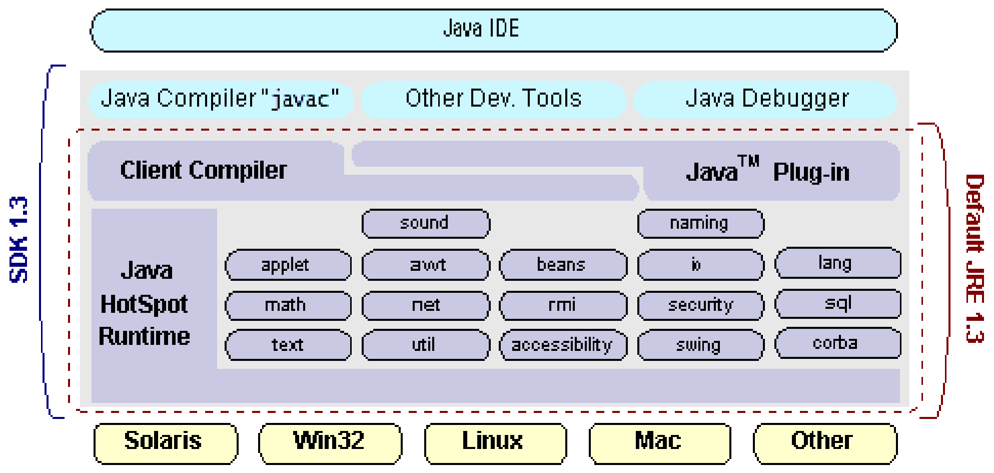
Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

Security: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

Software components: Known as JavaBeans^{TM}, can plug into existing component architectures.

Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

Java Database Connectivity (JDBC™): Provides uniform access to a wide range of relational databases.



6.4.2AWT:

The **Abstract Window Toolkit** (AWT) is Java's original platform-independent windowing, graphics, and user-interface widget toolkit. The AWT is now part of the Java Foundation Classes (JFC) — the standard API for providing a graphical user interface (GUI) for a Java program.AWT is also the GUI toolkit for a number of Java ME profiles. For example, Connected Device Configuration profiles require Java runtimes on mobile telephones to support AWT.

Components and containers

All the elements like buttons, text fields, scrollbars etc are known as components. In AWT we have classes for each component as shown in the above diagram. To have everything placed on a screen to a particular position, we have to add them to a container. A container is like a screen wherein we are placing components like buttons, text fields, checkbox etc. In short a container contains and controls the layout of components. A container itself is a component (shown in the above hierarchy diagram) thus we can add a container inside container.

Types of containers:

As explained above, a container is a place wherein we add components like text field, button, checkbox etc. There are four types of containers available in AWT: Window, Frame, Dialog and Panel. As shown in the hierarchy diagram above, Frame and Dialog are subclasses of Window class.

6.4.3 Swing:

Swing is the primary Java GUI widget toolkit. It is part of Oracle's Java Foundation Classes (JFC) — an API for providing a graphical user interface (GUI) for Java programs.
Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check box and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables and lists.
Unlike AWT components, Swing components are not implemented by platform-specific code. Instead they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element.

JPanel is Swing’s version of the AWT class Panel and uses the same default layout, FlowLayout. JPanel is descended directly from JComponent.

JFrame

is Swing’s version of Frame and is descended directly from that class. The components added to the frame are referred to as its contents; these are managed by the contentPane. To add a component to a JFrame, we must use its contentPane instead.

JInternalFrame

is confined to a visible area of a container it is placed in. It can be iconified , maximized and layered.

JWindow

is Swing’s version of Window and is descended directly from that class. Like Window, it uses BorderLayout by default. JDialog is Swing’s version of Dialog and is descended directly from that class. Like Dialog, it uses BorderLayout by default. Like JFrame and JWindow,JDialog contains a rootPane hierarchy including a contentPane, and it allows layered and glass panes. All dialogs are modal, which means the currentthread is blocked until user interaction with it has been completed. JDialog class is intended as the basis for creating custom dialogs; however, someof the most common dialogs are provided through static methods in the class JOptionPane.

JLabel, descended from JComponent, is used to create text labels.

JButton.

The abstract class AbstractButton extends class JComponent and provides a foundation for a family of button classes, including

JTextField

It allows editing of a single line of text. New features include the ability to justify the text left, right, or center, and to set the text's font.

JPasswordField

(a direct subclass of JTextField) you can suppress the display of input. Each character entered can be replaced by an echo character.This allows confidential input for passwords, for example. By default, the echo character is the asterisk.

JTextArea

allows editing of multiple lines of text. JTextArea can be used in conjunction with class JScrollPane to achieve scrolling. The underlying JScrollPane can be forced to always or never have either the vertical or horizontal scrollbar;JButton is a component the user clicks to trigger a specific action.

JRadioButton

is similar to JCheckbox, except for the default icon for each class. A set of radio buttons can be associated as a group in which onlyone button at a time can be selected.

JCheckBox

is not a member of a checkbox group. A checkbox can be selected and deselected, and it also displays its current state.

JComboBox

is like a drop down box. You can click a drop-down arrow and select an option from a list. For example, when the component has focus,pressing a key that corresponds to the first character in some entry's name selects that entry. A vertical scrollbar is used for longer lists.

JList

provides a scrollable set of items from which one or more may be selected. JList can be populated from an Array or Vector. JList does notsupport scrolling directly, instead, the list must be associated with a scrollpane. The view port used by the scroll pane can also have a user-defined border. JList actions are handled using ListSelectionListener.

JTabbedPane

contains a tab that can have a tool tip and a mnemonic, and it can display both text and an image.

JToolbar

contains a number of components whose type is usually some kind of button which can also include separators to group related componentswithin the toolbar.

FlowLayout

when used arranges swing components from left to right until there's no more space available. Then it begins a new row below it and movesfrom left to right again. Each component in a FlowLayout gets as much space as it needs and no more.

BorderLayout

places swing components in the North, South, East, West and center of a container. You can add horizontal and vertical gaps betweenthe areas.

GridLayout

is a layout manager that lays out a container's components in a rectangular grid. The container is divided into equal-sized rectangles,and one component is placed in each rectangle.

GridBagLayout

is a layout manager that lays out a container's components in a grid of cells with each component occupying one or more cells,called its display area. The display area aligns components vertically and horizontally, without requiring that the components be of the same size.

JMenubar

can contain several JMenu's. Each of the JMenu's can contain a series of JMenuItem 's that you can select. Swing provides support forpull-down and popup menus.

Scrollable JPopupMenu

is a scrollable popup menu that can be used whenever we have so many items in a popup menu that exceeds the screen visible height.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn’t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

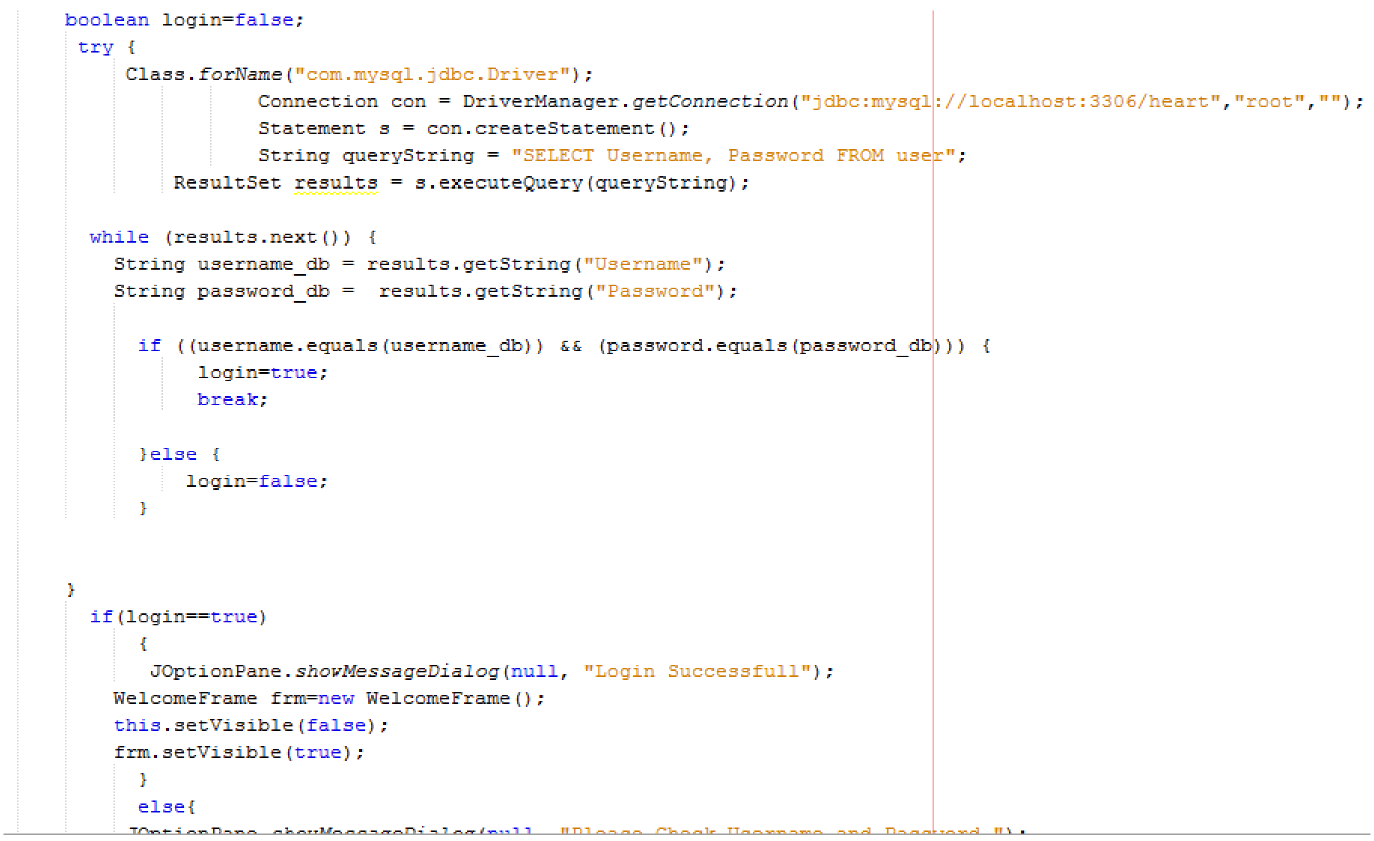
To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

7.IMPLEMENTATION

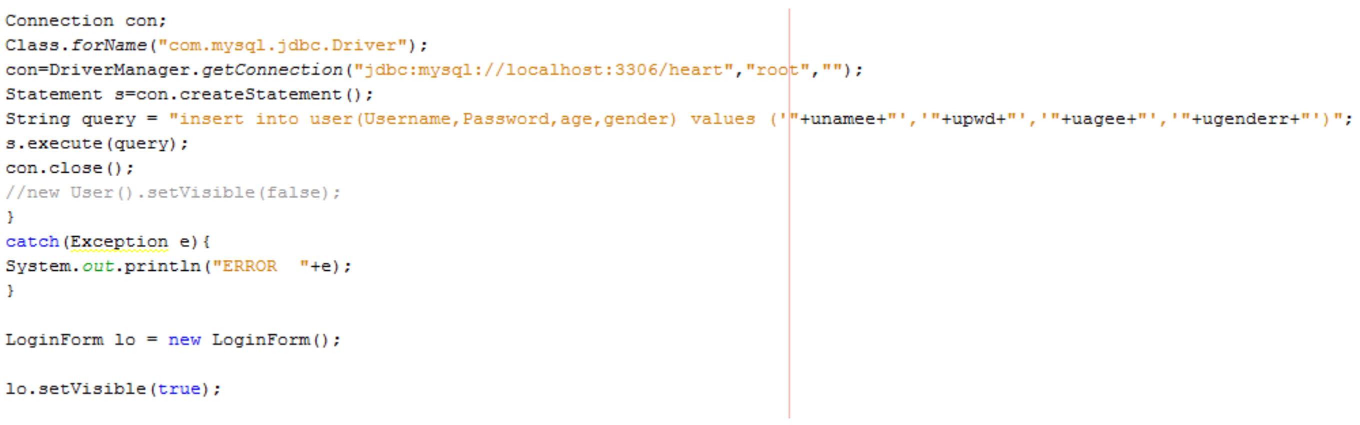
7.1 Coding:-

Login module :-



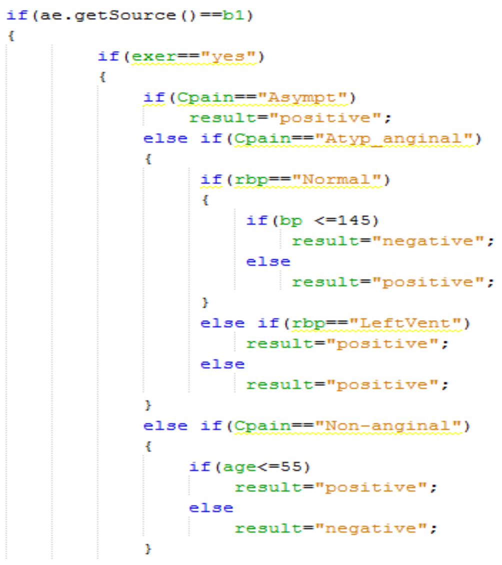
```
    boolean login=false;
     try {
         Class.forName("com.mysql.jdbc.Driver");
                    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/heart","root","");
                    Statement s = con.createStatement();
                    String queryString = "SELECT Username, Password FROM user";
             ResultSet results = s.executeQuery(queryString);

      while (results.next()) {
        String username_db = results.getString("Username");
        String password_db =  results.getString("Password");

          if ((username.equals(username_db)) && (password.equals(password_db))) {
               login=true;
               break;

          }else {
              login=false;
          }


    }
      if(login==true)
          {
           JOptionPane.showMessageDialog(null, "Login Successfull");
        WelcomeFrame frm=new WelcomeFrame();
        this.setVisible(false);
        frm.setVisible(true);
          }
          else{
         JOptionPane.showMessageDialog(null, "Please Check Username and Password ");
```

Register module :-



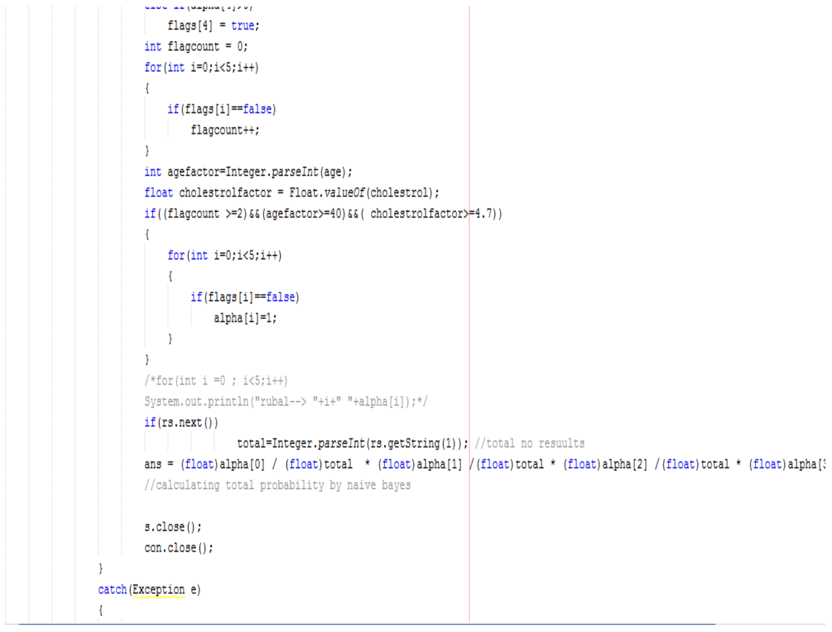
```
Connection con;
Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/heart","root","");
Statement s=con.createStatement();
String query = "insert into user(Username,Password,age,gender) values ('"+unamee+"','"+upwd+"','"+uagee+"','"+ugenderr+"')";
s.execute(query);
con.close();
//new User().setVisible(false);
}
catch(Exception e){
System.out.println("ERROR  "+e);
}

LoginForm lo = new LoginForm();

lo.setVisible(true);
```

Decision tree Algorithm:-



```
if(ae.getSource()==b1)
{
        if(exer=="yes")
        {
            if(Cpain=="Asympt")
                result="positive";
            else if(Cpain=="Atyp_anginal")
            {
                if(rbp=="Normal")
                {
                    if(bp <=145)
                        result="negative";
                    else
                        result="positive";
                }
                else if(rbp=="LeftVent")
                    result="positive";
                else
                    result="positive";
            }
            else if(Cpain=="Non-anginal")
            {
                if(age<=55)
                    result="positive";
                else
                    result="negative";
            }
```

Naïve bayes



```
            else if(alpha[4]>0)
                flags[4] = true;
            int flagcount = 0;
            for(int i=0;i<5;i++)
            {
                if(flags[i]==false)
                    flagcount++;
            }
            int agefactor=Integer.parseInt(age);
            float cholestrolfactor = Float.valueOf(cholestrol);
            if((flagcount >=2)&&(agefactor>=40)&&( cholestrolfactor>=4.7))
            {
                for(int i=0;i<5;i++)
                {
                    if(flags[i]==false)
                        alpha[i]=1;
                }
            }
            /*for(int i =0 ; i<5;i++)
            System.out.println("rubal--> "+i+" "+alpha[i]);*/
            if(rs.next())
                        total=Integer.parseInt(rs.getString(1)); //total no resuults
            ans = (float)alpha[0] / (float)total  * (float)alpha[1] /(float)total * (float)alpha[2] /(float)total * (float)alpha[
            //calculating total probability by naive bayes

            s.close();
            con.close();
        }
        catch(Exception e)
        {
```

7.2 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.

Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.
Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.2.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

7.2.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2.3 Acceptance Testing

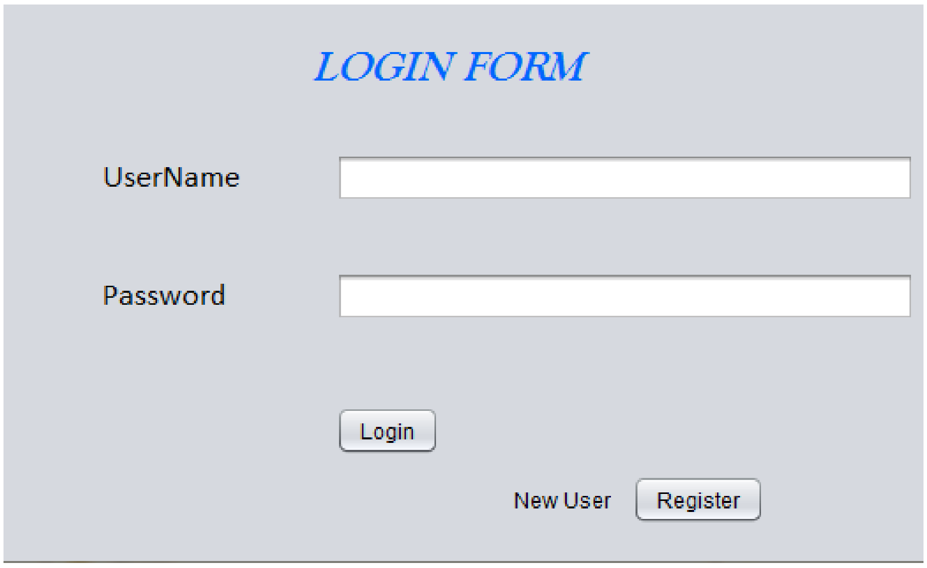
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

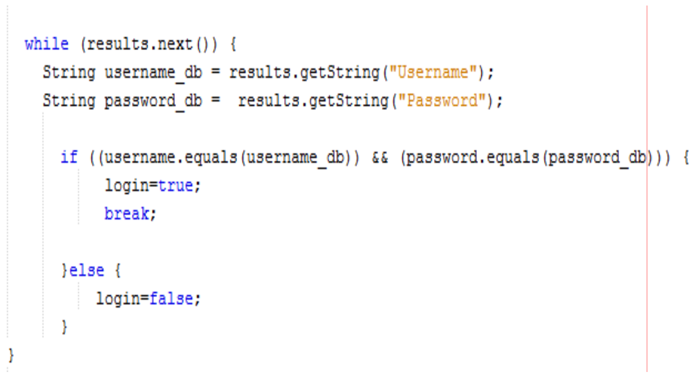
8.RESULTS

Login form:-

This module performs User Validation This is developed using AWT and the data from this page is Compared with the Registered Data in XAMPP (MYSQL) and with successful Validation an Alert message is Displayed .



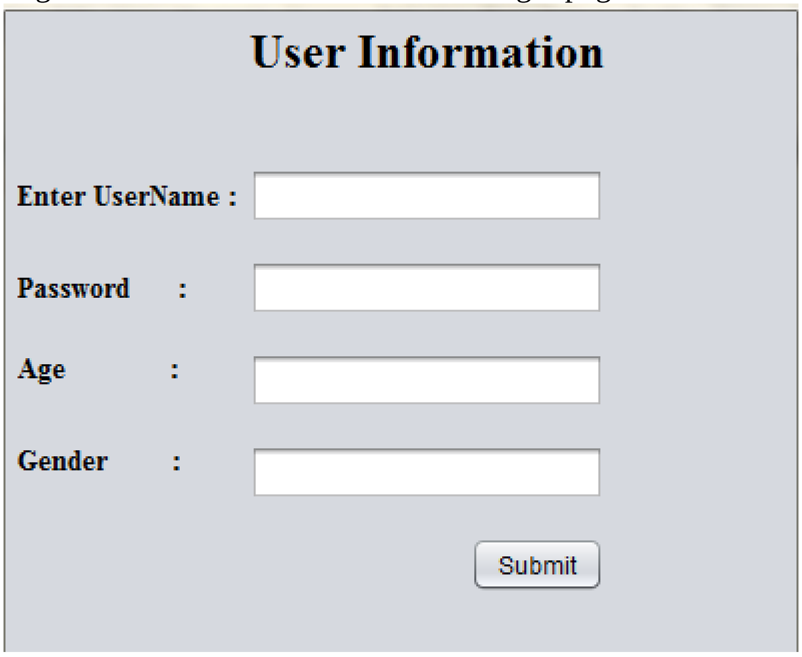
Data base validation :-



```
while (results.next()) {
  String username_db = results.getString("Username");
  String password_db =  results.getString("Password");

    if ((username.equals(username_db)) && (password.equals(password_db))) {
         login=true;
         break;

    }else {
        login=false;
    }
}
```

Registration :-

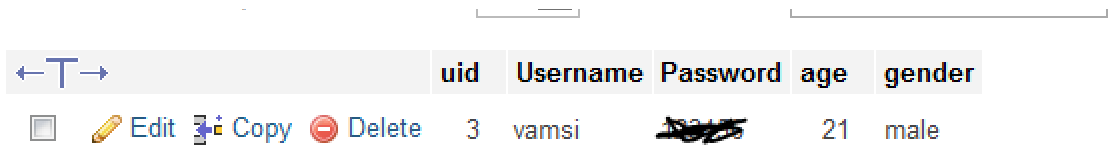
A new User's are requested to register before Login and the data from this page are collected and inserted into the MYSQL data base(XAMPP). After successful Registration The user is redirected to Login page.



User Information

Enter UserName :	
Password :	
Age :	
Gender :	

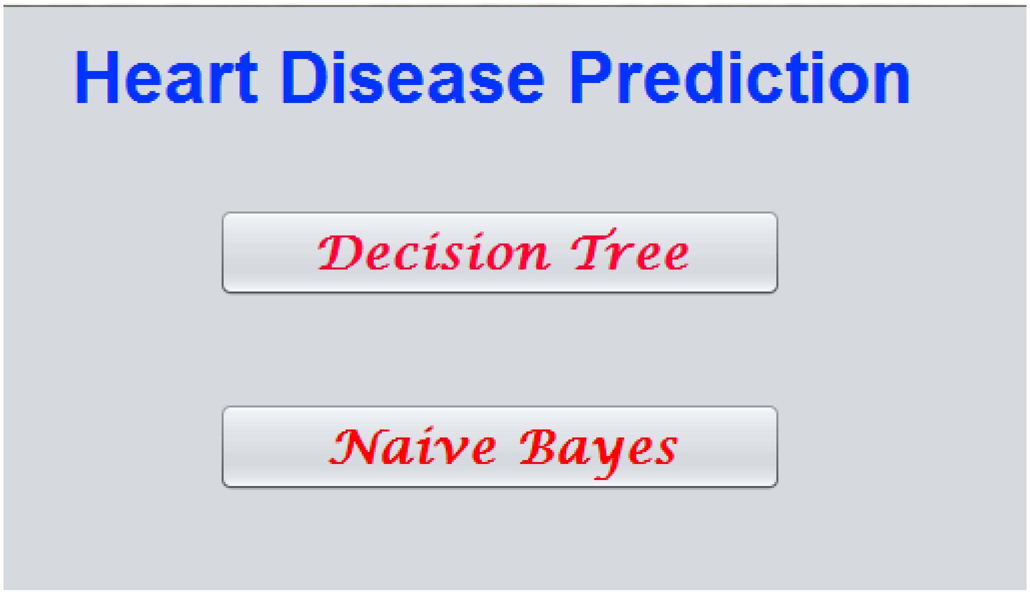
XAMPP Data Base



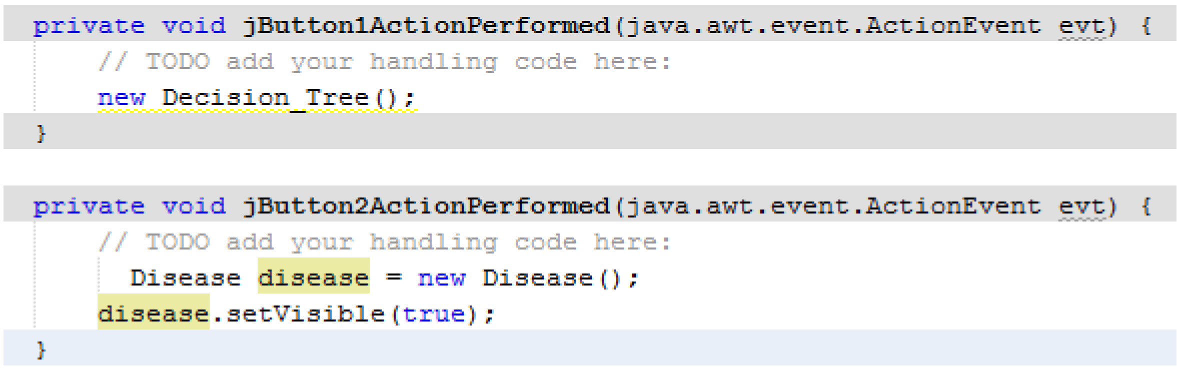
	uid	Username	Password	age	gender
Edit Copy Delete	3	vamsi	[redacted]	21	male

Welcome:-

This Module contains Two Options (Algorithm) Upon User interested Algorithm That Particular Algorithm Input file is Opened.



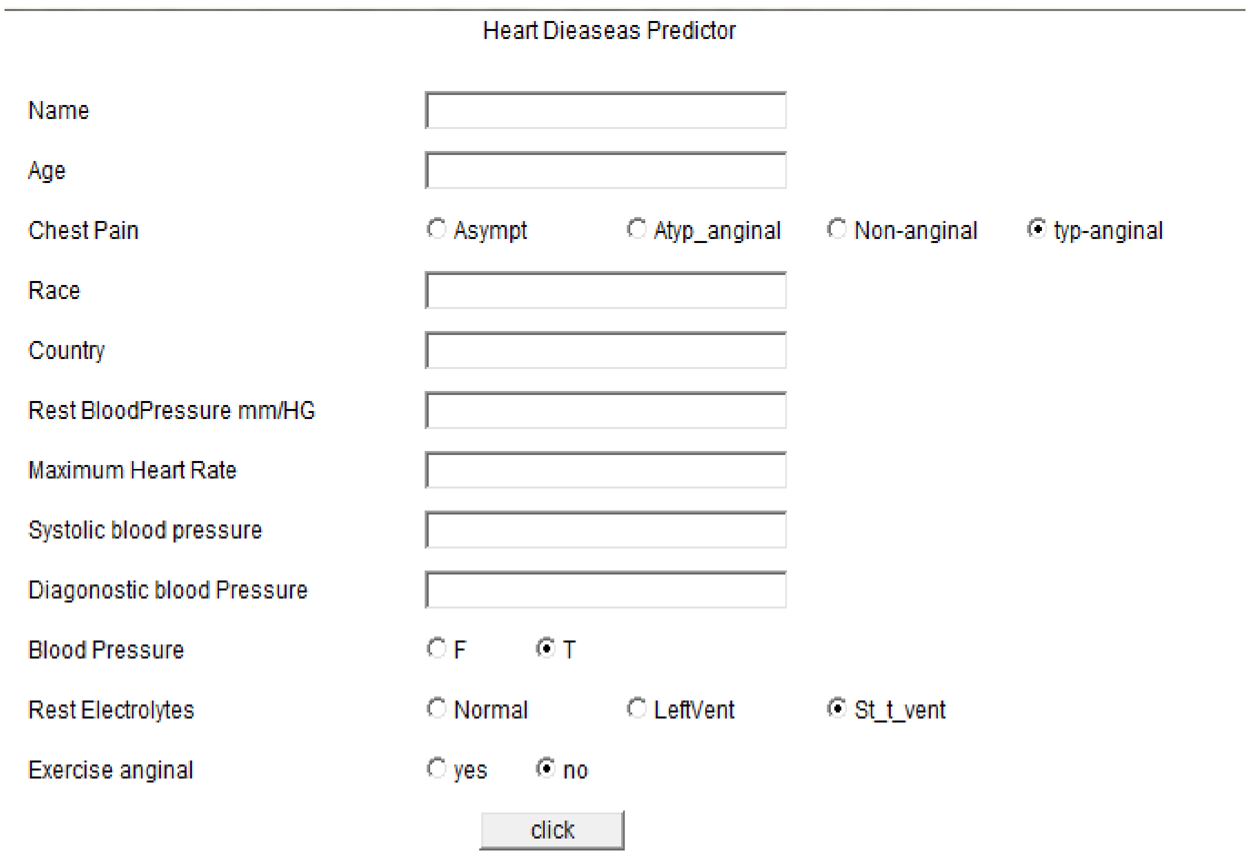
Welcome Page Code:-



```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Decision_Tree();
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
          Disease disease = new Disease();
        disease.setVisible(true);
}
```

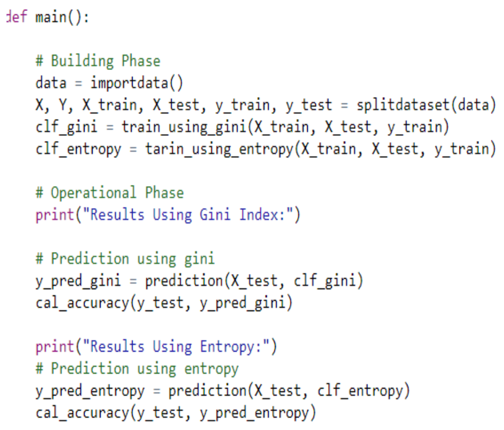
Decision tree:-

This Algorithm takes different inputs from user .Based on the input data the standard tree is evolved and Produce the Output.



Heart Dieaseas Predictor

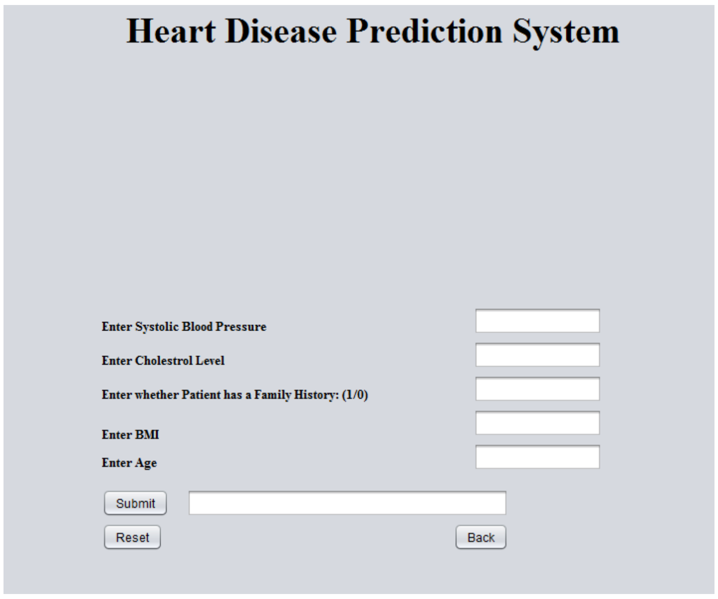
Name	
Age	
Chest Pain	Asympt · Atyp_anginal · Non-anginal · [✓ typ-anginal]
Race	
Country	
Rest BloodPressure mm/HG	
Maximum Heart Rate	
Systolic blood pressure	
Diagonostic blood Pressure	
Blood Pressure	F [✓ T]
Rest Electrolytes	Normal · LeftVent · [✓ St_t_vent]
Exercise anginal	yes [✓ no]



```
def main():

    # Building Phase
    data = importdata()
    X, Y, X_train, X_test, y_train, y_test = splitdataset(data)
    clf_gini = train_using_gini(X_train, X_test, y_train)
    clf_entropy = tarin_using_entropy(X_train, X_test, y_train)

    # Operational Phase
    print("Results Using Gini Index:")

    # Prediction using gini
    y_pred_gini = prediction(X_test, clf_gini)
    cal_accuracy(y_test, y_pred_gini)

    print("Results Using Entropy:")
    # Prediction using entropy
    y_pred_entropy = prediction(X_test, clf_entropy)
    cal_accuracy(y_test, y_pred_entropy)
```

Naïve Bayes Algorithm:-

This Algorithm takes a small set of input and this Algorithm performs Operation on this data Based on the Trained data sets and produce result as positive and negative.



Heart Disease Prediction System

Enter Systolic Blood Pressure

Enter Cholestrol Level

Enter whether Patient has a Family History: (1/0)

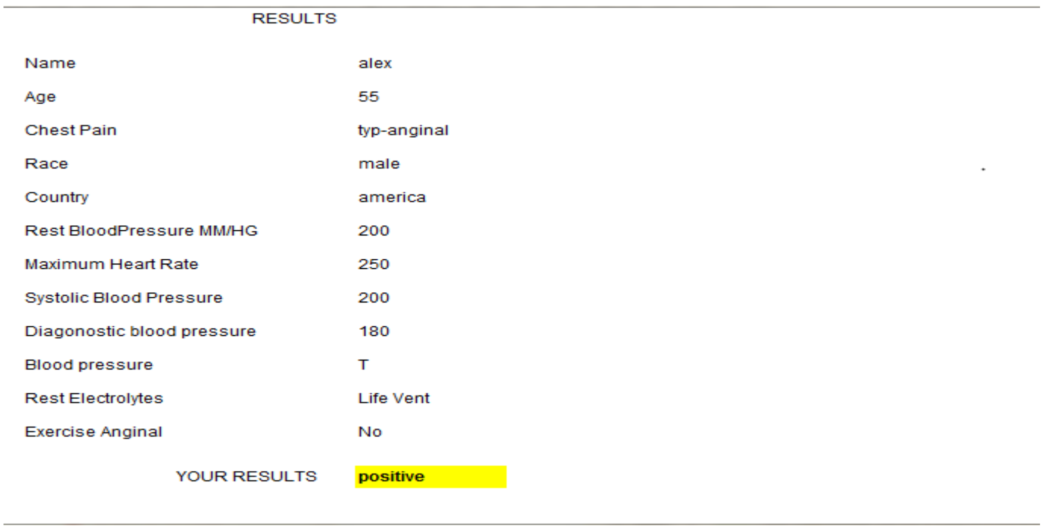
Enter BMI

Enter Age

Sample Output Images:

Decision Tree:-

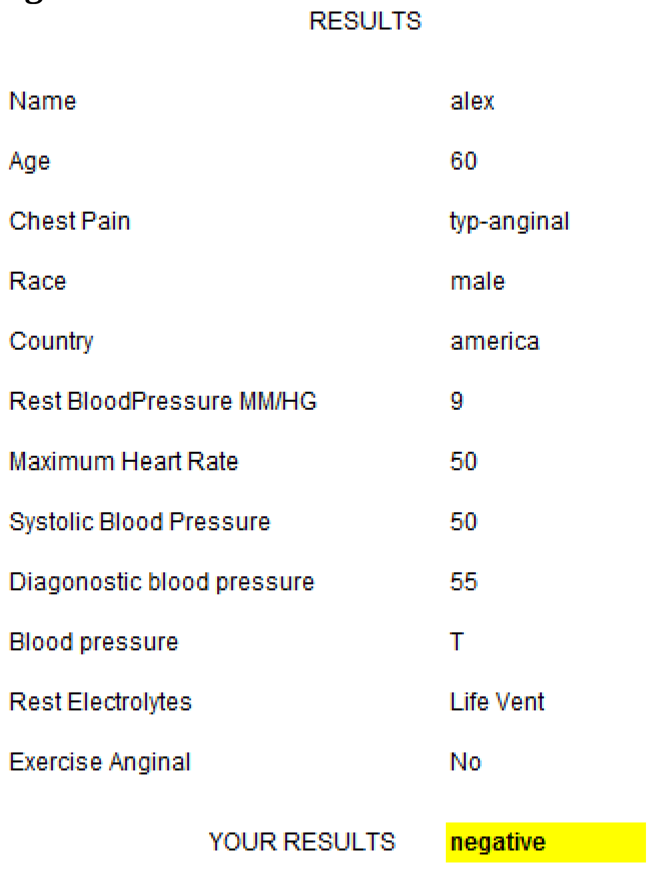
Positive result :-



RESULTS

Name	alex
Age	55
Chest Pain	typ-anginal
Race	male
Country	america
Rest BloodPressure MM/HG	200
Maximum Heart Rate	250
Systolic Blood Pressure	200
Diagonostic blood pressure	180
Blood pressure	T
Rest Electrolytes	Life Vent
Exercise Anginal	No
YOUR RESULTS	positive

Negative Result:-

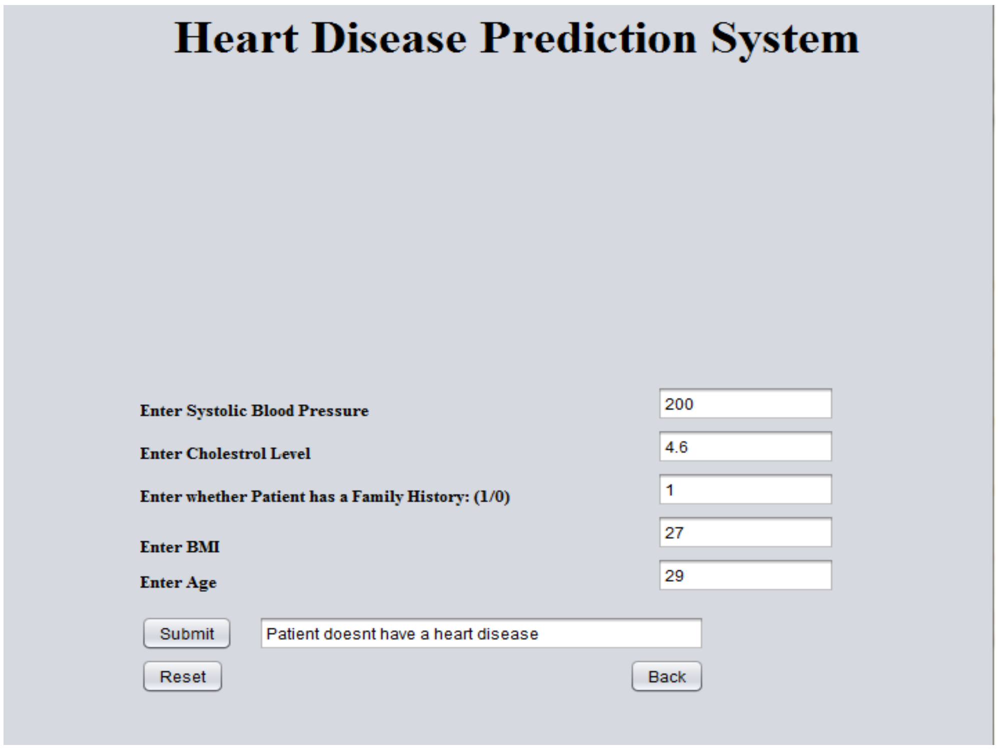


RESULTS

Name	alex
Age	60
Chest Pain	typ-anginal
Race	male
Country	america
Rest BloodPressure MM/HG	9
Maximum Heart Rate	50
Systolic Blood Pressure	50
Diagonostic blood pressure	55
Blood pressure	T
Rest Electrolytes	Life Vent
Exercise Anginal	No
YOUR RESULTS	negative

Naïve Bayes :-

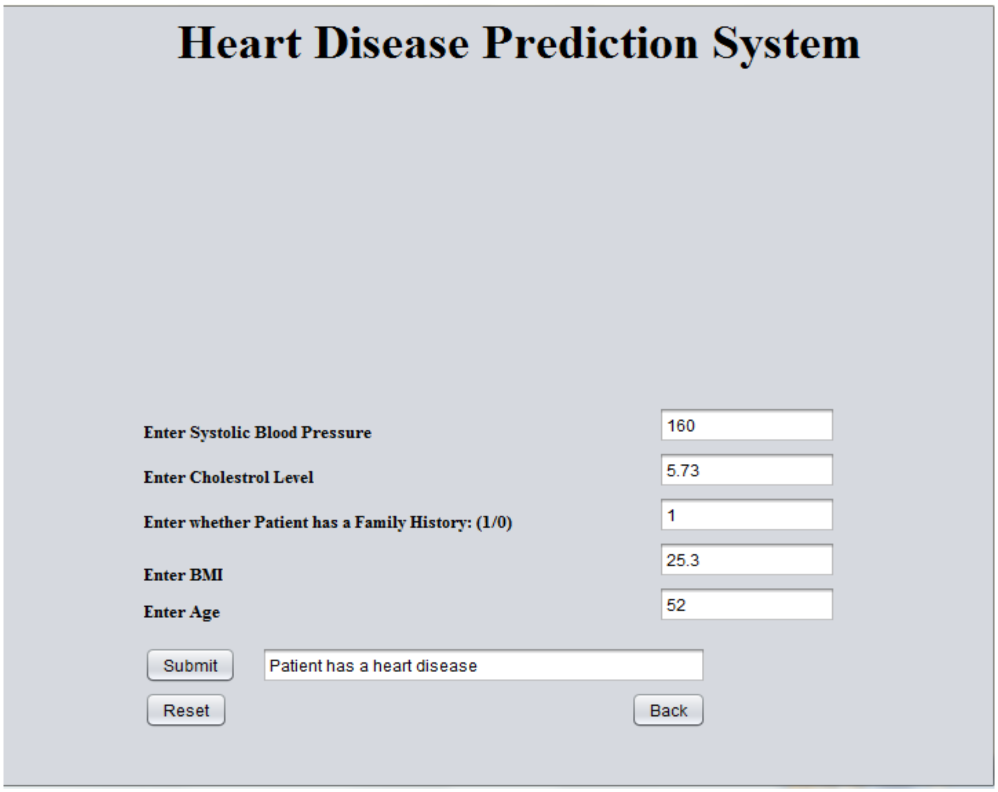
Positive result :-


Heart Disease Prediction System

Enter Systolic Blood Pressure	200
Enter Cholestrol Level	4.6
Enter whether Patient has a Family History: (1/0)	1
Enter BMI	27
Enter Age	29

Patient doesnt have a heart disease

Negative result:-


Heart Disease Prediction System

Enter Systolic Blood Pressure	160
Enter Cholestrol Level	5.73
Enter whether Patient has a Family History: (1/0)	1
Enter BMI	25.3
Enter Age	52

Patient has a heart disease

INPUT TERMINOLOGY :-

ANGINAL:-

It is a tightness, pain, or discomfort in the chest that occurs when an area of the heart muscle receives less blood oxygen than usual.

BLOOD PRESSURE:-

The pressure of the blood in the circulatory system, often measured for diagnosis since it is closely related to the force and rate of the heartbeat and the diameter and elasticity of the arterial walls.

BMI:-

A key index for relating weight to height. Abbreviated BMI. BMI is a person's weight in kilograms (kg) divided by his or her height in meters squared.

DIAGNOSTIC BLOOD PRESSURE:-

The diagnostolic pressure is specifically the minimum arterial pressure during relaxation and dilatation of the ventricles of the heart when the ventricles fill with blood

MAXIMUM HEART RATE:-

The age-related number of beats per minute of the heart when working at its maximum that is usually estimated as 220 minus one's age

REST BLOOD PRESSURE :-

When you stop exercising, your heart rate does not immediately return to your normal (resting) heart rate.

REST ELECTROLYTES:-

An electrolyte is one of the substances in the blood that helps to regulate the proper balance of body fluids.

SYSTOLIC BLOOD PRESSURE :-

The first or top number; the pressure in the arteries when the heart is beating and the arteries are filled with blood

9.CONCLUSION

The study compared the accuracy of the results obtained through two algorithms –Naïve Bayes & Decision tree. Even though both of them were good enough in predicting the Heart disease using various parameters, Decision tree was found to be the best. It gave the most accurate result whether the patient had the possibility of the heart disease. This system can also be used in future projects to detect the specific type of heart disease in particular. Thereby the diagnosis & management of Heart disease can be made simpler.

10.REFERENCES

Sites Referred:

http://java.sun.com

http://www.sourceforge.net

http://www.networkcomputing.com/

http://www.roseindia.com/

http://www.java2s.com/

http://www. javadb.com/

[1] L. Burke, J. Ma, K. Azar, G. Bennett, E.Peterson,Y. Zheng, W. Riley,J. Stephens, S. Shah, B.Suffoletto, T. Turan, B. Spring, J. SteinbergerandC.Quinn, "Current Science on Consumer Use of Mobile

Healthfor Cardiovascular Disease Prevention",Circulation, vol. 132, no. 12,pp. 1157-1213, 2015.

[2] M. Raihan1, Saikat Mondal2, Arun More3, Md.OmarFaruqe Sagor4, Gopal Sikder5, Mahbub Arab Majumder5, Mohammad Abdullah Al Manjur5 and Kushal Ghosh ," Smartphone Based Ischemic Heart Disease (Heart Attack) Risk Prediction using Clinical

Data and Data Mining Approaches, a Prototype Design", International Conference on Computer and Information Technology, December 18-20, 2016.

[3]VanishreeK,JyothiSingaraju,"Decision Support System for congential heart disease diagnosis based on signs and symptoms using neutral network" International Journal of computer applications,April 2011 Vol 19 no.6.

[4]. S.Kharya, D. Dubey, and S. Soni – Predictive Machine Learning Techniques for Breast Cancer Detection, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (6), 2013, 1023-1028.

[5] C. D. Katsis, I. Gkogkou, C.A. Papadopulos, Y.Goletsis, P. V. Boufounou, G. Stylios "Using artificial immune recognition systems in order to detect early breast cancer." International Journal of Intelligent Systems and Applications 5.2 (2013): 34.

[6] Ms. IshtakeS.H , Prof. Sanap S.A., ‘‘Intelligent Heart DiseasePrediction System Using Data Mining Techniques’’,International J. of Healthcare & Biomedical Research,Volume: 1, Issue: 3, April 2013, Pages 94-101.